DO NOWEJ PODSTAWY PROGRAMOWEJ

Część 2

PODRĘCZNIK dla szkół ponadgimnazjalnych

Informatyka Europejczyka

Zakres **rozszerzony**



Zawiera CD

Danuta Korman

Helion
EDUKACJA

Podręcznik dopuszczony do użytku szkolnego przez ministra właściwego
do spraw oświaty i wychowania i wpisany do wykazu podręczników przeznaczonych
do kształcenia ogólnego do nauczania informatyki, na podstawie opinii rzeczoznawców:
mgr. inż. Włodzimierza Kruszwickiego, dr. Leszka Rudaka, dr Iwony Wandy Grygiel.

Zakres kształcenia: rozszerzony.
Etap edukacyjny: IV.
Typ szkoły: szkoły ponadgimnazjalne.
Rok dopuszczenia 2012.

Numer ewidencyjny w wykazie: 410/2/2012

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości
lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione.
Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie
książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie
praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi
bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte
w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej
odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne
naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION
nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe
z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Joanna Zaręba
Projekt okładki: ULABUKA
Skład: Ewa Galczak

Ilustracja na okładce została wykorzystana za zgodą Shutterstock.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie?iepp22
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-4289-7

Copyright © Helion 2013
Wydanie II

Spis treści

Wstęp 6

Rozdział 1. Bazodanowe operacje arkusza kalkulacyjnego 9

1.1. Jednotabelowa baza danych — kartoteka 9

1.2. Podstawowe pojęcia 11

1.3. Tworzenie nowej bazy i import danych 12

1.4. Wyszukiwanie i organizowanie danych 14
1.4.1. Wyszukiwanie z wykorzystaniem formularza 14
1.4.2. Wyszukiwanie informacji za pomocą autofiltrowania 14
1.4.3. Wyszukiwanie informacji za pomocą filtru zaawansowanego 20
1.4.4. Sortowanie danych 23
1.4.5. Sumy częściowe (pośrednie) 25
1.4.6. Statystyki oparte na raporcie tabeli przestawnej 27
1.4.7. Zadania 30

Rozdział 2. Relacyjna baza danych 35

2.1. Bazy danych i systemy nimi zarządzające 35

2.2. Projektowanie 38
2.2.1. Zasady projektowania bazy danych 38
2.2.2. Projektowanie bazy danych w programie Microsoft Office Access 2007 43

2.3. Tabele 44
2.3.1. Tworzenie tabeli w programie Access 2007 44
2.3.2. Typy danych w tabelach 47
2.3.3. Podstawowe właściwości pól tabeli 50
2.3.4. Pole kluczowe tabeli 55
2.3.5. Definiowanie relacji 57
2.3.6. Operacje na tabelach bazy danych 63
2.3.7. Zmiana sposobu prezentowania danych 65
2.3.8. Import, eksport, załączanie tabeli 70

2.4. Kwerendy — zapytania 74
2.4.1. Kwerendy wybierające 75
2.4.2. Kwerendy funkcjonalne — modyfikujące 88
2.4.3. Język zapytań SQL 95

2.5. Formularze 103
2.5.1. Autoformularze 103
2.5.2. Projektowanie formularzy 104

2.6. Raporty 108
2.7. Makropolecenia 110
2.8. Ochrona i bezpieczeństwo danych 112
2.8.1. Ochrona bazy danych hasłem 113
2.8.2. Kodowanie i dekodowanie bazy danych 114
2.9. Zadania 114

Rozdział 3. Systemy operacyjne i sieci komputerowe 117

3.1. Budowa komputera 117
3.2. BIOS i rozruch komputera 122
3.3. System operacyjny 124
3.3.1. Budowa systemu operacyjnego 124
3.3.2. Rola systemu operacyjnego 126
3.3.3. Maszyna wirtualna 131
3.3.4. Zadania 142
3.4. Sieci komputerowe 143
3.4.1. Podział sieci komputerowych 144
3.4.2. Model OSI 157
3.4.3. Model protokołu TCP/IP 162
3.4.4. Adres IP 164
3.4.5. Podstawowe rodziny protokołów 177
3.4.6. Obsługa i konfiguracja sieci w Windows 182
3.4.7. Ogólne zasady administrowania siecią komputerową w architekturze klient-serwer 182
3.4.8. Bezpieczeństwo sieci 183
3.4.9. Zadania 191
3.5. Podstawy tworzenia stron WWW 192
3.5.1. Podstawowe elementy usługi WWW 193
3.5.2. Projektowanie witryny WWW 193
3.5.3. Popularne formaty hipertekstowe w sieci WWW 193
3.5.4. Struktura dokumentu 196
3.5.5. Kaskadowe arkusze stylów 202
3.5.6. Publikowanie witryny w sieci WWW 206
3.5.7. Systemy zarządzania treścią 207
3.5.8. Czym jest PHP? 210
3.5.9. Zadania 221

Rozdział 4. Multimedia i grafika komputerowa 225

4.1. Grafika komputerowa 225
4.1.1. Grafika rastrowa 226
4.1.2. Rozdzielczość a wymiary obrazka 226
4.1.3. Reprezentacja obrazu w komputerze 230
4.1.4. Istota obrazu cyfrowego 239
4.1.5. Cyfrowy obraz ze skanera i cyfrowego aparatu fotograficznego 240
4.1.6. Kompresja stratna i bezstratna 240
4.1.7. Formaty plików bitmapowych 247
4.1.8. Grafika na stronach WWW 249
4.1.9. Grafika wektorowa 253
4.1.10. Grafika bitmapowa a wektorowa 257
4.1.11. Grafika trójwymiarowa — 3D 258
4.1.12. Zadania 266

4.2. Dźwięk 269
4.2.1. Teoria dźwięku 269
4.2.2. Zapis dźwięku 271
4.2.3. Formaty plików dźwiękowych 274
4.2.4. Zadania 281

4.3. Wideo 281
4.3.1. Cyfrowy zapis filmu 282
4.3.2. Montaż filmu i jego publikacja 284

4.4. Zadania sprawdzające do rozdziału 4. 285

Rozdział 5. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki 291

5.1. Kierunki rozwoju informatyki i jej zastosowań 291
5.1.1. Krótkie kalendarium wydarzeń ważnych dla rozwoju komputeryzacji 292
5.1.2. Co przyniesie najbliższa przyszłość? 293
5.1.3. Zadania 299

5.2. Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki 300
5.2.1. Prawo autorskie 300
5.2.2. Rodzaje licencji programów komputerowych 300
5.2.3. Wolne licencje 301
5.2.4. Ustawa o prawie autorskim a szkolny referat 302
5.2.5. Zadania 303

Bibliografia 304

CD-ROM 305

Skorowidz 306

Wstęp

Technologia informacyjna i informatyka to dziedziny, których wykorzystanie i dostępność stale wzrastają, a tempo zachodzących w nich zmian jest nieporównywalne z rozwojem innych dyscyplin. Widać to również w zakresie edukacji informatycznej.

„Informatyka Europejczyka" to podręcznik do informatyki na poziomie rozszerzonym w szkołach ponadgimnazjalnych. Jego treści, struktura, duża liczba ćwiczeń i zadań pozwolą Ci na przygotowanie się do egzaminu maturalnego z tego przedmiotu. Podręcznik składa się z dwóch części.

W pierwszej zawarty jest materiał związany wyłącznie z algorytmiką i programowaniem. Są w niej przedstawione metody i techniki programowania, algorytmy klasyczne wskazane w podstawie programowej oraz podstawy programowania w języku C++. Układ materiału tej części podręcznika, kolejność wprowadzanych zagadnień, dobór przykładów i zadań stanowić będą wsparcie zarówno dla Ciebie, jak i dla nauczyciela. Praca z tą częścią podręcznika powinna pozwolić Ci na uzyskanie dobrych wyników na egzaminie maturalnym w zakresie algorytmiki i programowania.

Pierwsza część podręcznika nie jest przeznaczona do realizacji w układzie liniowym — rozdział po rozdziale. Najpierw zapoznaj się z podstawowymi pojęciami algorytmicznymi i elementami języka C++, które zawarte są w rozdziałach pierwszym i trzecim. Następnie przejdź do metod algorytmicznych i zaawansowanych algorytmów, opisanych w rozdziale drugim. Rozdział trzeci, poświęcony programowaniu w C++, ma charakter pomocniczy. To swoiste kompendium elementów języka programowania powinno stanowić dla Ciebie skuteczną pomoc w pisaniu programów.

W drugiej, bogato ilustrowanej części podręcznika zapoznasz się z budową współczesnych komputerów. Dzięki wykorzystaniu maszyny wirtualnej zrozumiesz rolę i zadania systemów operacyjnych. Poznasz też architekturę sieci komputerowych i sposób działania protokołów sieciowych. Pracując z pakietem biurowym Microsoft Office 2007, nauczysz się wykorzystywać do wyszukiwania informacji zaawansowane funkcje bazodanowe arkusza kalkulacyjnego, poznasz relacyjną bazę danych i język zapytań SQL. Opanujesz podstawy pracy z pakietem biurowym OpenOffice.org — bezpłatnym oprogramowaniem zalecanym do stosowania w administracji i w biznesie. Następnie przyswoisz sobie zagadnienia związane z multimediami, z uwzględnieniem grafiki bitmapowej, wektorowej, 3D, dźwięku i filmu. Odkryjesz sposób działania algorytmów kompresji danych. Będziesz mieć także szansę na stworzenie i opublikowanie w internecie swojej strony WWW oraz na administrowanie własnym systemem zarządzania treścią. Opanujesz podstawy języka PHP. Zapoznasz się z zagrożeniami związanymi z dostępem do internetu. Przeanalizujesz treść Ustawy o prawie autorskim i prawach pokrewnych. Zastanowisz się nad głównymi kierunkami rozwoju informatyki i rozwiązań technologicznych, takich jak na przykład praca „w chmurze". Większość tych wiadomości oraz umiejętności zdobędziesz, wykonując praktyczne ćwiczenia.

Do podręcznika dołączone są płyty CD, które zawierają przykładowe programy w językach C++ i Pascal, realizację wybranych algorytmów za pomocą arkusza kalkulacyjnego, materiał uzupełniający dotyczący programowania obiektowego i samodzielnego tworzenia witryny opartej na systemie zarządzania treścią (CMS), opis wykonania ćwiczeń i zadań bazodanowych w programie OpenOffice.org, liczne pliki z rozszerzeniem treści podręcznika, dane do zadań, pliki potrzebne do realizacji ćwiczeń z podręcznika, wybrane przykładowe zadania z egzaminów maturalnych, niezbędne oprogramowanie, obrazy maszyn wirtualnych z różnymi systemami operacyjnymi, zbiór fotografii, filmów i nagrań dźwiękowych.

Przygotowując się do egzaminu maturalnego z informatyki, zapoznaj się z jak największą liczbą przykładów, wykonaj ćwiczenia zawarte w podręczniku i rozwiąż jak najwięcej zadań. Na szczególną uwagę zasługują zadania z egzaminów maturalnych dołączone na płytach CD. Wykonując je, zapoznasz się z formą zadań pojawiających się na egzaminie maturalnym.

Danuta Korman

i Grażyna Zawadzka

Rozdział 1.

BAZODANOWE OPERACJE ARKUSZA KALKULACYJNEGO

Na wcześniejszych etapach edukacji informatycznej nauczyłaś/nauczyłeś się już wprowadzać dane do arkusza kalkulacyjnego, wykonywać obliczenia za pomocą formuł z użyciem adresów względnych i bezwzględnych, wykorzystywać podstawowe funkcje do podsumowania danych i prezentować dane w postaci graficznej (na wykresach).

W tym rozdziale zapoznasz się z bazami danych, jakie możesz stworzyć w arkuszu kalkulacyjnym. Dowiesz się, jakie operacje możesz wykonywać na zgromadzonych danych i jakie narzędzia możesz do tego wykorzystać. Arkusz kalkulacyjny pozwala bowiem na porządkowanie danych, wybieranie danych spełniających zadane kryteria, dokonywanie podsumowań, tworzenie nowych zestawów danych, analizowanie danych za pomocą wbudowanych funkcji oraz prezentowanie graficznych zestawień. Wszystkie te operacje na zbiorach danych nazwiemy ogólnie operacjami bazodanowymi.

W tym rozdziale opiszemy też podstawowe operacje bazodanowe, korzystając z programu MS Excel w wersji 2007. Na płycie CD dołączonej do podręcznika, w dokumencie *r01 OpenOffice.pdf*, zamieszczone są rysunki będące wskazówkami do wykonania zawartych w podręczniku ćwiczeń i zadań w pakiecie biurowym OpenOffice.org 3.2, który można zainstalować zarówno w środowisku Windows, jak i Linux, na przykład Ubuntu.

1.1. Jednotabelowa baza danych — kartoteka

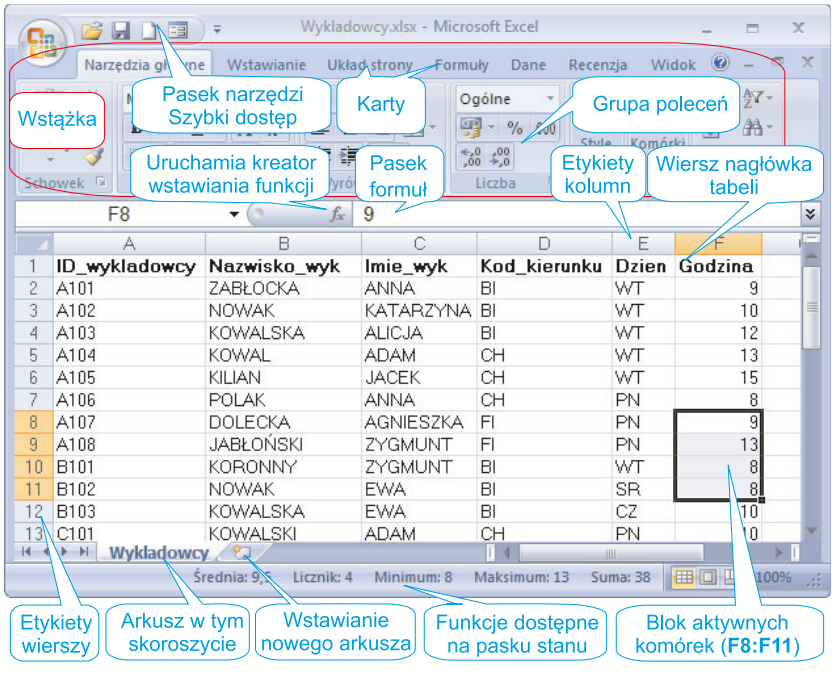
Definicja

Baza danych, której dane zgromadzone są w jednej tabeli, jest najprostszą — **kartotekową bazą danych**.

Kartotekowe bazy danych można tworzyć w dowolnym programie zarządzającym bazami danych, a także w arkuszach kalkulacyjnych, a nawet w prostych edytorach tekstu. Trzeba zdawać sobie sprawę z tego, że taka baza danych, czyli zbiór informacji przedstawionych w jednej tabeli, będzie dość skromną i sztywną strukturą, jednak taka forma gromadzenia danych często wystarcza do zdefiniowania niezbędnych zbiorów danych.

Rozpocznijmy od przypomnienia podstawowych pojęć i umiejętności. W tym celu na rysunku 1.1 zaprezentowany został opis otwartego skoroszytu danych w programie MS Excel 2007. Zamieszczone dane, zawarte w komórkach, mają określone typy i mogą być

w odpowiedni sposób formatowane (format komórki). Każda komórka ma swój **adres**, na przykład *B8* (**etykieta kolumny**, **etykieta wiersza** — nigdy odwrotnie). Do komórek arkusza można wprowadzać dane tekstowe, liczbowe (liczby, daty, godziny) oraz formuły. Formuła rozpoczyna się od znaku **=**. W formułach można stosować adresy względne (`=B8`), bezwzględne (`=$B$8`) i mieszane (`=B$8`) oraz funkcje wbudowane, których w Excelu, zainstalowanym w pełnej wersji z dodatkami, jest kilkaset.



	A	B	C	D	E	F
1	ID_wykladowcy	Nazwisko_wyk	Imie_wyk	Kod_kierunku	Dzien	Godzina
2	A101	ZABŁOCKA	ANNA	BI	WT	9
3	A102	NOWAK	KATARZYNA	BI	WT	10
4	A103	KOWALSKA	ALICJA	BI	WT	12
5	A104	KOWAL	ADAM	CH	WT	13
6	A105	KILIAN	JACEK	CH	WT	15
7	A106	POLAK	ANNA	CH	PN	8
8	A107	DOLECKA	AGNIESZKA	FI	PN	9
9	A108	JABŁOŃSKI	ZYGMUNT	FI	PN	13
10	B101	KORONNY	ZYGMUNT	BI	WT	8
11	B102	NOWAK	EWA	BI	SR	8
12	B103	KOWALSKA	EWA	BI	CZ	10
13	C101	KOWALSKI	ADAM	CH	PN	10

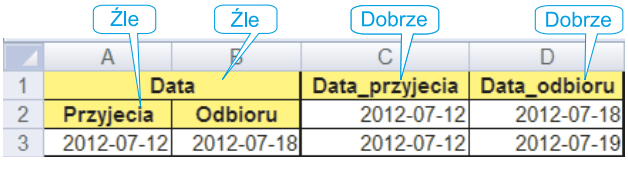
Wykladowcy

Średnia: 9,... Licznik: 4 Minimum: 8 Maksimum: 13 Suma: 38 100%

Rysunek 1.1. Skoroszyt w programie MS Excel 2007

Podstawowym zbiorem danych jest **tabela**. Aby można było korzystać z tabeli jak z bazy danych, musi ona spełniać kilka kryteriów:

- pierwszy wiersz tabeli powinien zawierać nagłówki tabeli (nazwy kolumn, nazwy pól);
- każda kolumna musi mieć niepowtarzalną nazwę;
- każda kolumna danych musi zawierać dane tego samego typu (na przykład w kolumnie *A* wszyscy mają identyfikator, a w kolumnie *F* godzinę);
- wewnątrz bazy nie należy zostawiać pustych kolumn ani wierszy. Poszczególne komórki mogą być puste (na przykład nie każdy ma telefon);
- komórki nie mogą być scalone, co ilustruje przykład na rysunku 1.2.



	A	B	C	D
1	Data		Data_przyjecia	Data_odbioru
2	Przyjecia	Odbioru	2012-07-12	2012-07-18
3	2012-07-12	2012-07-18	2012-07-12	2012-07-19

Rysunek 1.2. Przykład niepoprawnego i poprawnego nagłówka tabeli będącej bazą danych

Dobrym nawykiem jest unikanie w nazwach kolumn znaków diakrytycznych (charakterystycznych dla danego języka narodowego), interpunkcyjnych i spacji, co gwarantuje bezproblemowe eksportowanie danych do innych pakietów biurowych lub programów bazodanowych. Najlepiej będzie, jeśli zachowamy zasadę obowiązującą przy konstruowaniu nazw zmiennych w programowaniu, które budujemy z liter, cyfr i znaku podkreślenia.

Czy tabela pokazana na rysunku 1.1 może pełnić funkcję bazy danych w Excelu? Tak, ponieważ spełnia wszystkie wymogi z punktów wymienionych na poprzedniej stronie, dotyczących konstrukcji tabeli i sposobu gromadzenia w niej danych.

1.2. Podstawowe pojęcia

Najpierw przypomnijmy definicję bazy danych.

Definicja

Baza danych to zgromadzone dane, zapisane zgodnie z określonym porządkiem (strukturą) umożliwiającym łatwe ich wyszukiwanie.

W bazach danych możemy gromadzić dane z różnych dziedzin, od domowych wydatków, spisu książek czy płyt po bazę leków w aptece lub artykułów sieci sklepów. Dane te powinny być gromadzone w zorganizowany sposób, aby później można było łatwo je wyszukać, dokonać na ich podstawie obliczeń i zestawień. Wszelkie informacje dzielimy zatem na **elementarne pola**, na przykład rozdzielamy dane adresowe na poszczególne składowe adresu.

Definicja

Pole jest elementarnym składnikiem każdego wprowadzonego do bazy obiektu, opisującym jedną jego właściwość (atrybut).

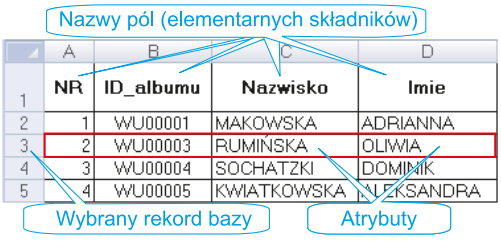
Każdy obiekt musi mieć przynajmniej jeden atrybut. Sam obiekt w języku bazodanowym nazywamy **rekordem**. W jednym wierszu tabeli arkusza kalkulacyjnego znajdują się atrybuty jednego rekordu danych.

Definicja

Rekord to zbiór informacji opisujących jeden obiekt bazy danych.

Na rysunku 1.3 przedstawiono tabelę z czterema rekordami — dane czterech osób, dla których zgromadzono cztery atrybuty (pola): *NR*, *ID_albumu*, *Nazwisko* i *Imie*. Zatem:

- każdy wiersz tabeli zawiera jeden rekord;
- w każdej kolumnie podajemy jeden atrybut (elementarną informację) poszczególnych rekordów.

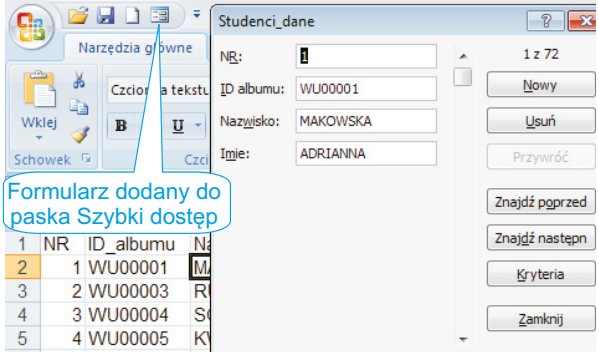


Rysunek 1.3. Rekordy i pola w przykładowej tabeli bazy danych

Pojęcia „pole” i „rekord” lepiej obrazuje widok formularza bazy danych. W tabeli widać wiele rekordów jednocześnie, natomiast formularz pokazuje jeden rekord i jego pola (atrybuty), jak na rysunku 1.4. Wskaż na formularzu danych nazwy pól i pola (atrybuty) danego rekordu. Odszukaj na rysunku liczbę wpisanych do bazy rekordów oraz informację o tym, który z rekordów jest na nim wyświetlany.

Wskazówka

Co się stało z formularzem danych w programie MS Excel w wersji 2007? Przycisk *Formularz* nie został uwzględniony na Wstążce, ale nadal można z niego korzystać — wystarczy dodać przycisk *Formularz* do paska narzędzi *Szybki dostęp*. W tym celu kliknij przycisk *Dostosuj pasek narzędzi Szybki dostęp* (strzałka obok paska narzędzi *Szybki dostęp*), a następnie z listy wybierz *Więcej poleceń…*. W polu *Wybierz polecenia z:* wskaż pozycję *Wszystkie polecenia*. Z rozwiniętej listy wybierz *Formularz…*, a następnie kliknij przycisk *Dodaj* >>.



Rysunek 1.4. Formularz do tabeli z rysunku 1.3

1.3. Tworzenie nowej bazy i import danych

Pracownicy pewnej firmy doszli do wniosku, że należałoby rejestrować przez jakiś czas sprzedaż towarów. Chcieli wiedzieć, na jakie towary jest największy popyt i który sprzedawca może pochwalić się najlepszymi wynikami pracy.

Nie potrzeba do tego pomocy programistów i wydatków na rozwój systemu komputerowego, ponieważ rejestracja sprzedawanych towarów może być wykonywana za pomocą bazy danych Excela. Twój stworzony domowym sposobem „system” będzie miał przyjazny dla użytkownika[1] formularz do wprowadzania danych i bez trudu znajdziesz w nim konkretne informacje. Gdy będzie potrzebny wyciąg z danych, łatwo utworzysz raport — podsumowanie danych.

Ćwiczenie 1.1. Załóż nowy skoroszyt, wpisz nazwy pól i wprowadź dane dotyczące **trzech pierwszych transakcji**, zgodnie z rysunkiem 1.5. Można powiedzieć, że baza jest

[1] Mamy świadomość, że w podręczniku należałoby używać form rodzaju męskiego i żeńskiego, ale dla potrzeb informatyki, ze względu na jednoznaczność poleceń w stosowanych językach i logikę wywodu, stosując formę męską (na przykład: użytkownicy, studenci, administratorzy, pacjenci, optycy, projektanci, uczniowie), mamy oczywiście na myśli zarówno kobiety, jak i mężczyzn.

gotowa. Ta tabela jest bowiem bazą danych gromadzącą informacje o liczbie artykułów sprzedanych w danym dniu przez każdego z trzech sprzedawców.

Do wpisywania danych można używać formularza lub wprowadzać je bezpośrednio w tabeli.

	A	B	C	D
1	**Data**	**Produkt**	**Liczba**	**Sprzedawca**
2	2012-08-11	HERBATA	2	WIŚNIEWSKI
3	2012-08-11	HERBATA	4	MALINOWSKI
4	2012-08-11	HERBATA	5	KOZŁOWSKI
5	2012-08-11	KAWA	2	KOZŁOWSKI
6	2012-08-11	KAWA	3	KOZŁOWSKI
7	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
8	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
9	2012-08-12	KAWA	5	MALINOWSKI
10	2012-08-14	HERBATA	5	MALINOWSKI
11	2012-08-14	KAWA	3	MALINOWSKI

Rysunek 1.5. Projekt bazy danych do ćwiczenia 1.1

Ćwiczenie 1.2. Korzystając z formularza danych, wprowadź dane pozostałych siedmiu transakcji.

W tym celu koniecznie umieść kursor w polu rekordu tabeli, na przykład w komórce *A1*. Z paska *Szybki dostęp* wybierz *Formularz*. Automatyczny formularz, zawierający wszystkie pola, pozwala na przeglądanie pojedynczych rekordów bazy, edycję już wprowadzonych danych oraz usuwanie treści poszczególnych pól (atrybutów). Kliknij przycisk *Nowy* i dodaj dane czwartego rekordu (czwartego wpisu do bazy). Klawisze *Tab* i *Shift+Tab* pozwalają na przełączanie pomiędzy polami rekordu. W ten sposób szybko wypełnisz cały formularz. Pamiętaj o kliknięciu przycisku *Nowy* po każdym wypełnieniu formularza. Zapisz utworzoną bazę danych (skoroszyt) pod nazwą *sklep*.

Dane do arkusza kalkulacyjnego można również importować z pliku tekstowego, w którym poszczególne pola rekordów są oddzielone znakiem separacji, na przykład średnikiem, tabulatorem lub spacją. Import danych z pliku tekstowego jest wykonywany z wykorzystaniem polecenia *Otwórz*, które po wyborze typu plików i wskazaniu zbioru uruchamia *Kreator importu tekstu*.

Ćwiczenie 1.3. Na dysku CD zgromadzone zostały dane, które będą wykorzystywane w następnym rozdziale podręcznika. Otwórz w Excelu wszystkie dane zgromadzone w plikach tekstowych, kolejno: *studenci_dane.txt*, *studenci_uczelnia.txt*, *studenci_oceny.txt*, i skopiuj wszystkie dane studiujących osób do jednego arkusza danych. Powstały skoroszyt zapisz pod nazwą *studenci_import*. Będzie on potrzebny do ćwiczeń końcowych.

Podczas wykonywania tego ćwiczenia zwróć uwagę na fakt tworzenia jednego arkusza z kilku plików tekstowych. Przed importem danych warto zobaczyć zawartość pliku, aby sprawdzić, w jaki sposób dane zostały rozdzielone. Po imporcie danych koniecznie zwróć uwagę na powtarzanie się niektórych pól w poszczególnych tabelach. W wynikowej tabeli pole *ID_albumu* powinno wystąpić tylko raz.

W bazie danych należy gromadzić dane tak, aby uniknąć ich dublowania. Dane, które można obliczyć, nie powinny stanowić bazy (źródła) danych.

Ciekawostka

CSV (ang. *Comma Separated Values* — wartości rozdzielone przecinkiem) to format przechowywania danych w plikach tekstowych. Wartości są rozdzielane przecinkami (wbrew nazwie formatu separatorem może być również tabulator, średnik lub inny znak), a rekordy znakiem nowej linii — znakiem nowego wiersza **CRLF** (ang. *carriage return + line feed* — powrót karetki + nowa linia). Pliki *CSV* stosowane są często jako uniwersalna platforma wymiany danych między różnymi systemami, w szczególności między arkuszami kalkulacyjnymi.

1.4. Wyszukiwanie i organizowanie danych

1.4.1. Wyszukiwanie z wykorzystaniem formularza

Formularz może również ułatwić i przyspieszyć wyszukiwanie danych. Docenisz tę funkcję, kiedy będzie trzeba znaleźć jakieś dane w bazie zawierającej setki rekordów.

Ćwiczenie 1.4. Spróbuj odszukać, co sprzedał Malinowski 14 sierpnia.

Otwórz formularz. Wciśnij przycisk *Kryteria* i wpisz w polu *Data*: `2012-08-14`, a w polu *Sprzedawca* wpisz `Malinowski`. Kliknij przycisk *Znajdź następny*. Wyświetlony został rekord spełniający kryteria wyszukiwania. Jeśli Malinowski sprzedał tego dnia więcej towarów, możesz obejrzeć wszystkie jego transakcje, klikając przyciski *Znajdź następny* i *Znajdź poprzedni*. Jeżeli po kliknięciu przycisku usłyszysz krótki dźwięk z głośnika systemowego, będzie to oznaczało, że nie ma więcej rekordów spełniających kryteria wyszukiwania. (Spróbuj również przeglądać rekordy za pomocą klawiszy strzałek ↑ i ↓). Na koniec zamknij formularz przyciskiem *Zamknij*.

Ciekawostka

Czasami, jak w przypadku tej aplikacji bazodanowej, na przyciskach wyświetlone są niepełne napisy: *Znajdź następ* i *Znajdź poprzed*. Zdarza się, że napis nie mieści się na przycisku. Prawdopodobnie plik translacji programu zawiera pełny ciąg, ale układ kompozycji okna (ang. *layout*) wymusza taką szerokość przycisku, że widoczny jest tylko fragment tekstu.

1.4.2. Wyszukiwanie informacji za pomocą autofiltrowania

Dla uzyskania przejrzystości dobrze byłoby wyodrębnić niektóre dane z listy tabeli. Ułatwiłoby to znalezienie odpowiedzi na pytania: Co zostało sprzedane 14 sierpnia? Co sprzedał Malinowski? Ile razy sprzedano herbatę? Co sprzedał Malinowski 14 sierpnia? Ile razy sprzedaż wynosiła co najmniej 5 sztuk? Jakiego towaru sprzedano najwięcej? Który ze sprzedawców go sprzedał i kiedy to było?

Ćwiczenie 1.5. Wyszukaj dane zawarte w pliku *sklep.xls*, na podstawie których można będzie odpowiedzieć na zadane powyżej pytania. W tym celu posłuż się *filtrowaniem*.

1. Jakie artykuły sprzedano 14 sierpnia?

Umieść kursor w dowolnym polu listy, w którym wpisano dane, i wybierz z karty *Dane* polecenie *Filtruj*. Przy wszystkich tytułach kolumn pojawiły się teraz strzałki. Kliknij tę, która znajduje się przy polu *Data*. Lista dostępnych filtrów zależy od typu danych zawartych w kolumnie i domyślnie jest ustawiona na filtrowanie typu *Równa się…*. Zatem dane typu „data" mają do wyboru *Filtry dat*. Na liście, która zostanie wyświetlona, pozostaw zaznaczenie przy dacie 14 sierpnia. Zwróć uwagę, że w miejscu strzałki jest obecnie symbol filtru i miniatura strzałki, a numery przefiltrowanych rekordów mają kolor niebieski. Oznacza to, że w kolumnie *Data* został uaktywniony filtr.

Wskazówka

Pamiętaj, że filtrowanie powoduje ukrycie (a nie usunięcie) rekordów, które nie spełniają podanych kryteriów.

Kliknij ponownie strzałkę w kolumnie *Data*. Wybierz z listy pozycję *Wyczyść filtr z „Data"*. Widzisz ponownie wszystkie rekordy i strzałkę bez znaku filtru w komórce *A1*.

2. Co sprzedał Malinowski?

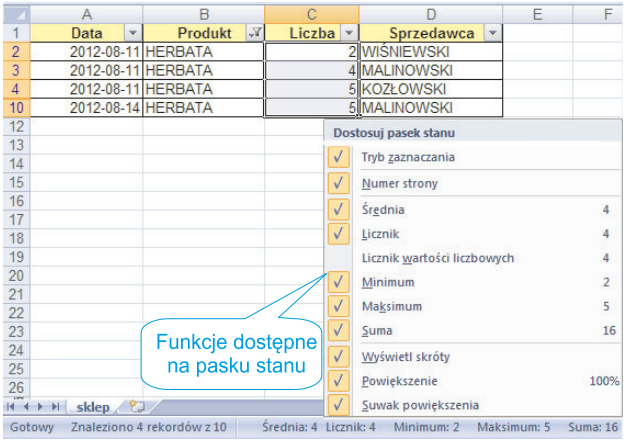
To zadanie wykonaj analogicznie do poprzedniego.

3. Ile razy sprzedano herbatę?

Kliknij strzałkę w kolumnie *Produkt* i z listy wybierz pozycję *HERBATA*. Zobaczysz wszystkie transakcje, podczas których sprzedano herbatę. Jednak w jaki sposób je zliczyć? W naszej maleńkiej bazie jest tylko kilka odnotowanych transakcji sprzedaży herbaty i można je po prostu policzyć. Najprościej jednak — bez wprowadzania żadnych dodatkowych wierszy z podliczeniami — odczytać informację na pasku stanu *Znaleziono 4 rekordów z 10* lub skorzystać z funkcji *Licznik*, która zliczy wystąpienia. W tym celu przeciągnij myszą wszystkie dane w polu *Produkt*, gdzie występuje teraz sama herbata. Funkcja *Licznik* natychmiast zliczyła zaznaczone produkty i na pasku stanu pojawiła się wartość obrazująca liczbę wystąpień herbaty *Licznik: 4*. Inne funkcje pojawią się, gdy zaznaczysz dane liczbowe, na przykład wszystkie wartości w polu *Liczba*, jak pokazano na rysunku 1.6. Aby zmienić funkcje, które mają wyświetlać wyniki, kliknij prawym przyciskiem myszy pasek stanu i wybierz odpowiednie funkcje z listy.

Wśród dostępnych funkcji są funkcje dotyczące głównie danych liczbowych. Należą do nich: *Średnia*, *Licznik* (zlicza wystąpienia zarówno danych liczbowych, czyli liczb, dat, godzin, jak i danych nieliczbowych), *Licznik wartości liczbowych*, *Minimum*, *Maksimum* i *Suma*.

Sprawdź teraz dodatkową informację: ile opakowań herbaty sprzedano? W tym celu skorzystaj z funkcji *Suma*, która po zaznaczeniu pól z liczbą sprzedaży zliczy wszystkie sprzedane opakowania herbaty, tworząc podsumowanie, z którego dowiesz się, jaka jest skala sprzedaży (rysunek 1.6).

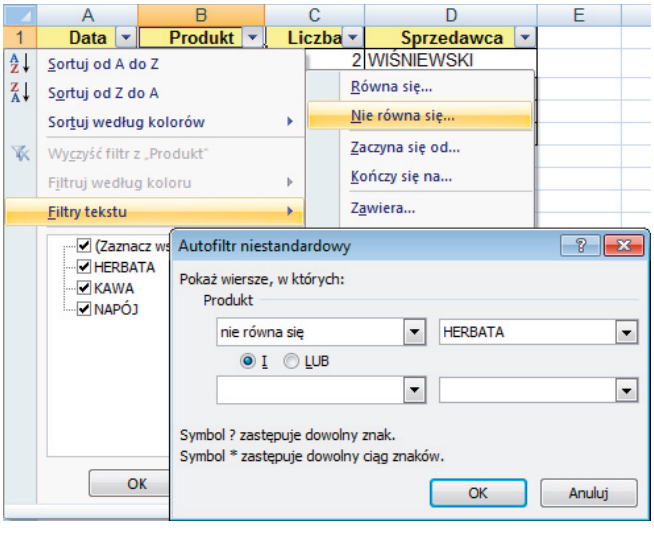


Rysunek 1.6. Funkcje dla zaznaczonych komórek dostępne na pasku stanu

Sumaryczne wyniki sprzedaży herbaty są już znane. Sprawdź teraz dodatkową informację: ile opakowań pozostałych produktów sprzedano? W tym celu ustaw filtrowanie według pola *Produkt* z wykluczeniem herbaty. Filtrowanie można wykonać na dwa sposoby. Pierwszy sposób polega na zaznaczeniu wszystkich produktów oprócz herbaty. Drugi sposób, pokazany na rysunku 1.7, polega na zmianie sposobu filtrowania w opcji *Filtry tekstu*, z których należy wybrać polecenie *Nie równa się…*. W oknie *Autofiltr niestandardowy* wybierz z listy produkt *HERBATA*. Dla wybranych rekordów odczytaj z paska stanu wartość funkcji *Suma*, która po zaznaczeniu pól z liczbą transakcji sprzedaży zliczy wszystkie sprzedane kawy i napoje, tworząc podsumowanie, z którego dowiesz się, jaka jest skala sprzedaży.

Wskazówka

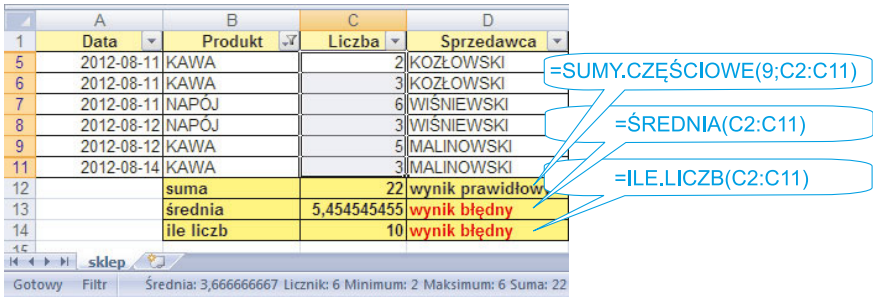
Naciśnięcie kombinacji klawiszy *Ctrl+A* spowoduje, że — mimo zastosowania filtrowania danych — zaznaczone zostaną wszystkie rekordy, nawet te niewidoczne, niespełniające zadanego kryterium.



Rysunek 1.7. Filtrowanie z wykluczeniem wyboru

Wskazówka

Uwaga! Dodanie komórki, w której zastosujemy podsumowania dla przefiltrowanych danych, może zwrócić błędne wyniki. Zostało to zilustrowane na rysunku 1.8 — w trzech komórkach *D12:D14* wprowadzono formuły zawierające standardowe funkcje: obliczania sumy, wartości średniej i zliczania wystąpienia liczb. Zakresy będące parametrami tych funkcji uwzględniają również rekordy ukryte w widoku filtrowania. Stąd nieprawidłowe obliczenia. W wynikach podawanych na pasku stanu tego błędu nie było.



Rysunek 1.8. Zakresy w parametrach funkcji podczas filtrowania danych

Powróć do wyświetlania wszystkich rekordów tabeli. Najszybszym wyłączeniem zastosowanego wcześniej filtrowania jest wybór polecenia *Wyczyść* z grupy *Sortowanie i filtrowanie* na karcie *Dane*. Możesz również usunąć filtrowanie opisanym już wcześniej poleceniem *Wyczyść filtr z „Produkt”*.

4. Co sprzedał Malinowski 11 sierpnia?

Wskazówka

Na przefiltrowanych już danych można stosować kolejne filtry.

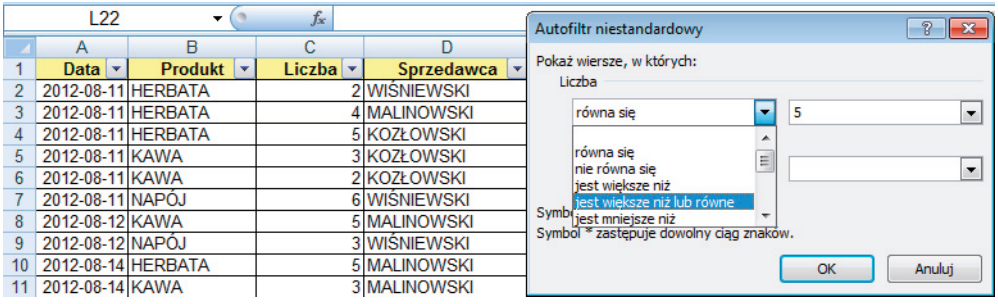
Kliknij strzałkę w kolumnie *Data* i z listy wybierz pozycję z poszukiwaną datą. Naciśnij strzałkę w kolumnie *Sprzedawca* i wybierz nazwisko *MALINOWSKI*. Gotowe.

Po obejrzeniu rezultatu powróć do wyświetlania wszystkich rekordów tabeli (*Wyczyść*).

5. Ile razy sprzedaż wynosiła co najmniej 5 sztuk?

Przyjrzyjmy się teraz wszystkim wartościom sprzedaży. Aby wyselekcjonować tylko te transakcje, których było co najmniej 5, należy skorzystać z kryteriów niestandardowych. Kliknij więc strzałkę w kolumnie *Liczba* i wybierz z listy pozycję *Filtry liczb*. Następnie z listy filtrów dostępnych dla danych liczbowych wybierz *Większe niż lub równe…*.

Zostanie otwarte okno dialogowe *Autofiltr niestandardowy* (to samo okno pojawi się, gdy z dostępnych filtrów wybierzesz *Filtr niestandardowy…*). Wprowadź kryterium wyboru dla pola *Liczba*. Rozwiń listę dostępnych operatorów porównań i wybierz *jest większe niż lub równe*, a w oknie dialogowym wpisz lub wybierz wartość *5* (jak na rysunku 1.9).



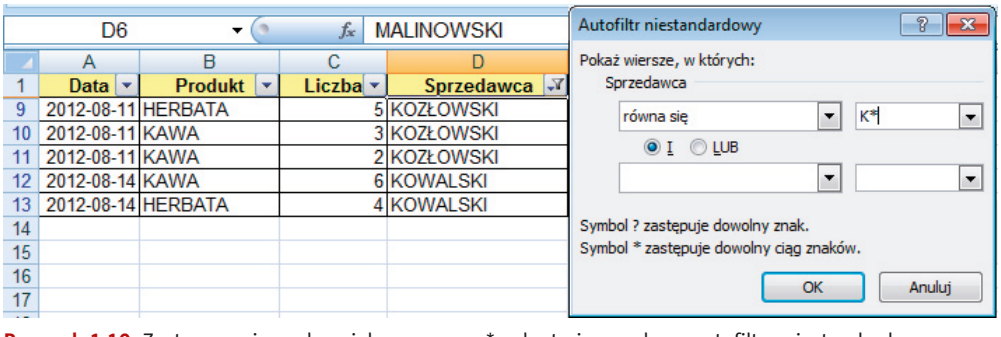
	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
2	2012-08-11	HERBATA	2	WIŚNIEWSKI
3	2012-08-11	HERBATA	4	MALINOWSKI
4	2012-08-11	HERBATA	5	KOZŁOWSKI
5	2012-08-11	KAWA	3	KOZŁOWSKI
6	2012-08-11	KAWA	2	KOZŁOWSKI
7	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
8	2012-08-12	KAWA	5	MALINOWSKI
9	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
10	2012-08-14	HERBATA	5	MALINOWSKI
11	2012-08-14	KAWA	3	MALINOWSKI

Rysunek 1.9. Autofiltr niestandardowy

Można w nim zdefiniować dwa kryteria (tylko dwa) i połączyć je spójnikiem koniunkcji *I* albo alternatywy *LUB*. Po zatwierdzeniu zostaną wyświetlone wszystkie przykłady dziennej sprzedaży wynoszącej co najmniej 5 sztuk. Teraz zlicz wystąpienia.

Autofiltr niestandardowy, w którym możesz wybrać dwa kryteria dla danego pola, umożliwia zastosowanie znaków wieloznacznych `?` oraz `*`, które zastępują jeden lub wiele znaków. Przykładowo polecenie `W*` spowoduje wybranie danych tekstowych zaczynających się na literę *W*.

Dodaj do wykazu jeszcze jednego sprzedawcę, Kowalskiego, który w dniu 14 sierpnia sprzedał 6 kaw i 3 herbaty. Teraz za pomocą znaku `*` wyszukaj wszystkie transakcje sprzedawców, których nazwisko rozpoczyna się na *K*. Gdy w kryterium autofiltru niestandardowego pola *Sprzedawca* wybierzesz opcję *równa się* i podasz wartość *K**, na liście zostaną wyświetlone rekordy z nazwiskami zaczynającymi się na literę *K* (rysunek 1.10). Natomiast gdy chcesz wyświetlić wszystkie nazwy produktów kończące się na literę `A`, to po wyborze *równa się* i porównaniu z wartością `*A` zobaczysz tę samą listę nazwisk sprzedawców, co po wyborze kryterium *kończy się na* i podaniu wartości `A`. Znaku `?` użyjesz do zastąpienia jednego znaku. Wyszukaj wszystkich sprzedawców, których nazwisko zawiera literę *O* na drugiej pozycji. W polu kryterium dla pola *Sprzedawca* wpisz `?O*`.



	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
9	2012-08-11	HERBATA	5	KOZŁOWSKI
10	2012-08-11	KAWA	3	KOZŁOWSKI
11	2012-08-11	KAWA	2	KOZŁOWSKI
12	2012-08-14	KAWA	6	KOWALSKI
13	2012-08-14	HERBATA	4	KOWALSKI
14				
15				
16				
17				

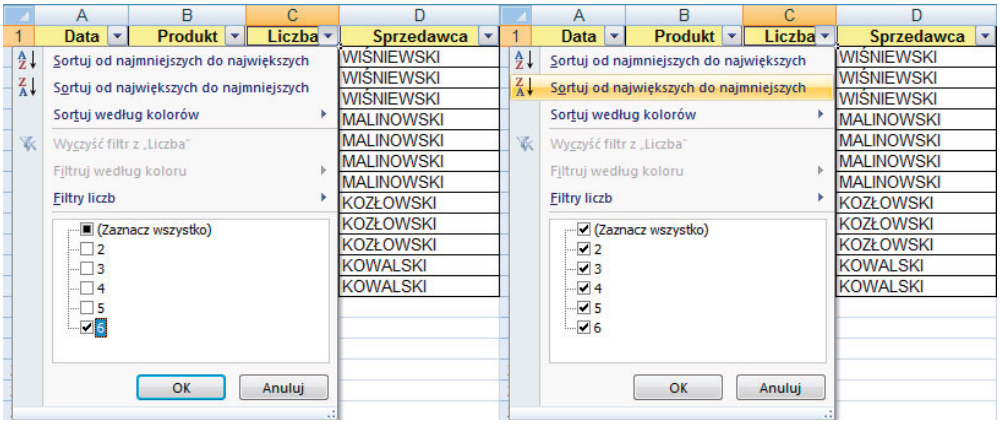
Rysunek 1.10. Zastosowanie znaku wieloznacznego * w kryterium wyboru autofiltru niestandardowego

6. Jakiego towaru sprzedano najwięcej, który ze sprzedawców go sprzedał i kiedy to było?

Oczywiście wybór będzie dotyczył pola *Liczba*. Na to pytanie można odpowiedzieć, stosując filtrowanie na trzy różne sposoby:

- wybór maksymalnej wartości z listy,
- sortowanie pola,
- wybór opcji *Pierwsze 10…*.

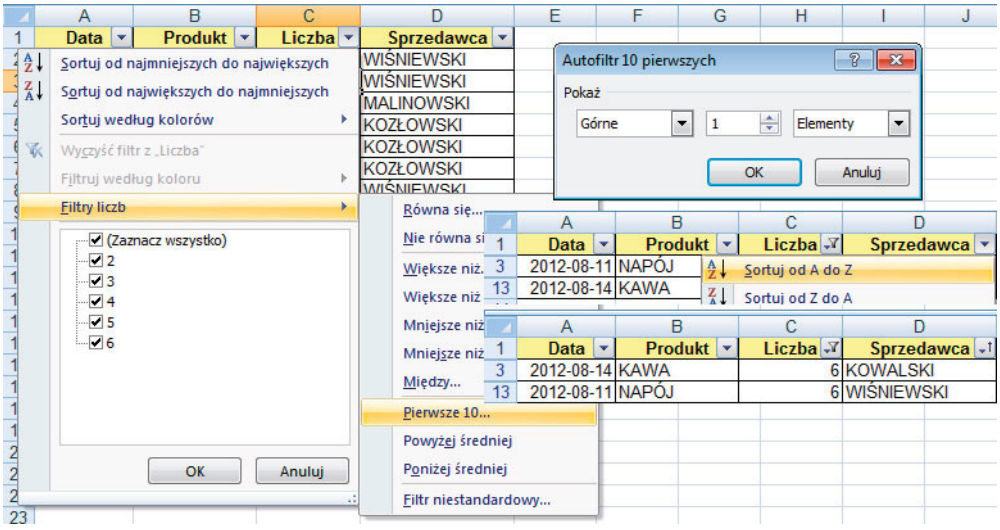
Na rysunku 1.11 pokazane zostały dwa pierwsze sposoby. Po wybraniu największej wartości z listy, czyli *6*, wyświetlone zostaną tylko te rekordy, które przechowują największą wartość sprzedaży. Po wybraniu sortowania malejącego na początku listy sprzedaży pojawi się żądany rekord (lub pojawią, gdy jest więcej rekordów o maksymalnej liczbie sprzedaży).



Rysunek 1.11. Wybór maksimum z listy rekordów

Trzeci wymieniony sposób wymaga krótkiego wyjaśnienia. Na rysunku 1.12 pokazano opcję filtrowania *Pierwsze 10…*. Tę metodę można stosować tylko do danych liczbowych (liczby, daty, godziny), więc można użyć jej do kolumny (pola) *Liczba*. W oknie dialogowym *Autofiltr 10 pierwszych* masz dwie możliwości wyboru wartości: mogą to być wartości *Górne* (maksymalne) i *Dolne* (minimalne). *10 pierwszych* to tylko nazwa, która wcale nie oznacza, że zawsze będziesz wybierać dziesiątkę — może to być trójka najlepszych (najgorszych) rekordów albo tak jak w naszym ćwiczeniu — jeden najlepszy rekord. Ponieważ nie jest jednoznacznie wiadome, który rekord jest najlepszy, po ustawieniu kryteriów *Pokaż* (`Górne`, `1`, `Elementy`) w wykazie danych będzie figurowało dwóch sprzedawców, którzy byli rekordzistami sprzedaży jednego artykułu w danym dniu. Na utworzonym zestawie zastosuj filtr sortowania według nazwiska sprzedawców — taki wykaz jest bardziej sprawiedliwy. Przyzwyczailiśmy się do tego, że Wiśniewski jest za Kowalskim, bo jego nazwisko zaczyna się na literę *W*.

Aby zyskać wprawę w posługiwaniu się kryterium wyboru *Pierwsze 10…*, powróć do widoku wszystkich danych, a następnie wybierz trzy najlepsze albo trzy najgorsze wyniki sprzedaży. Obserwuj tabelę wynikową. Sprawdź, jak działa kryterium *Pierwsze*



Rysunek 1.12. Autofiltr Pierwsze 10... jako kryterium wyboru

10… na datach. Czy w celu wyboru dwóch najstarszych wolumenów sprzedaży wybierzesz daty z początku, czy z końca listy? A jak działa kryterium *10 procent* zamiast *10 elementów*? Zmień opcje w oknie dialogowym i obserwuj wybory *10%*, *30%*, *50%*. To narzędzie jest naprawdę przydatne.

Na koniec powróć do widoku wszystkich rekordów.

Wskazówka

Jeśli chcesz zrezygnować z filtrowania, możesz je po prostu wyłączyć. W tym celu na karcie *Dane* wyłącz polecenie *Filtruj*.

1.4.3. Wyszukiwanie informacji za pomocą filtru zaawansowanego

Jeżeli chcemy zastosować złożone kryteria wyboru, będące alternatywą lub koniunkcją więcej niż dwóch warunków wyboru dla jednego pola tabeli, nie możemy skorzystać z autofiltrowania. Możemy natomiast zastosować tzw. *filtr zaawansowany*, którego użycie wymaga wcześniejszego przygotowania *tabeli kryteriów*.

Następne ćwiczenia wykonamy na kopii arkusza z danymi o sprzedaży. Przed wykonaniem kopii zmień domyślną nazwę arkusza z *Arkusz1* na *sklep*. Następnie na karcie arkusza *sklep* wybierz polecenie *Przenieś lub kopiuj…*. W oknie dialogowym zaznacz pole *Utwórz kopię* i umieść kopię za arkuszem *sklep*. Zmień nazwę utworzonej kopii arkusza *sklep (2)* na *filtry* i przejdź do tej kopii.

Wskazówka

Dobrym zwyczajem jest kopiowanie danych (arkusza lub skoroszytu) w celu próbnego wykonania działań matematycznych czy statystycznych — w razie złych efektów nie stracimy oryginalnych danych.

Ćwiczenie 1.6. Wyświetl w arkuszu wszystkie transakcje trzech sprzedawców: Kowalskiego, Malinowskiego i Wiśniewskiego. Listę posortuj alfabetycznie według kolumny *Produkt*.

Najpierw zajmij się posortowaniem listy. Do tego celu użyj filtrowania, które posortuje rosnąco wszystkie rekordy według kolumny *Produkt* przed ich wybraniem. Zastosowanie kolejnych filtrów zachowa nadany porządek. W treści ćwiczenia wystąpiły trzy kryteria wyboru dla jednego pola *Sprzedawca*, więc skorzystaj z polecenia *Filtruj*, a następnie *Zaawansowane…*, wymagającego przygotowania specjalnej *tabeli kryteriów*.

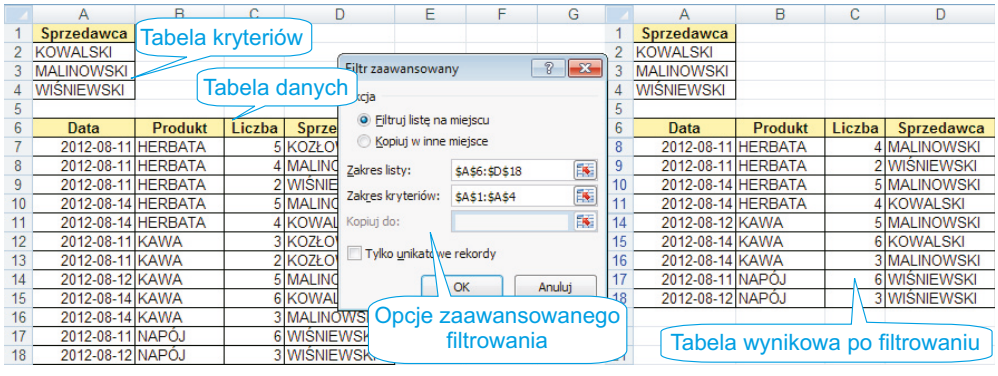
Przygotuj miejsce nad tabelą z danymi na tabelę kryteriów wyboru. Zaznacz pięć pierwszych wierszy (na górze tabeli), przeciągając po etykietach wierszy, i na karcie *Narzędzia główne* wybierz z grupy *Komórki* polecenie *Wstaw*. Powyżej zaznaczonych komórek pojawi się pięć pustych wierszy. W pierwszym z nich zamieść nagłówki kolumn (nazwy pól), według których będziesz filtrować dane. W tym ćwiczeniu jest to jedna kolumna (jedno pole) *Sprzedawca* — skopiuj je. Poniżej zamieść wszystkie trzy nazwiska, które mają być kryteriami wyboru.

Jednokolumnową tabelę kryteriów wyboru należy rozumieć jako alternatywę trzech kryteriów. Czytaj to następująco: wyświetl rekordy, w których *(Sprzedawca="KOWALSKI") lub (Sprzedawca="MALINOWSKI") lub (Sprzedawca="WIŚNIEWSKI")*.

Wskazówka

Tabelę kryteriów zawsze umieszczaj w wierszach wstawionych powyżej tabeli z danymi.

Po utworzeniu kryteriów uruchom filtr zaawansowany i w jego oknie dialogowym wskaż dwa zakresy: *Zakres listy* (cała tabela z danymi i nazwami pól) oraz *Zakres kryteriów* (tabela kryteriów wyboru), jak na rysunku 1.13. Nie ma konieczności zaznaczenia pola *Tylko unikatowe rekordy*, ponieważ w naszej bazie nie wystąpiły powtórzenia wpisanych danych.



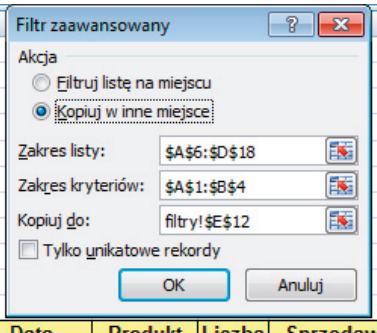
Rysunek 1.13. Tworzenie tabeli kryteriów do zaawansowanego filtrowania danych zawartych w tabeli

Wybierz opcję *Filtruj listę na miejscu*, aby ukryć dane niespełniające kryteriów wyboru. Po zatwierdzeniu opcji i zakresów filtru otrzymany zestaw danych zastąpi istniejące dane. Po obejrzeniu wyników filtrowania, na karcie *Dane/Sortowanie i filtrowanie* wybierz *Wyczyść*, co przywróci widok wszystkich rekordów bazy.

Ćwiczenie 1.7. Wyświetl wszystkie transakcje sprzedaży co najmniej czterech sztuk artykułów, zrealizowane przez trzech sprzedawców: Kowalskiego, Malinowskiego i Wiśniewskiego.

Do ćwiczenia 1.6 dodaj kryterium liczby sprzedanych artykułów. Do istniejącej tabeli kryteriów wyboru dołóż drugie pole, *Liczba*, i wpisz kryterium dla każdego sprzedawcy: *>=4*, jak na rysunku 1.14. Należy je czytać następująco: wybierz rekordy, w których *(Sprzedawca="KOWALSKI" i Liczba>=4)* lub *(Sprzedawca="MALINOWSKI" i Liczba>=4)* lub *(Sprzedawca="WIŚNIEWSKI" i Liczba>=4)*. Wiersze poziome w tabeli kryteriów wyboru oznaczają koniunkcję (*i*) kryteriów. Tym razem w sekcji *Akcja* wybierz opcję *Kopiuj w inne miejsce*, co uaktywnia pole *Kopiuj do*. Należy w nie wpisać adres komórki arkusza będącej lewym górnym wierzchołkiem nowo utworzonej listy, która jest przefiltrowanym zestawem danych.

	A	B	C	D
1	**Sprzedawca**	**Liczba**		
2	KOWALSKI	>=4		
3	MALINOWSKI	>=4		
4	WIŚNIEWSKI	>=4		
5				
6	**Data**	**Produkt**	**Liczba**	**Sprzedawca**
7	2012-08-11	HERBATA	5	KOZŁOWSKI
8	2012-08-11	HERBATA	4	MALINOWSKI
9	2012-08-11	HERBATA	2	WIŚNIEWSKI
10	2012-08-14	HERBATA	5	MALINOWSKI
11	2012-08-14	HERBATA	4	KOWALSKI
12	2012-08-11	KAWA	3	KOZŁOWSKI
13	2012-08-11	KAWA	2	KOZŁOWSKI
14	2012-08-12	KAWA	5	MALINOWSKI
15	2012-08-14	KAWA	6	KOWALSKI
16	2012-08-14	KAWA	3	MALINOWSKI
17	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
18	2012-08-12	NAPÓJ	3	WIŚNIEWSKI

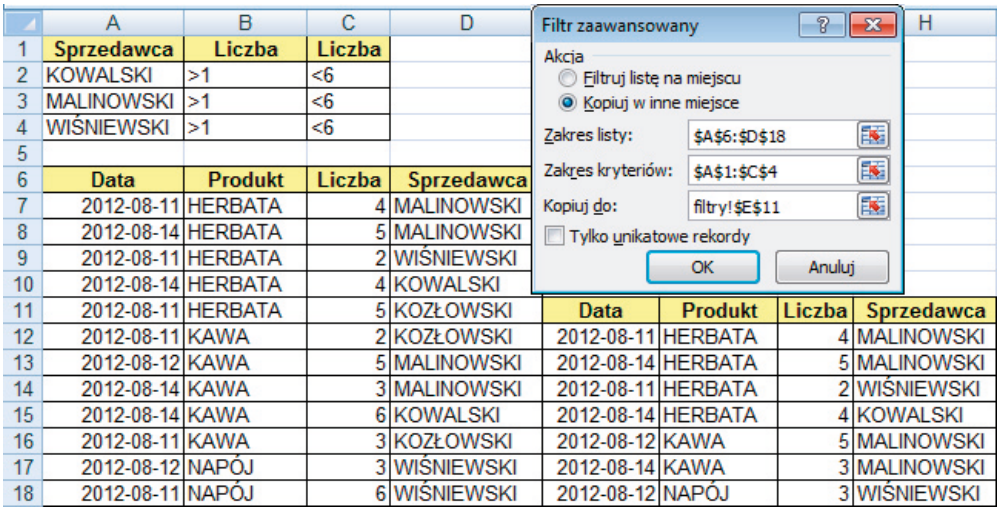


Data	Produkt	Liczba	Sprzedawca
2012-08-11	HERBATA	4	MALINOWSKI
2012-08-14	HERBATA	5	MALINOWSKI
2012-08-14	HERBATA	4	KOWALSKI
2012-08-12	KAWA	5	MALINOWSKI
2012-08-14	KAWA	6	KOWALSKI
2012-08-11	NAPÓJ	6	WIŚNIEWSKI

Rysunek 1.14. Filtr zaawansowany z kryteriami wyboru, tworzący nową, spełniającą kryteria listę danych

Ćwiczenie 1.8. Wyświetl wszystkie transakcje sprzedawców Kowalskiego, Malinowskiego i Wiśniewskiego, w których wielkość sprzedaży jest różna od maksymalnej i minimalnej.

To ćwiczenie wykonaj samodzielnie — pomocą służy rysunek 1.15. Kryteria w tabeli wyboru są następujące: wybierz rekordy, w których *(Sprzedawca="KOWALSKI" i Liczba>1 i Liczba<6)* lub *(Sprzedawca="MALINOWSKI" i Liczba>1 i Liczba<6)* lub *(Sprzedawca="WIŚNIEWSKI" i Liczba>1 i Liczba<6)*.



	A	B	C	D
1	**Sprzedawca**	**Liczba**	**Liczba**	
2	KOWALSKI	>1	<6	
3	MALINOWSKI	>1	<6	
4	WIŚNIEWSKI	>1	<6	
5				
6	**Data**	**Produkt**	**Liczba**	**Sprzedawca**
7	2012-08-11	HERBATA	4	MALINOWSKI
8	2012-08-14	HERBATA	5	MALINOWSKI
9	2012-08-11	HERBATA	2	WIŚNIEWSKI
10	2012-08-14	HERBATA	4	KOWALSKI
11	2012-08-11	HERBATA	5	KOZŁOWSKI
12	2012-08-11	KAWA	2	KOZŁOWSKI
13	2012-08-12	KAWA	5	MALINOWSKI
14	2012-08-14	KAWA	3	MALINOWSKI
15	2012-08-14	KAWA	6	KOWALSKI
16	2012-08-11	KAWA	3	KOZŁOWSKI
17	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
18	2012-08-11	NAPÓJ	6	WIŚNIEWSKI

Data	**Produkt**	**Liczba**	**Sprzedawca**
2012-08-11	HERBATA	4	MALINOWSKI
2012-08-14	HERBATA	5	MALINOWSKI
2012-08-11	HERBATA	2	WIŚNIEWSKI
2012-08-14	HERBATA	4	KOWALSKI
2012-08-12	KAWA	5	MALINOWSKI
2012-08-14	KAWA	3	MALINOWSKI
2012-08-12	NAPÓJ	3	WIŚNIEWSKI

Rysunek 1.15. Tablica kryteriów, opcje filtrowania i tabela wynikowa do ćwiczenia 1.8

1.4.4. Sortowanie danych

Filtrowanie pozwala uzyskać przejrzystość i czytelność danych. Zastosowanie kryteriów daje jednoznaczny zestaw rekordów, które możesz odpowiednio posortować, aby zobaczyć dane uporządkowane według wybranego pola. Często jednak trzeba posortowania danych bez używania filtrów, których obecność sygnalizują strzałki obok nazw pól. Dane można również posortować w prosty sposób.

Wskazówka

Pamiętaj! Nie zaznaczaj kolumn, a jedynie umieść kursor w dowolnym miejscu tej kolumny, według której chcesz posortować dane.

Ćwiczenie 1.9. Zacznijmy od alfabetycznego sortowania danych według nazwisk sprzedawców. Umieść kursor w dowolnej komórce, w której wpisane jest nazwisko sprzedawcy, na przykład Kowalski. Kliknij przycisk *Sortuj rosnąco* . Gotowe. A teraz sortowanie według daty w porządku malejącym. Umieść kursor w kolumnie z datami. Kliknij przycisk narzędzia *Sortuj malejąco* .

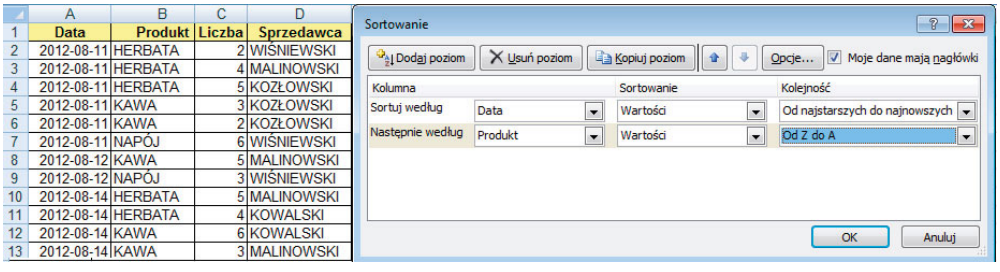
Wskazówka

Daty to liczby, więc polecenia: „Uporządkuj dane tak, aby najmłodsze były na początku" albo „Na końcu listy wyświetl pliki zmodyfikowane najpóźniej" należy traktować jak operacje na danych liczbowych. Może więc to zdanie będzie cenną wskazówką do wykonania tego typu sortowania: *Ojciec jest starszy od swojego syna, ale jego data urodzenia jest mniejsza.*

Jeśli dane mają być posortowane według kilku kryteriów, nie wystarczy już tylko albo . Potrzebne będzie narzędzie do sortowania zaawansowanego.

Ćwiczenie 1.10. Posortuj dane według daty sprzedaży tak, aby na początku listy znajdowały się najwcześniejsze transakcje. Rekordy z tą samą datą sprzedaży uporządkuj według nazw produktów w malejącym porządku alfabetycznym.

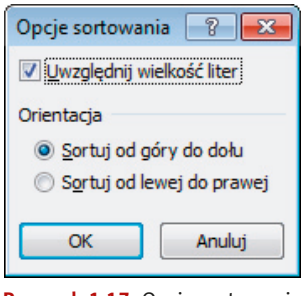
Wybierz z karty *Dane/Sortowanie i filtrowanie* polecenie *Sortuj*. Wypełnij pola okna dialogowego tak, jak pokazano na rysunku 1.16. W pierwszym kryterium sortowania w polu *Sortuj według* wybierz kolumnę *Data*, ustal sortowanie według *Wartości* i kolejność sortowania rosnąco, czyli wskaż opcję *Od najstarszych do najnowszych*. Kliknięcie przycisku *Dodaj poziom* uaktywni następny poziom sortowania. Wybierz w nim kolumnę *Produkt*, sortowanie *Wartości* i kolejność malejącą, czyli *Od Z do A*. Na górze okna dialogowego zaznacz opcję *Moje dane mają nagłówki* — to jest bardzo ważne, w przeciwnym razie nastąpi posortowanie również pierwszego wiersza tabeli. Kliknij przycisk *OK*.



	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
2	2012-08-11	HERBATA	2	WIŚNIEWSKI
3	2012-08-11	HERBATA	4	MALINOWSKI
4	2012-08-11	HERBATA	5	KOZŁOWSKI
5	2012-08-11	KAWA	3	KOZŁOWSKI
6	2012-08-11	KAWA	2	KOZŁOWSKI
7	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
8	2012-08-12	KAWA	5	MALINOWSKI
9	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
10	2012-08-14	HERBATA	5	MALINOWSKI
11	2012-08-14	HERBATA	4	KOWALSKI
12	2012-08-14	KAWA	6	KOWALSKI
13	2012-08-14	KAWA	3	MALINOWSKI

Rysunek 1.16. Złożone kryteria sortowania

W oknie dialogowym *Sortowanie* jest możliwość wybrania dodatkowych opcji, pokazanych na rysunku 1.17, wśród nich dość rzadko stosowanego w bazach danych sortowania *od lewej do prawej*, czyli dla danych ustawionych w wierszach, a nie w kolumnach. Tutaj też można zaznaczyć opcję *Uwzględnij wielkość liter*, co jest czasami istotne. Sprawdź, jak działa ta opcja. Zapisz w bazie nazwisko jednego ze sprzedawców małymi literami. Posortuj rekordy z uwzględnieniem wielkości liter. Jaki jest efekt? Mała litera jest przed wielką. Nie zapomnij usunąć zaznaczenia pola uwzględniania wielkości liter. MS Excel zapamięta bowiem ustawienia tej opcji do dalszej pracy z arkuszem.



Rysunek 1.17. Opcje sortowania

Ćwiczenie 1.11. Wyświetl dane tak, aby można było zobaczyć je w porządku od najmłodszego do najstarszego wpisu. Rekordy mające tę samą datę powinny być ułożone od największej do najmniejszej sprzedaży — tak jak to pokazano na rysunku 1.18.

	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
2	2012-08-14	KAWA	6	KOWALSKI
3	2012-08-14	HERBATA	5	MALINOWSKI
4	2012-08-14	KAWA	3	MALINOWSKI
5	2012-08-14	HERBATA	3	KOWALSKI
6	2012-08-12	KAWA	5	MALINOWSKI
7	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
8	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
9	2012-08-11	HERBATA	5	KOZŁOWSKI
10	2012-08-11	HERBATA	4	MALINOWSKI
11	2012-08-11	KAWA	3	KOZŁOWSKI
12	2012-08-11	KAWA	2	KOZŁOWSKI
13	2012-08-11	HERBATA	2	WIŚNIEWSKI

Rysunek 1.18. Efekt złożonego sortowania do ćwiczenia 1.11

1.4.5. Sumy częściowe (pośrednie)

Opcja *Suma częściowa* jest dostępna na karcie *Dane*, w grupie poleceń *Konspekt*.

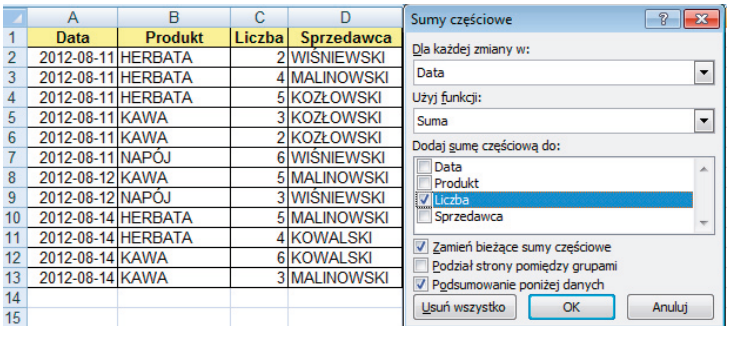
Sumy częściowe i końcowe są tworzone dla tych samych wartości (powtarzających się) występujących w kolejnych wierszach wybranej kolumny (pola), dlatego najpierw należy posortować rekordy według tej właśnie kolumny.

Sumy częściowe informują nas między innymi o całkowitej sprzedaży na przykład z danego dnia, danego produktu, każdego ze sprzedawców. Sumy pośrednie dla każdej daty możesz uzyskać tylko wtedy, gdy lista jest posortowana według daty. Innymi słowy, należy zacząć od posortowania danych, a reszta zrobi się prawie sama!

Operację tworzenia sumy częściowej najlepiej wykonywać na kopii danych zamieszczonej w sąsiednim arkuszu lub kopii skoroszytu. Przed przystąpieniem do następnych ćwiczeń utwórz kopię arkusza *sklep* i nazwij ją *sumy*.

Ćwiczenie 1.12. Wyświetl towary sprzedane w każdym kolejnym dniu pracy. Sporządź podsumowanie liczby artykułów sprzedanych w każdym dniu oraz oblicz, ile ogółem sprzedano wszystkich artykułów.

Jest to zadanie typowe dla zastosowania sum częściowych. Polem, które się powtarza i dla którego wybierzesz artykuły, jest *Data*. Zacznij od posortowania danych. Umieść kursor w dowolnym miejscu kolumny *Data* i kliknij przycisk narzędzia *Sortuj od najstarszych do najnowszych*. Teraz wybierz narzędzie *Suma częściowa.* Zostanie otwarte okno dialogowe *Sumy częściowe*, widoczne na rysunku 1.19.



	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
2	2012-08-11	HERBATA	2	WIŚNIEWSKI
3	2012-08-11	HERBATA	4	MALINOWSKI
4	2012-08-11	HERBATA	5	KOZŁOWSKI
5	2012-08-11	KAWA	3	KOZŁOWSKI
6	2012-08-11	KAWA	2	KOZŁOWSKI
7	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
8	2012-08-12	KAWA	5	MALINOWSKI
9	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
10	2012-08-14	HERBATA	5	MALINOWSKI
11	2012-08-14	HERBATA	4	KOWALSKI
12	2012-08-14	KAWA	6	KOWALSKI
13	2012-08-14	KAWA	3	MALINOWSKI
14				
15				

Rysunek 1.19. Opcje wyboru sum częściowych

W polu *Dla każdej zmiany w:* wybierz pole *Data* (powinno być już wybrane). W polu *Użyj funkcji:* wybierz funkcję *Suma*. W polu *Dodaj sumę częściową do:* zaznacz pozycję (pole) *Liczba*. Znajdziesz tu pasek przewijania, którym możesz przewinąć listę pozycji w górę. Warto sprawdzić, czy żadna inna opcja nie jest zaznaczona.

Wciśnij *OK*. Lista wygląda teraz zupełnie inaczej (rysunek 1.20). Po pierwsze, pojawiły się

	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
2	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
3	2012-08-11	HERBATA	5	KOZŁOWSKI
4	2012-08-11	HERBATA	4	MALINOWSKI
5	2012-08-11	KAWA	3	KOZŁOWSKI
6	2012-08-11	KAWA	2	KOZŁOWSKI
7	2012-08-11	HERBATA	2	WIŚNIEWSKI
8	**2012-08-11 Suma**		22	
9	2012-08-12	KAWA	5	MALINOWSKI
10	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
11	**2012-08-12 Suma**		8	
12	2012-08-14	KAWA	6	KOWALSKI
13	2012-08-14	HERBATA	5	MALINOWSKI
14	2012-08-14	KAWA	3	MALINOWSKI
15	2012-08-14	HERBATA	3	KOWALSKI
16	**2012-08-14 Suma**		17	
17	**Suma końcowa**		47	

Rysunek 1.20. Tabela sum częściowych

sumy częściowe i **suma końcowa**, po drugie, zmieniła się sama forma listy — po lewej stronie widzisz przyciski oznaczające formę konspektu. Kliknij po kolei przyciski z symbolem minusa. Co się dzieje? Lista jest zwijana tak, że widoczne są tylko sumy częściowe. Klikaj kolejno przyciski z symbolem plusa, żeby ponownie rozwinąć listę. Są też dostępne trzy przyciski z numerami poziomów konspektu *1*, *2*, *3*, które odpowiadają kolejno: *1* — sumie końcowej, *2* — sumom częściowym dla poszczególnych grup, *3* — wszystkim sumom wraz z pozycjami zgrupowanymi w każdym dniu.

Wskazówka

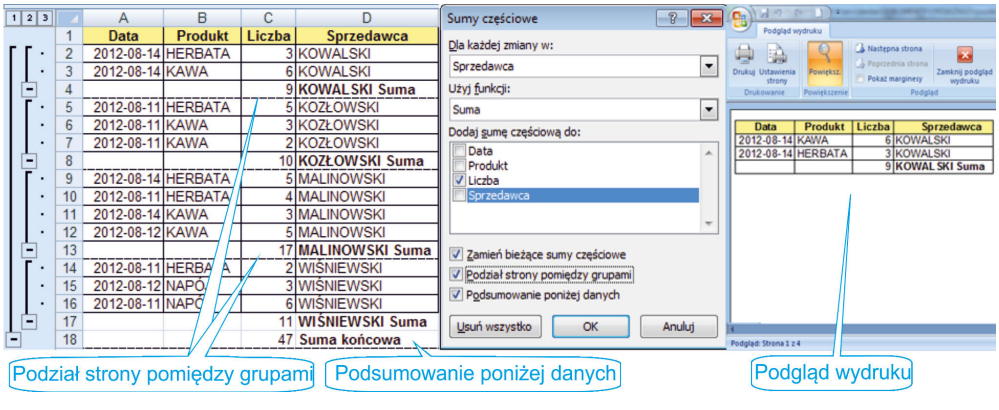
Sum częściowych i końcowych pozbędziesz się w następujący sposób: w tabeli kliknij komórkę zawierającą sumę częściową. Na karcie *Dane*, w grupie *Konspekt*, kliknij przycisk *Suma częściowa*, a następnie kliknij *Usuń wszystko* w oknie dialogowym. Podczas usuwania sum częściowych Excel usuwa również konspekt i wszelkie podziały stron umieszczone na liście wraz z sumami częściowymi.

Ćwiczenie 1.13. Czas na podsumowanie sprzedaży. Sporządź zestawienie wszystkich artykułów i podlicz ich sprzedaż, tak jak pokazano na rysunku 1.21.

Ćwiczenie 1.14. Aby obliczyć premię za sprzedaż, sporządź zestawienie liczby artykułów sprzedanych przez każdego sprzedawcę. Dodaj znak podziału strony pomiędzy grupami (na wydruku każdy pracownik będzie na osobnej stronie), tak jak na rysunku 1.22. Przedstaw wyniki na wykresie.

	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
2	2012-08-14	HERBATA	5	MALINOWSKI
3	2012-08-14	HERBATA	3	KOWALSKI
4	2012-08-11	HERBATA	5	KOZŁOWSKI
5	2012-08-11	HERBATA	4	MALINOWSKI
6	2012-08-11	HERBATA	2	WIŚNIEWSKI
7		**HERBATA Suma**	19	
8	2012-08-14	KAWA	6	KOWALSKI
9	2012-08-14	KAWA	3	MALINOWSKI
10	2012-08-12	KAWA	5	MALINOWSKI
11	2012-08-11	KAWA	3	KOZŁOWSKI
12	2012-08-11	KAWA	2	KOZŁOWSKI
13		**KAWA Suma**	19	
14	2012-08-12	NAPÓJ	3	WIŚNIEWSKI
15	2012-08-11	NAPÓJ	6	WIŚNIEWSKI
16		**NAPÓJ Suma**	9	
17		**Suma końcowa**	47	

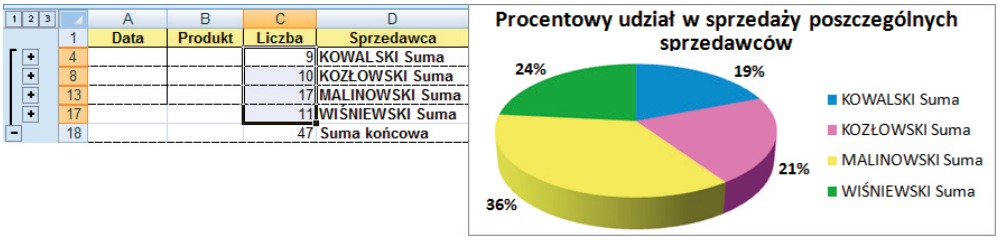
Rysunek 1.21. Podsumowanie sprzedanych artykułów za pomocą sum częściowych



Rysunek 1.22. Opcje tabeli podsumowań do ćwiczenia 1.14

Aby wykonać wykres przedstawiający podsumowania liczby sprzedanych artykułów każdego sprzedawcy, należy wyświetlić poziom 2. konspektu sum częściowych, czyli ze-

stawienia dla poszczególnych grup, bez widoku szczegółów zestawienia w grupach. Na podstawie takiego zestawienia przygotuj wykres kołowy, który pokaże procentowy udział każdego sprzedawcy w ogólnej sprzedaży, tak jak to zaprezentowano na rysunku 1.23.



	A	B	C	D
1	Data	Produkt	Liczba	Sprzedawca
4			9	KOWALSKI Suma
8			10	KOZŁOWSKI Suma
13			17	MALINOWSKI Suma
17			11	WIŚNIEWSKI Suma
18			47	Suma końcowa

Rysunek 1.23. Prezentacja graficzna danych na podstawie sum częściowych

Tworzenie wykresu rozpocznij od zaznaczenia wartości podsumowań liczby sprzedanych produktów (bez sumy końcowej). Na karcie *Wstawianie* z grupy *Wykres* wybierz *Kołowy*. Po utworzeniu wykresu zostaną uaktywnione karty *Narzędzia wykresów*. Na karcie *Projektowanie* z grupy *Dane* wybierz polecenie *Zaznacz dane*. W sekcji *Etykiety osi poziomej (kategorii)* znajdują się obecnie numery serii od 1 do 4. Wybierz więc polecenie *Edytuj* i wskaż nazwiska wszystkich sprzedawców, po czym zatwierdź zmianę. Aby dodać tytuł, wybierz kartę *Układ* i polecenie *Tytuł wykresu*. W celu dodania etykiet z procentowym rozkładem udziału w ogólnej sprzedaży kliknij polecenie *Etykiety danych* i z listy wybierz polecenie *Więcej opcji etykiet danych*…. W oknie dialogowym ustal, że etykieta ma zawierać wartości procentowe zamiast liczbowych, określ również jej położenie. Pozostaje jeszcze formatowanie wyglądu arkusza.

Po utworzeniu wykresu zobacz, jak on się zachowuje, gdy włączysz poziom 3. widoku konspektu sum pośrednich, czyli widok wszystkich rekordów należących do grupy. Powiększ obszar wykresu, aby zobaczyć szczegóły.

1.4.6. Statystyki oparte na raporcie tabeli przestawnej

Tabela przestawna pozwala łączyć, porównywać i analizować duże ilości danych. Umożliwia przeglądanie różnego rodzaju zestawień danych źródłowych, wyświetlanie szczegółowych danych z interesujących obszarów i tworzenie raportów. Raporty w postaci tabel przestawnych to takie zestawienia danych, które uwidaczniają prawidłowości i zależności oraz umożliwiają analizę trendów. Tabela utworzona za pomocą raportu tabeli przestawnej jest tablicą interaktywną. Dane mogą być rozmieszczane na wiele sposobów i zestawiane zgodnie z różnymi punktami widzenia.

Jak zatem w prosty sposób porównywać i analizować dane — wyświetlać ich różne widoki i zmieniać dane w czytelne informacje?

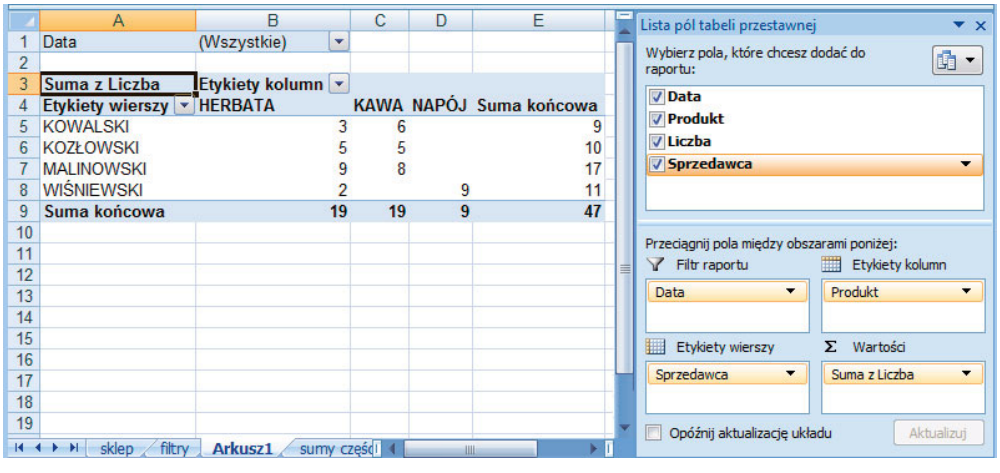
Tworzenie raportu tabeli przestawnej polega na rozmieszczeniu informacji na wiele różnych sposobów. W każdej chwili można zmienić układ wierszy i kolumn, można przenosić dane tak długo, aż efekt będzie zadowalający. Stąd nazwa: *tabela przestawna*, gdyż można dowolnie przestawiać wiersze i kolumny w nowo powstałej tabeli, która jest tworzona z jednotabelowej bazy danych.

Zobaczmy więc, w jaki sposób tabela przestawna ułatwia pracę, kiedy potrzebne jest szybkie zestawienie informacji o tym, kto, co i kiedy sprzedał.

Ćwiczenie 1.15. Stwórz statystykę sprzedaży w taki sposób, aby po utworzeniu tabeli przestawnej można było zobaczyć, ile sztuk produktów danego rodzaju sprzedał każdy ze sprzedawców w danym dniu.

Umieść kursor w dowolnym miejscu listy, w którym znajdują się dane. Na karcie *Wstawianie* z grupy *Tabele* wybierz polecenie *Tabela przestawna*. Uruchomiony zostanie kreator *Tworzenie tabeli przestawnej*. Podaj lub zaznacz myszą obszar w arkuszu zawierający dane i ustal położenie raportu w nowym arkuszu. Przejdź do projektowania tabeli, zatwierdzając ustawienia przyciskiem *OK*.

Tabelę raportu konstruujemy, wybierając dane do zestawienia. Tworzenie tabeli przestawnej polega na przeciąganiu odpowiednich pól do odpowiednich obszarów tabeli, co ilustruje rysunek 1.24. Pole *Produkt* przeciągnij myszą do obszaru o nazwie *Etykiety kolumn*. Pole *Sprzedawca* przeciągnij do obszaru o nazwie *Etykiety wierszy*. Tabela powstaje na bieżąco i w tej chwili ma już nagłówki kolumn i wierszy. Podsumowane mają być liczby sprzedanych towarów, zatem pole *Liczba* przeciągnij do obszaru *Wartości* (spowoduje to zamianę tego elementu na *Suma z Liczba*). Zestawienie ma sumować transakcje w danym dniu, więc do obszaru *Filtr raportu* przeciągnij pole *Data*. Zamknij okno listy pól tabeli przestawnej.

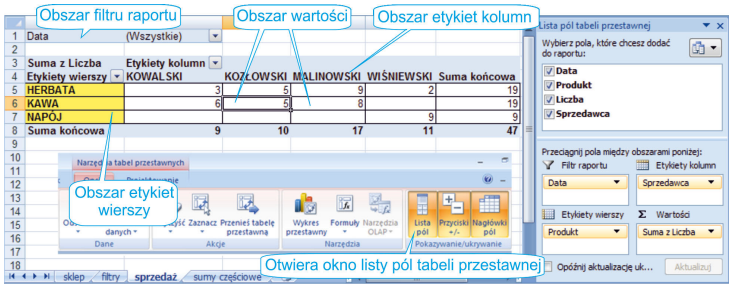


Rysunek 1.24. Tworzenie tabeli przestawnej — układ pól

W ten sposób została utworzona tabela przestawna. Zwróć uwagę na znajdujące się u dołu okna Excela zakładki arkuszy — dla potrzeb tabeli stworzony został całkiem nowy arkusz. Zmień jego nazwę na *sprzedaż*.

Tabelę możesz jeszcze poprawić. Zauważ, że na Wstążce pojawiły się nowe karty — *Narzędzia tabel przestawnych*. Na karcie *Opcje*, w grupie *Pokazywanie/ukrywanie*, zamieszczono przycisk *Lista pól*, który wyświetli okno *Lista pól tabeli przestawnej*. W przypadku błędnego przyporządkowania pól do obszarów dokonaj korekty, a następnie zamień miejscami ko-

lumny i wiersze, przenosząc pola *Sprzedawca* i *Produkt* (rysunek 1.25). Sformatuj powstałą tabelę (na przykład dodaj krawędzie, ustal szerokości kolumn).



Rysunek 1.25. Dodawanie pól do istniejącej tabeli przestawnej

Teraz możesz oglądać rekordy sprzedaży dla danego dnia albo po prostu wyświetlić wszystkie rekordy, niezależnie od daty sprzedaży. Kliknij strzałkę znajdującą się w polu obok pola o nazwie *Data* i sprawdź możliwości tabeli. Wszystkie pola, przy których widnieją strzałki, pozwalają na wybór poszczególnych kryteriów lub pokazanie wszystkich rekordów. Możesz więc wybrać podsumowanie tylko dla jednego produktu lub nie uwzględnić w zestawieniu jednego ze sprzedawców. Jeżeli nie chcesz filtrować danych na etykietach wierszy i kolumn, możesz wyłączyć ich widok przyciskiem *Nagłówki pól* znajdującym się w grupie *Pokazywanie/ukrywanie*. Gotowa tabela przestawna powinna wyglądać tak jak na rysunku 1.26.

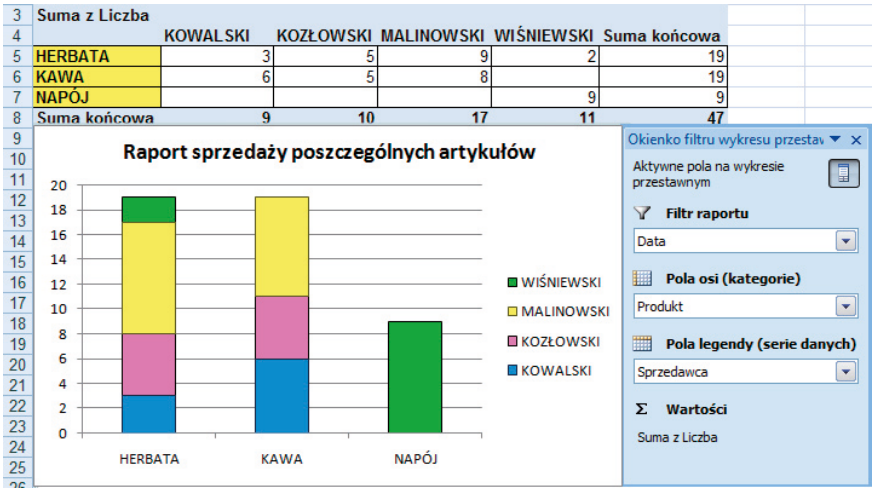
	A	B	C	D	E	F
1	Data	(Wszystkie)				
2						
3	Suma z Liczba					
4		KOWALSKI	KOZŁOWSKI	MALINOWSKI	WIŚNIEWSKI	Suma końcowa
5	HERBATA	3	5	9	2	19
6	KAWA	6	5	8		19
7	NAPÓJ				9	9
8	Suma końcowa	9	10	17	11	47

Rysunek 1.26. Raport tabeli przestawnej

Takiej tabeli nie można utworzyć na przykład przy użyciu sum pośrednich. Tutaj wyraźnie widać, że nagłówki kolumn i wierszy pochodzą z jednej tabeli danych. Dane zawarte w tabeli przestawnej można przedstawić również graficznie, tworząc wykres przestawny.

Ćwiczenie 1.16. Na podstawie danych zawartych w raporcie tabeli przestawnej utwórz wykres przestawny.

Ćwiczenie to wykonasz za pomocą kreatora *Wykres przestawny* dostępnego na karcie *Opcje*, w grupie *Narzędzia*. Dane zawarte w tabeli przestawnej najlepiej zilustrować na wykresach skumulowanych. W tym ćwiczeniu użyto wykresu kolumnowego skumulowanego, gdzie na osi pionowej pokazane są wartości sprzedaży poszczególnych produktów. Utworzony wykres sformatuj tak jak na rysunku 1.27. Wykres przestawny formatuje się tak jak zwykły wykres — pomocne jest przy tym menu podręczne. Sprawdź, jak elastycznie zmieniają się dane, zgodnie z wyborem kryteriów. Jedyne, o czym musisz pamiętać, wykonując to ćwiczenie, to umieszczenie kursora wewnątrz danych tabeli przestawnej.



Rysunek 1.27. Wykres przestawny

Kreator wykresów natychmiast rysuje wykres. Dla porównania wykonaj drugi wykres przestawny, wykorzystując typ wykresu pozwalający na procentowe przedstawienie udziału sprzedawców w sprzedaży danego artykułu. Zastosuj typ wykresu *100% skumulowany kolumnowy*.

Ćwiczenie 1.17. Opracuj statystykę sprzedaży w nieco inny sposób. Nadal będzie można zobaczyć, ile sztuk produktów danego rodzaju sprzedał każdy ze sprzedawców danego dnia, jednak układ kolumn i wierszy zostanie zmieniony.

Teraz wiersze i kolumny to kolejne daty i nazwiska sprzedawców. Należy obliczyć wielkość sprzedaży, a filtr raportu ma pokazywać poszczególne artykuły. Powodzenia!

1.4.7. Zadania

W ćwiczeniu 1.3 należało pobrać dane z plików tekstowych zamieszczonych na płycie CD dołączonej do podręcznika i utworzyć jedną tabelę zawierającą dane studentów. Tabela miała zostać zapisana pod nazwą *studenci_import.xls*.

Ponieważ dane zgromadzone w tych plikach tekstowych będą podstawą do rozwiązania zadania, plik wynikowy *studenci.xls* znajduje się również na płycie CD. Części pobrane z różnych plików tekstowych pocieniowano różnymi kolorami. Utwórz kopię tego pliku na swoim dysku i wykonaj kilka poleceń związanych z danymi w nim zawartymi.

Na kilku kierunkach Uniwersytetu Wirtualnego studiuje spora grupa studentów. W arkuszu *studenci.xls* znajdują się dane tych studentów, którzy mieszkają w trzech domach studenckich (akademikach). Studia trwają trzy lata. W arkuszu są zawarte dane osobowe, dane uczelniane oraz oceny studentów.

Zadania sprawdzają umiejętności uzyskane na lekcjach technologii informacyjnej oraz informatyki.

Zadanie 1.1. Odpowiedz na kilka pytań związanych z ocenami studentów. Wszystkie obliczenia zamieść w skoroszycie *studenci*; jeśli to konieczne, dodaj kolejne arkusze.

a) Podaj ID trojga studentów, którzy uzyskali najwyższą średnią.

b) Jaką średnią ocen studenci uzyskiwali najczęściej?

c) Z jakiego przedmiotu średnia ocen była najniższa?

d) Ilu studentów uzyskało więcej niż jedną ocenę niedostateczną (2 jest oceną niedostateczną, a skala wynosi od 2 do 5)?

e) Ile ocen niedostatecznych otrzymali wszyscy studenci?

f) Z jakiego miasta pochodzi największa liczba studentów tego uniwersytetu?

Zadanie 1.2. Dodaj do arkusza *studenci.xls* pole *Komentarz*. W tym polu wpisz odpowiednią uwagę każdemu studentowi.

Jeżeli wszystkie oceny ma pozytywne, napisz `ZALICZ ROK`, gdy student jest na pierwszym lub drugim roku, albo `KONIEC STUDIÓW`, gdy jest to student trzeciego roku.

Jeżeli student otrzymał jedną ocenę niedostateczną, napisz `POPRAWKA`.

Jeżeli otrzymał dwie oceny niedostateczne, napisz `POWTARZA ROK`, jednak nie można powtarzać pierwszego roku, więc studenci pierwszego roku otrzymują wówczas wpis: `SKREŚLIĆ`.

Jako rozwiązanie podaj, ile poszczególnych komentarzy wystąpiło w polu *Komentarz*.

Komentarz	Liczba wystąpień
ZALICZ ROK	
KONIEC STUDIÓW	
POPRAWKA	
POWTARZA ROK	
SKREŚLIĆ	

Uwaga! Oczywiście można, a nawet należy wstawić jeszcze dodatkowe pola, w których zostanie obliczona na przykład liczba ocen niedostatecznych. Do rozwiązania użyj funkcji warunkowej.

Zadanie 1.3. Ilu studentów każdego kierunku zaliczyło rok, ukończyło studia, ma poprawkę, powtarza rok, a ilu trzeba skreślić? Wypełnij tabelę.

Komentarz	Liczba studentów na poszczególnych kierunkach studiów				
	BI	CH	FI	IN	MA
ZALICZ ROK					
KONIEC STUDIÓW					
POPRAWKA					
POWTARZA ROK					
SKREŚLIĆ					

Zadanie 1.4. Ilu studentów z każdego kierunku studiów mieszka w domach studenta? Wypełnij tabelę.

Akademik	Liczba studentów z danego kierunku mieszkających w poszczególnych domach studenta				
	BI	CH	FI	IN	MA
DS1					
DS2					
DS3					

Zadanie 1.5. Ilu studentów z każdego rocznika studiuje na poszczególnych kierunkach studiów? Wypełnij tabelę.

Rocznik	Liczba studentów z danego kierunku urodzonych w danym roku				
	BI	CH	FI	IN	MA

Wskazówka: przejrzyj funkcje daty i czasu.

Zadanie 1.6. Pokaż na wykresie liczbę ubezpieczonych i nieubezpieczonych studentów mieszkających w poszczególnych domach studenta. W którym DS-ie mieszka najwięcej nieubezpieczonych?

Zadanie 1.7. Odpowiedz na pytania typu prawda – fałsz:

1. Każda tabela zawierająca dane może pełnić rolę bazy danych.
2. Zgromadzone w telefonie komórkowym kontakty są kartotekową bazą danych.
3. W tabeli bazy danych możesz zamieszczać scalone komórki.
4. *Rekordy* i *pola* to dwa podstawowe pojęcia bazodanowe.
5. Dane w tabeli można posortować tylko według jednego kryterium.
6. Wykres jest graficzną prezentacją danych.
7. Dane gromadzone w bazie danych należy rozbijać na pola elementarne.
8. Formularze służą do formatowania danych w tabeli.
9. Tabela przestawna powstaje po zamianie wierszy z kolumnami tabeli.
10. Dane w pliku tekstowym należy oddzielić separatorem, aby mogły stanowić źródło danych.
11. Autofiltrowanie danych pozwala na wyszukanie danych z jednym kryterium wyszukiwawczym.
12. W filtrze zaawansowanym stosujemy tabelę kryteriów.

13. Raporty tabel przestawnych można prezentować na wykresie przestawnym.
14. Stosowanie funkcji obliczeniowych na danych filtrowanych daje poprawne wyniki.
15. Pojęcia *suma pośrednia* i *suma częściowa* to synonimy.

Zadanie 1.8. Na płycie CD dołączonej do podręcznika znajduje się plik *miejscowosci.xls*. W arkuszu *dane* zamieszczono informacje na temat miejscowości, ich ludności, wojska, głównego źródła dochodów. Na podstawie danych zgromadzonych w arkuszu przygotuj, wykorzystując funkcje arkusza kalkulacyjnego oraz poznane operacje bazodanowe, zestawienia danych pozwalające odpowiedzieć na następujące pytania:

a) Ile ludności ma każdy władca?
b) Ilu mieszkańców mieszka w poszczególnych typach miejscowości?
c) Ile miejscowości każdego typu jest pod rządami poszczególnych władców?
d) Ilu mieszkańców każdego władcy żyje w biedzie, a ilu w dostatku?
e) Który władca ma najwięcej doskonale wyszkolonego wojska, który średnio dobrych żołnierzy, a który najliczniejszą słabą armię?
f) Przedstaw na wykresie liczbę poddanych tego władcy, który ma najliczniejszą armię. Dokonaj procentowego podziału poddanych w zależności od źródła ich dochodów.

Dodatkowe zadania Na płycie CD dołączonej do podręcznika zamieszczono dodatkowe pliki z rozwiązaniami zadań, które pozwolą na lepsze przygotowanie się do matury. Znajdują się tam zadania algorytmiczne, do których rozwiązania użyto standardowych i dodatkowych funkcji arkusza kalkulacyjnego. Wszystkie pliki znajdują się w folderze *Excel algorytmy — ciekawe rozwiazania*. Oto krótki opis ich zawartości:

- *decymalne_na_binarne.xls*: zamienia podaną liczbę całkowitą na binarną i na odwrót;
- *formaty.xls*: wypisuje liczby w różnych formatach, na przykład wypisując tzw. wiodące (początkowe) zera przed liczbą;
- *kombinacje.xls*: oblicza dwiema metodami kombinacje, na przykład liczbę zakładów totolotka gwarantującą wygraną;
- *NWD_i_NWW_Euklides.xls*: oblicza NWD i NWW algorytmem Euklidesa metodą odejmowania i dzielenia całkowitego;
- *NWD_skracanie_ulamka.xls*: wykorzystuje algorytm Euklidesa do skrócenia ułamka i zamiany ułamka niewłaściwego na właściwy;
- *parzysta_nieparzysta.xls*: sprawdza, czy podana liczba jest parzysta czy nieparzysta. Dla danych nieliczbowych wypisuje odpowiedni komunikat;
- *pierwiastek_n_stopnia.xls*: oblicza pierwiastek n-tego stopnia podanej liczby;
- *przyblizony_pierwiastek_Newtona.xls*: oblicza dwiema metodami pierwiastek kwadratowy podanej liczby;
- *rzut_kostka.xls*: losuje liczby i przelicza je na rzut kostką. Sprawdza, czy liczba jest parzysta. Jest to dobry pomysł na wygenerowanie wielu danych do zadań. Plik ten może być dużą pomocą w rozwiązaniu zadania maturalnego *matura1.2.pdf* zamieszczonego na płycie CD dołączonej do podręcznika;

- *silnia.xls*: oblicza silnię za pomocą funkcji *Silnia* oraz rekurencyjnie (to nie pomyłka — rekurencja w arkuszu kalkulacyjnym!);
- *szyfr_Cezara.xls*: szyfruje tekst kluczem podanym przez użytkownika;
- *wyszukaj_pionowo.xls*: wyszukuje wartość z tablicy danych położoną w wybranej kolumnie tabeli na wskazanej pozycji (we wskazanym wierszu).



Wybrane zadania maturalne

Na płycie CD dołączonej do podręcznika zamieszczono treści wybranych zadań maturalnych oraz niezbędne dane. Rozwiązanie tych zadań zapewni lepsze przygotowanie się do matury. Wymagają one od ucznia/uczennicy umiejętności wykorzystania standardowych funkcji oraz bazodanowych operacji arkusza kalkulacyjnego.

- *matura1.1.pdf* (*Egzamin maj 2005 r. Arkusz II, zadanie 4. PROJEKT*).
- *matura1.2.pdf* (*Egzamin próbny 2005 r. Arkusz II, zadanie 7. GRA*).
- *matura1.3.pdf* (*Egzamin maj 2007 r. Arkusz II, zadanie 6. AUDIOTELE*).
- *matura1.4.pdf* (*Informator maturalny od 2009 r. Arkusz II, zadanie 4. LICZBY*).
- *matura1.5.pdf* (*Egzamin maj 2002 r. Arkusz II, zadanie 4. FIRMA*).
- *matura1.6.pdf* (*Egzamin maj 2010 r. Arkusz II, poziom podstawowy, zadanie 5. UPUSTY*).
- *matura1.7.pdf* (*Egzamin maj 2010 r. Arkusz II, poziom rozszerzony, zadanie 5. NUMER PESEL*).
- *matura1.8.pdf* (*Egzamin maj 2009 r. Arkusz II, poziom podstawowy, zadanie 4. TEMPERATURY*).

Rozdział 2.

RELACYJNA BAZA DANYCH

Dawniej, gdy danych było niewiele, można je było zapamiętać lub zapisać w notesie. Obecnie to już nie wystarcza. Trudno wyobrazić sobie sprawne funkcjonowanie supermarketu, w którym kasjerki notują ręcznie w księdze rachunkowej liczbę i wartość sprzedanych towarów. Dzisiaj nie można sobie pozwolić na zamknięcie sklepów na kilka dni w celu zliczenia towarów, czyli wykonania inwentaryzacji. Kody kreskowe na wszystkich towarach i doskonale działający system zarządzający bazą towarów pozwalają nie tylko wydrukować paragon kasowy, ale i na bieżąco kontrolować stan artykułów w sklepie, a nawet ich terminy przydatności do spożycia.

Jeżeli myślisz, że nie korzystasz z żadnej bazy danych, jesteś w błędzie. Bazy danych są wszędzie. Szufladki z fiszkami książek w bibliotece, album z wizytówkami, karty zdrowia pacjentów w przychodni, tradycyjny dziennik lekcyjny, książka telefoniczna — zarówno drukowana, jak i ta w Twoim telefonie komórkowym — to tylko niektóre z baz, z których prawdopodobnie korzystasz.

Zanim przystąpisz do projektowania bazy danych, odpowiedz na pytanie: *Co to jest baza danych?*

2.1. Bazy danych i systemy nimi zarządzające

Bazy danych służą do przechowywania zbiorów danych. Zapewniają szybki i wygodny dostęp do danych oraz możliwość ich przetwarzania. Baza danych (ang. *Data Base, Database*) ma określoną strukturę, format i reguły poprawności.

Do obsługi bazy danych wykorzystuje się system zarządzania bazą danych (ang. *Database Management System*), który zapewnia możliwość projektowania i tworzenia nowych baz danych, dostęp do danych, administrowanie nimi oraz wyszukiwanie potrzebnych informacji.

Systemy zarządzania bazą danych spotykane są więc wszędzie tam, gdzie jest potrzeba zarządzania danymi i zapewnienia szybkiego dostępu do nich. Każdy z nas spotyka się z systemami zarządzania bazą danych, rezerwując bilet lotniczy czy kolejowy, wykonując operację bankową czy odwiedzając urząd ewidencji ludności.

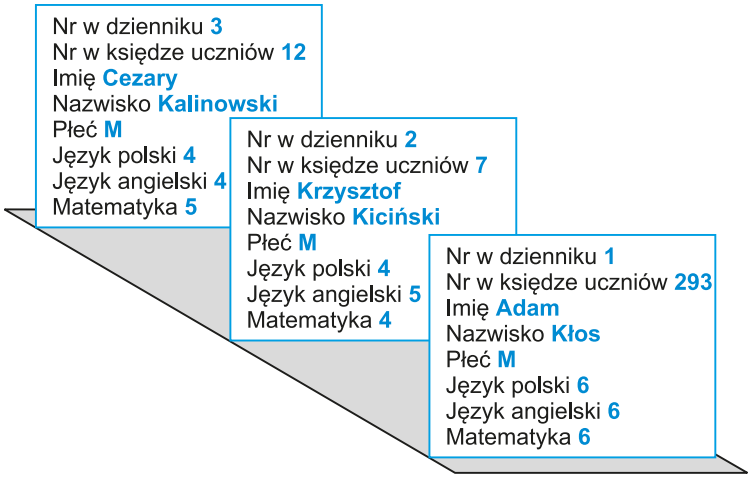
Podstawowe operacje na danych obejmują:
- wprowadzanie danych,
- zapamiętywanie i przechowywanie danych,

- wyszukiwanie i prezentację danych,
- aktualizację danych,
- przetwarzanie danych.

Do podstawowych zadań systemu zarządzania bazą danych należy zaliczyć:
- umożliwienie projektowania i tworzenia bazy danych,
- umożliwienie dostępu do danych,
- umożliwienie wykonywania operacji na danych,
- zapewnienie integralności danych,
- zapewnienie dostępu do danych dla wielu użytkowników,
- zapewnienie komunikacji między różnymi systemami.

Znane są różne *typy baz danych*, a ich podział wynika z zastosowanego w nich *modelu reprezentacji danych*, czyli z wewnętrznej struktury baz danych. Stosuje się podział na dwie grupy, w których można wyróżnić między innymi następujące typy baz:

1. Bazy proste:
- *Kartotekowa baza danych*, z której korzystają liczne programy typu: książka telefoniczna, spis płyt, książek. Przykładowa baza kartotekowa pokazana jest na rysunku 2.1. Każdy *rekord* (uczeń) zapisany jest na oddzielnym *arkuszu* (karcie). Pozwala to na sortowanie rekordów (arkuszy) według dowolnego kryterium, przeszukiwanie, filtrowanie ograniczające zakres wyświetlanych danych. Jeżeli do tej tabeli zechcemy dodać informacje dotyczące nauczyciela wychowawcy, to dla wielu uczniów dane te będą się powtarzały. Ponieważ dane kartotekowej bazy danych gromadzone są w jednej tabeli (zob. tabela 2.1), taką bazę danych można określić terminem *płaska*.



Rysunek 2.1. Kartotekowa baza danych

Tabela 2.1. Kartotekowa baza danych

Nr w dzienniku	Nr w księdze uczniów	Imię	Nazwisko	Płeć	Język polski	Język angielski	Matematyka
1	293	Adam	Kłos	M	6	6	6
2	7	Krzysztof	Kiciński	M	4	5	4
3	12	Cezary	Kalinowski	M	4	4	5
4	9	Krzysztof	Jankowski	M	4	5	6
5	17	Agata	Hoppa	K	4	3	4
6	4	Paweł	Haliniak	M	4	5	3
7	213	Brygida	Gazda	K	4	3	6

- *Hierarchiczna baza danych* (ang. *hierarchical database*), w której zastosowano drzewiastą strukturę danych o wielu gałęziach. Wszystkie elementy danych są logicznie zorganizowane, a każda wartość obiektu danych jest powiązana z jedną lub z kilkoma wartościami innego obiektu danych.

2. Bazy złożone:

- *Relacyjna baza danych* (ang. *relational database*), w której tabele są zbiorami rekordów powiązanych za pomocą związków (relacji) zachodzących pomiędzy danymi obsługiwanymi przez system zarządzania bazą danych (ang. *Database Management Systems*, *DBMS*). Ten typ baz danych zostanie omówiony szerzej.
- *Sieciowa baza danych*, w której elementy danych są zorganizowane w strukturę drzewiastą, podobnie jak w przypadku modelu hierarchicznego.
- *Obiektowa baza danych*, która przechowuje obiekty, czyli dane o dowolnej strukturze, zdefiniowanej przez projektanta.
- *Obiektowo-relacyjna baza danych*, która ma wszystkie cechy bazy relacyjnej, rozszerzonej o możliwości bazy obiektowej — umożliwia przechowywanie danych multimedialnych, przestrzennych, a nawet abstrakcyjnych. Można również spotkać określenie *multimedialna baza danych*.

Definicja

Baza danych to zbiór danych zgromadzonych zgodnie z określonym porządkiem (strukturą), dostępnych według kryteriów umożliwiających ich wyszukiwanie i obsługiwanych przez *system zarządzania bazą danych*.

Definicja

System zarządzania bazą danych to odpowiednio zorganizowane, specjalistyczne oprogramowanie narzędziowe pozwalające na wykonywanie operacji na danych.

Najbardziej powszechnymi systemami zarządzania bazami danych są systemy relacyjnych baz danych. Do najpopularniejszych należy zaliczyć profesjonalne systemy:

Oracle Universal Server (Oracle), DB2 (IBM), PostgreSQL (PostgreSQL Global Development Group), Sybase SQL Server (Sybase) oraz Access z pakietu biurowego MS Office (Microsoft). Inne popularne systemy zarządzania to: Microsoft SQL Server (Microsoft), MySQL (Sun Microsystems) oraz SQLite autorstwa Richarda Hippa.

Działanie systemu zarządzania bazą danych omówimy na przykładzie programu Access z pakietu biurowego MS Office.

2.2. Projektowanie

W dużych, profesjonalnych bazach danych każdy użytkownik ma inne zadania i uprawnienia. I tak, *zleceniodawca* zleca opracowanie bazy, określając jej przeznaczenie, charakter przechowywanych informacji, atrybuty gromadzonych danych, liczbę użytkowników. *Projektanci* projektują schemat i wygląd bazy, opracowują mechanizm wydobywania informacji z danych. Wstępnie przygotowana baza jest sprawdzana i testowana przez *analityków* badających jej własności. Na koniec *administratorzy*, czuwający nad bezpieczeństwem gromadzonych danych, przydzielają uprawnienia zwykłym użytkownikom, dzieląc ich na użytkowników z prawem dokonywania zmian (modyfikacji, uaktualniania) oraz dodawania danych i na użytkowników zwykłych — jedynie z prawem odczytu wybranych danych.

W związku z tym, przystępując do projektowania bazy, musimy wiedzieć, jakim celom będzie ona służyła, jakie informacje będzie przechowywała, ile osób będzie ją użytkowało i jakie uprawnienia należy im przydzielić. Należy przemyśleć i szczegółowo zaplanować sposób gromadzenia danych, aby można było na ich podstawie otrzymać pożądane informacje.

2.2.1. Zasady projektowania bazy danych

Początek pracy nad bazą danych jest zawsze taki sam. Powinniśmy zadać sobie pytania:

- Jakiego rodzaju informacje będziemy przechowywać w bazie?
- Czego oczekujemy od bazy?
- Jakie informacje będziemy pobierać z bazy?

Przystępując do projektowania, zawsze należy pamiętać o trzech podstawowych regułach, które są omówione poniżej.

Unikanie nadmiarowości danych

Dane zajmują miejsce na nośniku i w pamięci komputera. Wraz ze wzrostem ich ilości wydłuża się czas wykonywania operacji. Należy zatem — w miarę możliwości — wyeliminować powtarzanie danych w bazie. Można tego dokonać, zamieszczając dane w kilku tabelach zamiast w jednej dużej tabeli. Posłużmy się przykładem optyka, który realizuje recepty na okulary. Może on wszystkie niezbędne dane zgromadzić w jednej tabeli. Będzie ona miała dużo kolumn, w których szereg danych się powtarza. W tabeli 2.2 zamieszczono jedynie mały fragment takiej tabeli z zestawieniem wykonanych usług.

Tabela 2.2. Realizacja recept

Nr_usługi	PESEL	Nazwisko pacjenta	Imię pacjenta	Kod	Miejscowość	Ulica	Data urodzenia pacjenta	Wykonana usługa	Liczba wykonanych usług
1	34082305835	Wiśniewska	Stanisława	87-100	Toruń	Rydygiera	1934-08-23	sferyczne od 0,00 do ±4,00 dpt.	1
2	34082305835	Wiśniewska	Stanisława	87-100	Toruń	Rydygiera	1934-08-23	toryczne: sfera od 0,00 do ±6,00 dpt., cylinder do +2,00 dpt.	1
3	34082305835	Wiśniewska	Stanisława	87-100	Toruń	Rydygiera	1934-08-23	sferyczne od 0,00 do ±4,00 dpt.	1
4	34082305835	Wiśniewska	Stanisława	87-100	Toruń	Rydygiera	1934-08-23	toryczne: sfera od 0,00 do ±6,00 dpt., cylinder do +2,00 dpt.	1
5	45031509896	Miller	Irena	88-100	Inowrocław	Matejki	1945-03-15	sferyczne od 0,00 do ±4,00 dpt.	1
6	45031509896	Miller	Irena	88-100	Inowrocław	Matejki	1945-03-15	sferyczne od ±4,25 do ±6,00 dpt.	1
7	45031509896	Miller	Irena	88-100	Inowrocław	Matejki	1945-03-15	toryczne: sfera od 0,00 do ±6,00 dpt., cylinder do +2,00 dpt.	2

Przy każdej realizacji recepty na okulary, w których każde szkło okularowe jest innego typu i w innej cenie, powtarzane są dane osoby dokonującej zakupu. Powtarzają się również informacje o typie i parametrach szkła korekcyjnego przy każdej pozycji realizowanej recepty. Informacje te są rozbudowane i opisowe. Opis jest możliwie najkrótszy, ale mimo to zajmuje od 20 do 40 znaków. Na przykład dane osobowe pani Stanisławy Wiśniewskiej powtórzono aż czterokrotnie, podając wszystkie niezbędne informacje.

Należy zatem wyszukać niepotrzebnie powtarzające się w bazie informacje i umieścić je w oddzielnej tabeli lub w oddzielnych tabelach. Aby wyeliminować powtarzanie danych, należy utworzyć dwie dodatkowe tabele: *Pacjenci* oraz *Asortyment*.

Tabela *Pacjenci* będzie zawierała dane osobowe pacjentów (tabela 2.3). W tej tabeli w kolejnych miesiącach pracy będą dopisywani nowi pacjenci, natomiast ci, którzy przyjdą ponownie, zostaną wyszukani w istniejącej już bazie i do rozliczenia użyty zostanie ich *ID_pacjenta*.

Tabela 2.3. Pacjenci

ID_pacjenta	PESEL	Nazwisko	Imię	Data urodzenia	Kod	Miejscowość	Ulica
327	34082305835	Wiśniewska	Stanisława	1934-08-23	87-100	Toruń	Rydygiera
672	45031509896	Miller	Irena	1945-03-15	88-100	Inowrocław	Matejki
677	45040300421	Krawiec	Józef	1945-04-03	87-100	Toruń	Sienkiewicza
807	47081205754	Witkowska	Wiesława	1947-08-12	87-100	Toruń	Stamma
905	49010413467	Malec	Ryszard	1949-01-04	87-800	Włocławek	Reja
1200	53013106811	Sitkowska	Bożena	1953-01-31	87-100	Toruń	Matejki
2197	91091906260	Bednarczyk	Tomasz	1991-09-19	87-100	Toruń	Rydygiera

Korzyści z unikania nadmiarowości danych są dwojakie:

- dane osobowe zajmują mniej miejsca w bazie, ponieważ teraz występują tylko raz;
- użytkownik bazy, któremu nadano uprawnienia do dodawania i edycji danych, ma mniej pracy.

Tabela *Pacjenci* będzie się rozrastała. Należy zatem już na etapie projektowania przemyśleć sposób kodowania osób. W omawianym przykładzie użyto kolejnego numeru. Posortowanie danych pacjentów według *ID_pacjenta* pokazuje kolejność wprowadzania pacjentów do bazy.

Kosztem dodania nowego pola, *ID_pacjenta*, udało się rozdzielić tabelę. Pole to występuje zarówno w tabeli *Pacjenci*, jak i w tabeli *Realizacja* (tabela 2.5), która — po wyłączeniu z niej danych osobowych i adresowych pacjentów — została znacznie zmniejszona. Dodanie pola *ID_pacjenta* jest konieczne dla zachowania spójności danych. Powtarzana kolumna zawiera minimalną liczbę znaków — tylko identyfikator będący kolejnym numerem pacjenta realizującego receptę. Daje to wymierne korzyści, zwłaszcza w dużych bazach danych.

Natomiast w tabeli *Asortyment* będą zawarte informacje o ofercie wraz z cennikiem zakładu (tabela 2.4).

Tabela 2.4. Asortyment szkieł korekcyjnych z cennikiem

Nazwa handlowa	ID_szkła	Cennik
Sferyczne od 0,00 do ±4,00 dpt.	S1	6
Sferyczne od ±4,25 do ±6,00 dpt.	S2	7
Sferyczne od ±6,25 do ±10,00 dpt.	S3	19
Sferyczne powyżej ±10,00 dpt.	S4	32
Toryczne: sfera od 0,00 do ±6,00 dpt., cylinder do +2,00 dpt.	T1	14
Toryczne: sfera od 0,00 do ±6,00 dpt., cylinder od +2,25 do +4,00 dpt.	T2	19
Toryczne: sfera od 0,00 do ±6,00 dpt., cylinder powyżej +4,00 dpt.	T3	25
Toryczne: sfera powyżej ±6,00 dpt., cylinder — wszystkie	T4	30

Tabela *Asortyment* zawiera wszystkie dostępne szkła korekcyjne i ich ceny. Ta tabela może się wydawać dość duża i niedająca oszczędności, jednak należy pamiętać, że w każdym miesiącu realizowanych jest kilkaset recept, a na każdej recepcie są przynajmniej dwa szkła. Oszczędności są więc ogromne.

W tej tabeli ponownie — dzięki zastosowaniu *identyfikatora* szkła korekcyjnego — została zmniejszona liczba wprowadzanych liter, opisujących wykonaną usługę. W tym przypadku rodzajów szkieł jest 8, więc można by zastosować numer jako kod (identyfikator), jednak tu zostały zastosowane kody, które mają maksymalnie 2 znaki. W ogólnym rozrachunku jest to bardzo duża oszczędność wynikająca z powtarzania jedynie kodu, a nie całego opisu rodzaju szkła. Pole *ID_szkła* zostało dodane do obu tabel: *Asortyment* i *Realizacja*. Ostatecznie tabela 2.5 jest dużo mniejsza i z pewnością uniknięto w niej *nadmiarowości danych*.

Tabela 2.5. Realizacja recept po zastosowaniu identyfikatorów

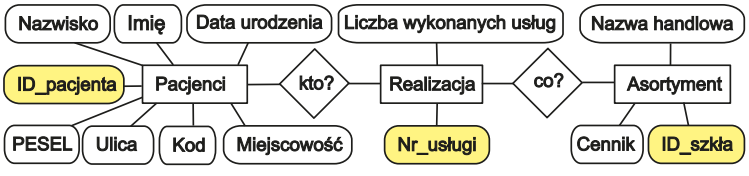
Nr_usługi	ID_pacjenta	ID_szkła	Liczba wykonanych usług
1	327	S1	1
2	327	T1	1
3	327	S1	1
4	327	T1	1
5	672	S1	1
6	672	S2	1
7	672	T1	2

Jednoznaczna identyfikacja danych

Tabela musi zawierać pewne unikatowe dane w celu jednoznacznej identyfikacji zgromadzonych danych.

Każdy rekord bazy danych musi różnić się od pozostałych. Jeżeli w tabeli występują takie same dane, powstaje pytanie, jak je odróżnić. Najprostszym i najczęściej stosowanym sposobem jest dodanie do tabeli kolumny zawierającej unikatowe dane. W celu odróżnienia danych można zastosować niepowtarzalny, kolejny numer, jak w tabeli *Realizacja*, lub specjalny kod pełniący funkcję identyfikatora, jak w tabeli *Asortyment*.

Można również przedstawić tę bazę w postaci tzw. *diagramu obiektowo-związkowego*. Na etapie projektowania narysowanie takiego diagramu pozwala dostrzec zależności i związki między poszczególnymi danymi. Na rysunku 2.2 pokazano diagram obiektowo-związkowy dla omawianej bazy zakładu optycznego, na którym wyraźnie widać pola jednoznacznie identyfikujące dane w każdej tabeli, czyli pola zawierające unikatowe dane.



Rysunek 2.2. Diagram obiektowo-związkowy dla bazy danych zakładu optycznego

W celu jednoznacznej identyfikacji danych można użyć nie tylko jednej kolumny danych, ale także większej ich liczby. Na przykład uczeń w szkole ma przypisany numer w dzienniku. Jednak dopiero po połączeniu numeru ucznia z oddziałem szkolnym można dokonać jednoznacznej identyfikacji każdego ucznia w szkole. W tym przypadku potrzebne są aż dwie kolumny (dwa pola) danych: *Nr_dziennika* i *Oddział*.

Czym jest klucz w bazie danych?

Definicja

Kluczem głównym, zwanym również **podstawowym** lub **własnym** (ang. *primary key*), może być ta kolumna lub zestaw kolumn tabeli, które zawierają niepowtarzalne informacje pozwalające jednoznacznie zidentyfikować rekord w tabeli.

Na rysunku 2.2 pola unikatowe w każdej z tabel, czyli: *Nr_usługi*, *ID_szkła* i *ID_pacjenta*, stają się *polami kluczowymi*. Taki klucz w każdej tabeli nosi nazwę *klucza własnego*.

Tabela może mieć jeszcze inne klucze, tzw. *klucze obce* (klucze własne innej tabeli). *Klucze obce* pozwalają spoić, czyli połączyć relacjami dane z różnych tabel bazy. Dzięki tym kluczom możliwe jest pobranie informacji o ofercie zakładu oraz przypisanie usługi do danego pacjenta. Po zestawieniu obok siebie wszystkich trzech tabel omawianego przykładu można z łatwością określić, które pola pełnią funkcję klucza własnego, a które są kluczami obcymi, czyli kluczami własnymi innych tabel.

Realizacja
Nr_usługi (klucz własny)
ID_pacjenta (klucz obcy)
ID_szkła (klucz obcy)
Liczba wykonanych usług

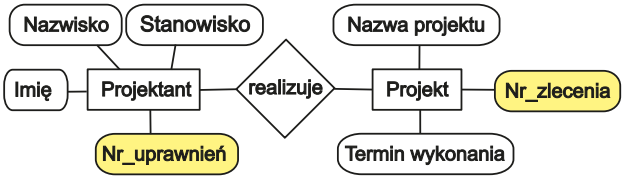
Asortyment
Nazwa handlowa
ID_szkła (klucz własny)
Cennik

Pacjenci
ID_pacjenta (klucz własny)
PESEL
Nazwisko
Imię
Data urodzenia
Kod
Miejscowość
Ulica

Korzystając z klucza, można zagwarantować spójność danych zawartych w kilku tabelach, a także sortować lub wyszukiwać potrzebne informacje.

Ćwiczenie 2.1. Zaprojektuj bazę danych pacjentów przychodni lekarskiej dla dzieci i dorosłych. W przychodni pracuje pięciu lekarzy — dwóch pediatrów i trzech lekarzy ogólnych dla dorosłych. Zapisując się do przychodni, każdy pacjent musiał wybrać swojego lekarza. Każdego dnia przychodnia musi złożyć raport o liczbie przyjętych pacjentów. Kierownik przychodni chciałby dowiedzieć się, do którego lekarza zapisało się najwięcej pacjentów i który lekarz miał danego dnia najwięcej wizyt. Wszystkie te informacje będą możliwe do uzyskania tylko przy prawidłowo zaprojektowanej bazie. Dokładnie przemyśl układ tabel, z których będzie zbudowana baza. Ustal klucze własne i klucze obce dla planowanych tabel.

Ćwiczenie 2.2. Na rysunku 2.3 przedstawiony został diagram obiektowo-związkowy. Na jego podstawie zaprojektuj bazę danych zawierającą odpowiednią liczbę tabel o odpowiedniej strukturze.



Rysunek 2.3. Diagram do ćwiczenia 2.2

Zastanów się, czy to zadanie ma tylko jedno rozwiązanie. Porównaj swoje rozwiązanie z innymi.

2.2.2. Projektowanie bazy danych w programie Microsoft Office Access 2007

Za pomocą programu Microsoft Office Access 2007 można zarządzać wszystkimi niezbędnymi informacjami, używając *jednego pliku bazy danych*. W takim pliku dane można dzielić na odrębne części, zwane *tabelami*. Za pomocą *formularzy* elektronicznych można oglądać, dodawać i aktualizować dane w tabelach. Za pomocą *kwerend* można znajdować i pobierać tylko te dane, które są rzeczywiście aktualnie niezbędne. Za pomocą *raportów* można analizować i drukować dane w określonym układzie.

W tym rozdziale opiszemy podstawowe operacje na przykładzie programu Microsoft Office Access 2007. Na płycie CD dołączonej do podręcznika, w dokumencie *r02 Open Office.pdf*, zamieszczone zostało szczegółowe wprowadzenie wraz z rysunkami — dzięki temu możliwe będzie wykonanie zawartych w podręczniku ćwiczeń i zadań w pakiecie biurowym OpenOffice.org 3.2, który może być zainstalowany zarówno w środowisku Windows, jak i Linux, na przykład Ubuntu.

Po uruchomieniu programu Office Access 2007 zostanie wyświetlona strona *Wprowadzenie do programu Microsoft Office Access*. Na tej stronie są wyświetlone instrukcje dotyczące postępowania przy rozpoczynaniu pracy w programie.

W relacyjnej strukturze danych informacje są logicznie podzielone na zbiory. Każdy zbiór jest umieszczany w osobnej tabeli.

Możliwe jest na przykład utworzenie nowej, pustej bazy danych, utworzenie bazy danych na podstawie szablonu lub otwarcie ostatnio używanej bazy danych. Można także przejść bezpośrednio do witryny Microsoft Office Online, aby uzyskać więcej informacji o pakiecie Microsoft Office 2007 i programie Office Access 2007, lub otworzyć istniejącą bazę danych.

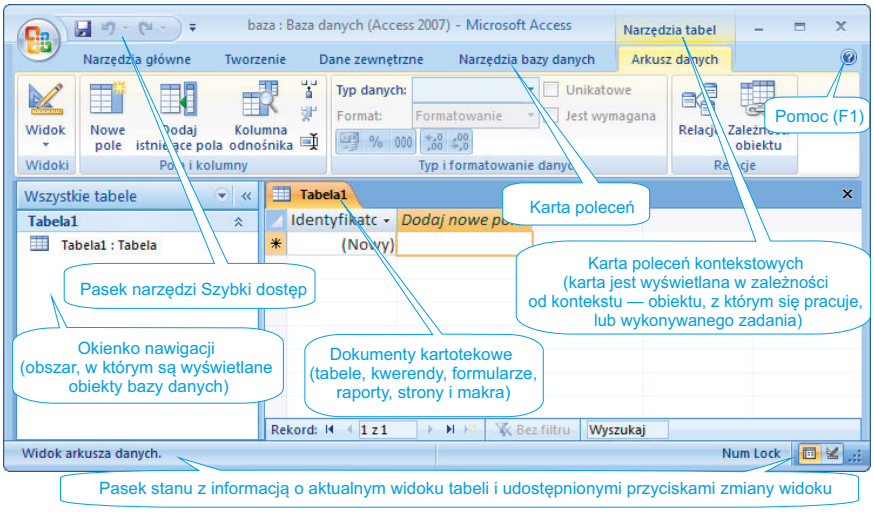
Na stronie *Wprowadzenie do programu Microsoft Office Access*, w obszarze *Nowa pusta baza danych*, kliknij przycisk *Pusta baza danych*. W okienku *Pusta baza danych*, w polu *Nazwa pliku*, wpisz nazwę pliku, na przykład *baza*, lub użyj nazwy sugerowanej. Program doda do nazwy pliku rozszerzenie *.accdb*. Kliknij przycisk *Utwórz*.

Zostanie utworzona nowa baza danych, a w widoku arkusza danych zostanie otwarta nowa tabela. Elementy okna nowej bazy danych pokazano na rysunku 2.4.

Po otwarciu lub utworzeniu nowej bazy danych nazwy obiektów bazy danych są wyświetlane w *okienku nawigacji*. Obiekty bazy danych to *tabele*, *formularze*, *raporty*, *strony*, *makra* i *moduły*.

Podczas pracy z programem Microsoft Office Access 2007 można w dowolnym momencie uzyskać pomoc, naciskając klawisz *F1* lub klikając ikonę ze znakiem zapytania po prawej stronie Wstążki.

Osoby, które rozpoczynają pracę z programem Office Access 2007, mogą skorzystać ze szkoleń zamieszczonych w oficjalnej witrynie *Office Online*.



Rysunek 2.4. Elementy okna bazy danych w MS Access 2007

2.3. Tabele

Definicja

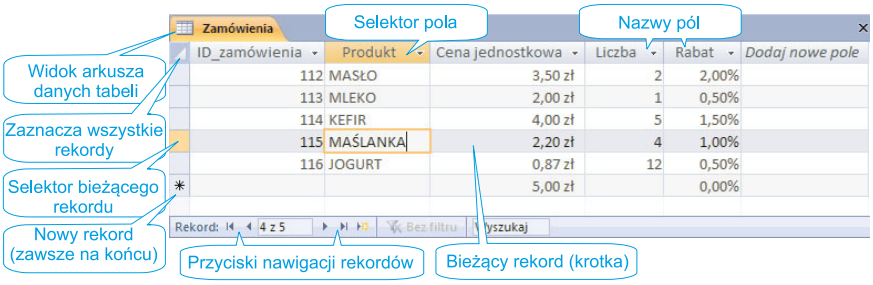
Tabela jest podstawowym obiektem relacyjnej bazy danych.

Wszystkie dane wprowadzane przez użytkownika są gromadzone w tabelach. Tabela przechowuje informacje na dany temat, powinna więc być poświęcona jednemu tematowi. Wszystkie dane dotyczące pojedynczego elementu umieszczone są w jednym wierszu, zwanym *rekordem* tabeli lub *krotką*. Używanie oddzielnych tabel dla każdego tematu eliminuje duplikowanie danych (*redundancję*), co czyni przechowywanie danych bardziej efektywnym, a pliki bazy zajmują mniej pamięci operacyjnej i mniejszą przestrzeń dyskową. Unikamy również zjawiska *anomalii*, które może wystąpić na przykład wtedy, gdy w bazie, w której kilkakrotnie występują dane klienta, należy zmienić jego dane adresowe, a my dokonamy zmian nie we wszystkich wpisach. Ta sama osoba będzie figurowała z dwoma różnymi adresami. Inną przyczyną powstania anomalii jest wpisanie w dwóch miejscach klienta Kowalskiego raz jako `Kowalski`, drugi raz jako `Koralski`. W tym przypadku nie wiadomo, która wersja jest poprawna. Oczywistą korzyścią jednokrotnego wpisywania danych jest oszczędność czasu.

2.3.1. Tworzenie tabeli w programie Access 2007

Najlepiej przechowywać dane, tworząc po jednej tabeli dla każdego typu informacji, które mają być analizowane. Dane z różnych tabel można później łączyć — należy w tym celu zdefiniować relacje między tabelami. Każda baza danych musi więc zawierać przynajmniej jedną tabelę.

Tabele wymuszają organizowanie danych w kolumnach (zwanych *polami* lub *atrybutami*) i w wierszach (zwanych *rekordami* lub *krotkami*). Na rysunku 2.5 pokazano układ danych w przykładowej tabeli.



Rysunek 2.5. Opis tabeli w widoku arkusza danych

Projektując tabele, musisz uwzględnić kilka zasad. Oto najważniejsze z nich:

- kolumny tabeli, nazywane polami lub atrybutami, muszą mieć jednoznaczne nazwy (nie mogą się powtarzać);
- w danej kolumnie muszą być wartości tego samego typu (nie można na przykład przechowywać tekstu w polu o typie danych *Liczba);*
- kolejność kolumn w tabeli nie ma wpływu na sposób działania bazy;
- w każdym wierszu wpisane są dane jednego rekordu (osoby, artykułu);
- kolejność wierszy (rekordów) może się zmieniać, na przykład w wyniku sortowania.

Zanim zaprojektujemy pierwszą bazę, przypomnijmy raz jeszcze, że powtarzanie tej samej informacji jest nieefektywne. *Redundancja* danych (niepotrzebne powtarzanie danych w tabeli czy w bazie danych):

- powiększa objętość bazy;
- zmniejsza efektywność obsługi bazy;
- powoduje zagrożenie nieuchronnymi błędami przy wprowadzaniu danych.

Wielokrotne powielanie danych, na przykład danych adresowych klienta, stwarza problemy z ich aktualizacją, które mogą prowadzić do powstania *anomalii* i kłopotów z identyfikacją danych.

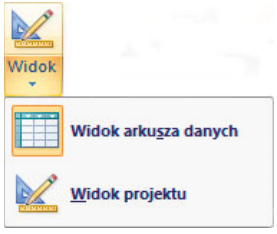
Sprawą podstawową jest wymóg, aby pojedyncze pole tabeli zawierało informację elementarną, jednostkową. Pole *Adres* należy rozbić na mniejsze, elementarne składowe adresu: miasto, kod, ulicę, nr domu. Pozwoli to na wyszukiwanie danych według kodu czy nazwy miasta.

Utworzymy przykładową bazę danych *Uczelnia* zawierającą dane studentów. Tworzenie bazy rozpoczniemy od tabel z danymi studentów oraz wykładowców Uniwersytetu Wirtualnego.

W projekcie bazy danych *Uczelnia* w celu ewidencjonowania studentów ich dane osobowe zapiszemy jako pojedynczy wiersz (rekord) w tabeli *Studenci_dane*. Poszczególne kategorie danych (data urodzenia, miasto, kod, ulica) stanowić będą pola tej tabeli. Będzie ona zawierała tyle wierszy (rekordów), ilu jest studentów.

Dane uczelniane o naszych studentach powinniśmy umieścić w osobnej tabeli, na przykład *Studenci_uczelnia*. Teraz każdy rekord, zarówno w tabeli *Studenci_dane*, jak i *Studenci_uczelnia*, musi zawierać pole identyfikujące studenta, tzw. *klucz*, który pozwoli połączyć dane osobowe z danymi uczelnianymi. Może to być pole będące numerem indeksu (albumu), czyli wpisu studenta do księgi uczelni. Numer ten jednoznacznie identyfikuje studenta.

Zacznijmy więc tworzyć nową bazę danych. Po uruchomieniu programu Office Access 2007 podajemy lokalizację pliku nowej, pustej bazy danych oraz jej nazwę: *Uczelnia*. Zostanie utworzona nowa baza danych, a w widoku arkusza danych zostanie otwarta nowa tabela o tymczasowej nazwie *Tabela1*, jak pokazano to już wcześniej na rysunku 2.4. Zwróć uwagę na dodatkową kartę *Arkusz danych*, czyli na kartę poleceń kontekstowych *Narzędzia tabel*, związaną z obsługą tabel. Tabelę można tworzyć w dwóch widokach: *projektu* i *arkusza danych*, jak to pokazano na rysunku 2.6. Jednak po jej utworzeniu zostaną dodane jeszcze dwa widoki, w których można pracować, są to *Widok tabeli przestawnej* oraz *Widok wykresu przestawnego*, które omówimy w dalszej części rozdziału.



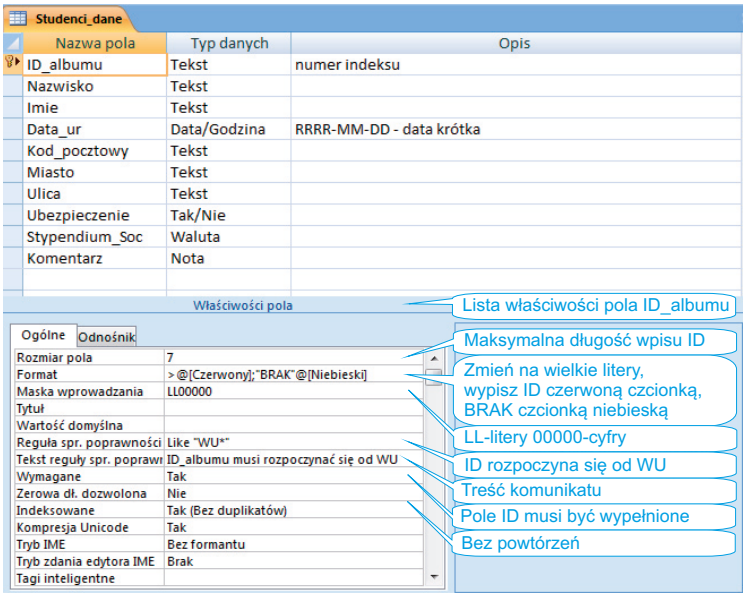
Rysunek 2.6. Różne widoki tworzenia tabeli

W widoku *arkusza danych* można dodawać, edytować lub wyświetlać dane znajdujące się w tabeli. Można także sprawdzać pisownię, drukować dane tabeli, filtrować lub sortować rekordy, zmieniać wygląd arkusza danych oraz zmieniać strukturę tabeli, dodając lub usuwając kolumny.

W widoku *projektu* można tworzyć tabelę od początku, a także dodawać, usuwać i zmieniać pola istniejącej tabeli oraz określać właściwości każdego z pól.

Ponieważ chcemy zaprojektować nową tabelę i samodzielnie określić nazwy i typy pól oraz ich właściwości, wybierzemy *Widok projektu*. Po podaniu nazwy tabeli (*Studenci_dane*) zmieni się widok tabeli oraz karta poleceń kontekstowych na *Projekt*. Automatycznie utworzone zostanie pole o nazwie *Identyfikator*, dla którego ustalono typ danych *Autonumerowanie*. Obok pola ustawiono znak klucza, oznaczając je jako *pole kluczowe*. Zmienimy tę nazwę na *ID_albumu* i typ danych na *Tekst*. Dodamy pozostałe pola tabeli: *Nazwisko*, *Imie*, *Data_ur*, *Kod_pocztowy*, *Miasto*, *Ulica*, *Ubezpieczenie*, *Stypendium_soc*, *Komentarz* (przyjmijmy zasadę, że nazwy pól tabeli nie zawierają spacji oraz polskich znaków). Widok projektu tabeli *Studenci_dane* pokazano na rysunku 2.7.

Dla każdego pola określimy *Typ danych*, wybierając go z listy rozwijanej. Po wyborze typu danych ustalimy, w dolnej części okna, *Właściwości pola*. W kolumnie *Opis* umieścimy komentarze dotyczące tworzonych pól. Pola *Nazwa pola* i *Typ danych* wypełnimy obowiązkowo dla każdej kolumny danych wprowadzanej do projektu. *Opis* sporządzimy opcjonalnie (nie jest konieczny — pełni funkcję komentarza).



Rysunek 2.7. Widok projektu tabeli Studenci_dane

2.3.2. Typy danych w tabelach

Definicja

Typ danych pola to cecha pola, która określa, jakiego rodzaju dane mogą być w nim przechowywane.

Na przykład w polu, którego typem danych jest *Tekst*, mogą być przechowywane dane składające się ze znaków zarówno tekstowych, jak i liczbowych, ale w polu typu *Liczba* mogą być przechowywane tylko dane liczbowe.

W tabeli *Studenci_dane* pole *Data_ur* jest typu *Data/Godzina*, pole *Ubezpieczenie* jest typu *Tak/Nie*, pole ze stypendium ma typ *Waluta*, wszystkie pozostałe to pola typu *Tekst* lub *Nota*.

Przy podejmowaniu decyzji, jakiego typu danych użyć w polu tabeli, należy uwzględnić poniższe czynniki.

- Jakie wartości będą dozwolone dla pola?
- Ile miejsca będą mogły zajmować dane przechowywane w polu?
- Jakie operacje będą wykonywane na wartościach w polu?
- Czy wartości w polu będą sortowane lub indeksowane?
- W jaki sposób wartości pola będą sortowane?
- Czy pole będzie używane do grupowania rekordów w kwerendach lub raportach?
- Czy w danym polu będą przechowywane dokumenty, obrazy, dźwięki i dane binarne innego typu utworzone przy użyciu programów Microsoft Office Word, Excel lub innych?

Poniżej przedstawione są wszystkie typy danych pól dostępne w programie Access, ich zastosowanie i rozmiary.

Tekst

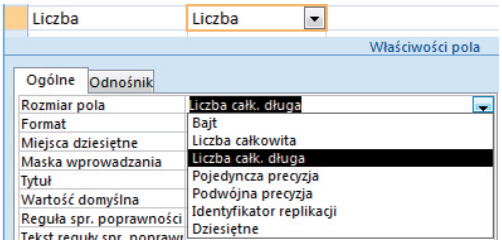
Pól typu *Tekst* należy używać do przechowywania takich danych jak nazwy czy adresy oraz danych liczbowych, na których nie będą dokonywane żadne obliczenia (na przykład: numery telefonów, numery części czy kody pocztowe). Pole tekstowe może zawierać do 255 znaków, jednak domyślnie mieści 50 znaków. Aby umożliwić wprowadzenie do pola większej ilości danych, należy zmienić ustawienie właściwości *Rozmiar pola*, określając maksymalną liczbę znaków, które można wprowadzić.

Nota

Pola typu *Nota* są używane do przechowywania danych dłuższych niż 255 znaków, czyli długich tekstów i liczb, na przykład notatek i opisów. Pole *Nota* może zawierać do 65 536 znaków (64 kB). Do przechowywania sformatowanych tekstów i długich dokumentów należy użyć pola typu *OLE* zamiast pola typu *Nota*.

Liczba

Typ używany dla danych liczbowych, na których są przeprowadzane obliczenia matematyczne, z wyjątkiem obliczeń walutowych (tu należy użyć pola typu *Waluta*). Konkretny rodzaj pola typu *Liczba* jest definiowany przez właściwość *Rozmiar pola*, jak pokazano na rysunku 2.8. Pole *Liczba* może przechowywać 1 bajt, 2 bajty, 4 bajty, 8 bajtów, a nawet 16 bajtów w przypadku identyfikatora replikacji (GUID).



Rysunek 2.8. Zakresy i typy liczb wprowadzanych do pola

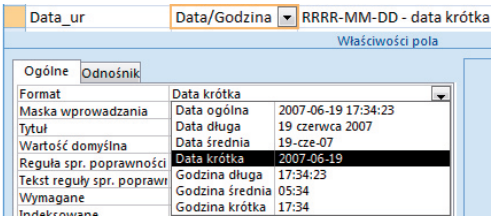
Najczęściej wybierane typy liczby to:

- *Bajt* — jest to liczba całkowita dodatnia zajmująca jeden bajt, czyli osiem bitów przeznaczonych na liczbę — bez bitu znaku. Najmniejszą zapisaną liczbą w tym formacie jest 0, a największą 255.
- *Liczba całkowita* — jest to liczba całkowita zapisana na dwóch bajtach, z czego pierwszy bit przeznaczony jest na znak liczby, a pozostałe 15 bitów na zapis modułu liczby. Jest to odpowiednik znanego z języków programowania typu *Integer* i ma taki sam zakres. Liczby mieszczą się w przedziale od –32 768 do 32 767.
- *Liczba całkowita długa* — jest to liczba całkowita zapisana na czterech bajtach, stąd jej wartość mieści się w przedziale od –2 147 483 648 do 2 147 483 647. Jest to odpowiednik typu *Longint*.
- *Pojedyncza precyzja* — jest to liczba rzeczywista zapisana na czterech bajtach, w której wyróżniamy część całkowitą i część ułamkową.
- *Podwójna precyzja* — jest to liczba rzeczywista zapisana na ośmiu bajtach, w której wyróżniamy część całkowitą i część ułamkową.

Ze względu na tak dużą różnorodność, rozpiętość zakresów i rozmiary danych typu *Liczba* należy na etapie projektowania ustalać typy danych liczbowych zgodnie z ich przeznaczeniem.

Data/Godzina

Typ pola używany dla dat i godzin. Przechowuje 8 bajtów. Można wybrać kilka sposobów wyświetlania daty i godziny, co ilustruje rysunek 2.9.



Rysunek 2.9. Sposoby wyświetlania formatu Data/Godzina

Waluta

Używany dla wartości pieniężnych. Wartości tego typu są podawane w ustalonej walucie. Typ *Waluta* stosowany jest dla wartości walut oraz w celu zapobiegania zaokrąglaniu podczas obliczeń. Przechowuje 8 bajtów. Innym sposobem na uzyskanie takiego pola jest stworzenie pola *Liczba* i ustawienie formatu jako *Waluta*. Jest to bowiem *Liczba* z włączonym formatowaniem wydruku w postaci waluty. Rysunek 2.10 pokazuje różne sposoby wyświetlania formatu *Waluta*.



Rysunek 2.10. Sposoby wyświetlania formatu Waluta

Autonumerowanie

Typ danych pola przechowujący unikatową liczbę automatycznie przydzielaną rekordowi w chwili jego dołączania do tabeli. Jest to niepowtarzalna sekwencja o przyroście równym 1 (*Przyrostowy*) lub są to liczby losowe wstawiane automatycznie podczas dodawania rekordu (*Losowy*) rozpoczynającego się od wartości *l*. *Autonumerowanie* jest najczęściej wykorzystywanym typem w polach będących kluczami tabeli. Przechowuje 4 bajty.

Tak/Nie

Pola zawierające tylko jedną z dwóch możliwych wartości, na przykład: *Tak/Nie*, *Prawda/Fałsz*, *Wł/Wył*. Wartości *Null* nie są dozwolone[1]. Przechowuje 1 bit. Wpisując wartości do tabeli w widoku *arkusza danych*, dla pola typu *Tak/Nie* można podawać wartości „tak" lub „nie", ale i *1* — odpowiadające wartości „tak" lub *0* — odpowiadające wartości „nie".

Obiekt OLE

Typ danych pola stosowany dla *obiektów OLE* (ang. *Object Linking and Embedding*) utworzonych w innych aplikacjach (na przykład: dokumentów Microsoft Word, arkuszy kalku-

[1] **Null** to wartość używana w polach tabel lub w wyrażeniach i kwarendach, oznaczająca dane brakujące lub nieznane (nie mylić z *0* lub *spacją*). Pola *klucza podstawowego* nie mogą zawierać wartości *Null*. Wartość *Null* może sprawiać kłopot użytkownikom baz danych. Oznacza ona brakującą lub nieznaną wartość albo informuje, że wartość nie ma zastosowania w bieżącym kontekście.
Rozważmy na przykład pole *NR_tel*, które miałoby zawierać numer telefonu studenta. Jeżeli numer telefonu danego studenta jest nieznany, wartość atrybutu (pola) *NR_tel* wynosi *Null*. Nie oznacza to, że student nie ma numeru telefonu, ale że nie wiadomo, czy numer istnieje, czy też nie. Jeżeli wiedzielibyśmy, że dany student nie ma numeru telefonu, informacja nie byłaby już nieznana. W tym przypadku wartością atrybutu *NR_tel* byłby *pusty łańcuch* znaków lub łańcuch znaków *Brak telefonu*, a nie wartość *Null*. Zatem to, czy wartość *Null* jest właściwą wartością, zależy od kontekstu.

lacyjnych Microsoft Excel, obrazów, dźwięków lub innych danych binarnych) za pomocą *protokołu OLE*, dzięki czemu można współużytkować informacje przez łączenie i osadzanie obiektów. Źródłem obiektu, nazywanym serwerem OLE, może więc być dowolna aplikacja, która obsługuje łączenie i osadzanie obiektów. Łączenie polega na utworzeniu łącza między obiektem OLE i jego serwerem OLE lub między dokumentem źródłowym i dokumentem docelowym dynamicznej wymiany danych. Osadzanie polega na wstawianiu kopii obiektu OLE z innej aplikacji. Zmiany wprowadzone w obiekcie osadzonym nie są odzwierciedlane w obiekcie oryginalnym.

Pól typu *Obiekt OLE* nie można sumować, sortować ani indeksować, nie można używać ich do grupowania rekordów. Rozmiar obiektu może wynosić maksymalnie 1 gigabajt (ograniczeniem jest też miejsce na dysku).

Załącznik

Załączniki można dołączać tylko do baz danych utworzonych za pomocą programu Office Access 2007, które to bazy używają formatu pliku *.accdb*. Załączniki pozwalają przechowywać wiele plików (także pliki różnych typów) w jednym polu. Można na przykład przechowywać obrazy i pliki utworzone za pomocą edytorów tekstu i arkuszy kalkulacyjnych. Ważną cechą tego nowego typu danych jest możliwość wyświetlania i edytowania dołączonych plików. Pola załączników zapewniają większą elastyczność niż pola typu *Obiekt OLE* i efektywniej wykorzystują miejsce na dysku. Pojedynczy załączony plik nie może być większy niż 256 MB, a wszystkie załączniki nie mogą przekroczyć wartości 2 GB, gdyż taki jest maksymalny rozmiar pliku bazy danych programu Access. Załączone pliki można zapisywać na dysku twardym lub w sieci.

Uwaga! Nie można przekonwertować pola typu *Załącznik* na inny typ danych, natomiast możliwe jest usunięcie pola w przypadku popełnienia błędu.

Hiperłącze

W polach tego typu można przechowywać hiperłącza do witryn lub plików w intranecie lub sieci lokalnej oraz witryn lub plików na komputerze użytkownika.

Podczas ustawiania typów danych w widoku projektu jest widoczna jeszcze jedna opcja: *Kreator odnośników*. Ta opcja nie oznacza typu danych. Za pomocą *Kreatora odnośników* można tworzyć pola odnośników, które łączą się z innymi tabelami, pozwalając podczas wprowadzania danych do tabeli na wybór danych z listy. Używanie odnośników jest nie tylko wygodne, ale daje również pewność uniknięcia błędnie wprowadzanych danych do pola. Domyślnie program Access ustawia dla pól odnośników typ danych *Liczba*.

2.3.3. Podstawowe właściwości pól tabeli

Ustalenie właściwości pola używane jest między innymi do określenia sposobu wyświetlania danych lub wykonywania określonych akcji (na przykład wypisywania komunikatów o błędnie podawanych danych). Dla każdego pola możemy ustalić jego właściwości, jednak lista dostępnych właściwości jest zmienna i zależy od typu danych pola.

Rozmiar pola

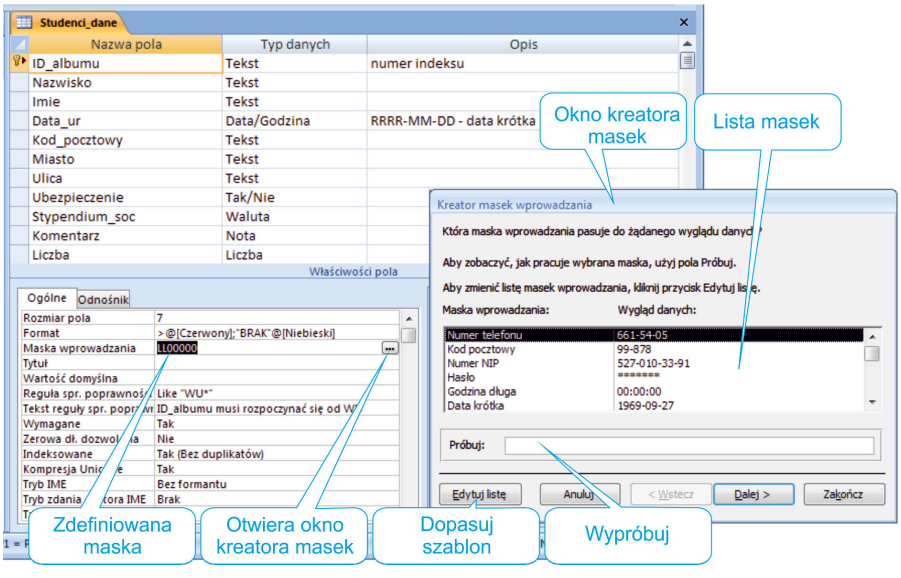
Dla danych typu *Tekst* określa maksymalną możliwą do wprowadzenia liczbę znaków. Przy próbie wprowadzenia dłuższego tekstu znaki nie są już przyjmowane.

Dla danych typu *Liczba* właściwość *Rozmiar pola* oznacza wybór typu liczby pokazany na rysunku 2.8 i opisany wcześniej na s. 48.

Ograniczenie długości tekstu do podanej wartości ma bardzo duże znaczenie ze względu na optymalizację objętości bazy danych, ale stosowane jest również w celu uniknięcia błędów podczas wprowadzania ustalonej długości danych tekstowych, na przykład kodów. Wybór odpowiedniego typu liczby pozwala na dokładne obliczenia, a także na zmniejszenie objętości bazy do minimalnych, niezbędnych rozmiarów.

Maska wprowadzania

Określa wzorzec dla wprowadzanych danych (na przykład format daty, numeru telefonu itp.). Maskę wprowadzania można ustalić dla pól typu *Tekst* lub *Data/Godzina* — to ułatwia wpisywanie i formatowanie danych oraz pozwala uniknąć błędów w przypadku wprowadzania danych numerycznych o specyficznej składni, takich jak: numer telefonu, numer konta bankowego, kod pocztowy, numer PESEL, NIP i inne. Istnieje możliwość skorzystania z gotowej maski dostępnej w postaci przygotowanych już szablonów, pokazanych na rysunku 2.11, lub samodzielnego ustalenia maski przez edycję gotowych szablonów.



Rysunek 2.11. Lista szablonów kreatora masek

Można też stworzyć własny projekt maski dostosowany do aktualnych potrzeb. W oknie kreatora obok nazw gotowych masek widać podgląd danych. Przyjrzyj się przykładowym sposobom użycia maski wprowadzania. Zwróć uwagę na zapis maski we właściwościach pola. Pomoże to w zrozumieniu symboli zastosowanych w maskach.

Program Microsoft Access interpretuje znaki definicji właściwości *Maska wprowadzania* w sposób przedstawiony w tabeli 2.6.

Tabela 2.6. Prawidłowe znaki dla masek wprowadzania

Symbol	Opis
`0`	Cyfra (od 0 do 9, pozycja wymagana; znaki plus [+] i minus [–] nie są dozwolone)
`9`	Cyfra lub spacja (pozycja opcjonalna; znaki plus i minus nie są dozwolone)
`#`	Cyfra lub spacja (pozycja opcjonalna; puste miejsca konwertowane są na spacje, znaki plus i minus dozwolone)
`L`	Litera (od A do Z, pozycja wymagana)
`?`	Litera (od A do Z, pozycja opcjonalna)
`A`	Litera lub cyfra (pozycja wymagana)
`a`	Litera lub cyfra (pozycja opcjonalna)
`&`	Dowolny znak lub spacja (pozycja wymagana)
`C`	Dowolny znak lub spacja (pozycja opcjonalna)
`. , : ; - /`	Dziesiętny symbol zastępczy oraz separatory tysięcy, daty i godziny. (Znak, który faktycznie zostanie użyty, zależy od ustawień regionalnych określonych w *Panelu sterowania* systemu Microsoft Windows)
`<`	Powoduje, że wszystkie następujące po nim znaki są konwertowane na małe znaki
`>`	Powoduje, że wszystkie następujące po nim znaki są konwertowane na wielkie znaki
`!`	Powoduje, że znaki w masce wprowadzania są wyświetlane od strony prawej do lewej zamiast od lewej do prawej. Znaki wpisywane w masce wprowadzania zawsze wypełniają ją od strony lewej do prawej. Znaku wykrzyknika można użyć w dowolnym miejscu w masce wprowadzania
`\`	Powoduje, że kolejny znak jest wyświetlany jako znak literałowy. Jest on używany w celu wyświetlenia dowolnego znaku uwzględnionego w tej tabeli jako znak literałowy (na przykład ciąg `"\A"` jest wyświetlany jako *„A”*)
`Hasło`	Ustawienie właściwości *Maska wprowadzania* na wartość *Hasło* powoduje utworzenie pola tekstowego dla hasła. Dowolny znak wpisany w tym polu jest przechowywany jako znak, ale wyświetlany jako gwiazdka (`*`)

Ćwiczenie 2.3. Jakie wartości można wpisać w polu, dla którego określono następującą maskę: `900\-00\-00`? W jaki sposób zostanie wyświetlone pole oczekujące na podanie wartości?

Na pozycji numer 1 należy wpisać cyfrę lub nic (`9`), na pozostałych pozycjach obowiązkowo cyfry (`0`). Zapis `\-` oznacza, że w polu pojawi się, oczekując na podanie numeru, następujący szablon: `"  -  -  "`. Wpisanie liczb do takiego pola jest wygodne i czytelne. Podczas wpisywania nie musimy dodawać myślników, które już tam są, podajemy jedynie liczby. Ta maska oznacza numer telefonu, który może składać się z 6 lub 7 cyfr.

Obecnie wszystkie numery telefonów mają 9 cyfr. Utwórz maski wprowadzania numeru podawanego na dwa różne sposoby, na przykład 123-456-789 i 12-345-67-89.

Wartość domyślna

To wartość wpisywana automatycznie przy wypełnianiu nowych rekordów tabeli. Tę właściwość danych wypełniamy opcjonalnie. Jeśli na przykład większość studentów jest ubezpieczona, w polu *Ubezpieczenie*, w którym ustalono typ danych *Tak/Nie*, jako wartość domyślną podajemy *Tak*, co znacznie upraszcza i przyspiesza wprowadzanie danych do tabeli, gdyż tylko dla nielicznych, nieubezpieczonych studentów trzeba zaznaczyć opcję *Nie*.

Reguła sprawdzania poprawności

Jest to wyrażenie, które ogranicza wartości wprowadzane do tabeli, uniemożliwiając podanie danych niespełniających określonego kryterium. Ograniczenie skali podawanych ocen (ocena ucznia nie może mieć wartości większej niż 6 ani mniejszej niż l) pozwoli uniknąć błędów podczas wpisywania danych.

Reguła sprawdzania poprawności pola używana jest do sprawdzania wartości wprowadzonej do pola w momencie opuszczania pola przez użytkownika. Te reguły nie mogą odwoływać się do innych pól. Dzięki zastosowaniu sprawdzania poprawności unikniemy wielu błędów, które mogłyby się pojawić w bazie w wyniku pomyłkowego wprowadzenia danych.

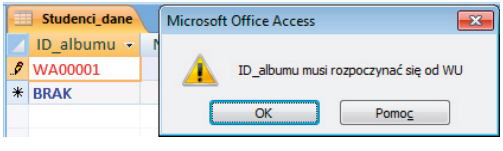
Przykłady zastosowania reguł poprawności:

1. `<>0` Należy wprowadzić wartość różną od zera.
2. `<0 Or >100` Wartość musi być mniejsza od 0 lub większa od 100.
3. `>=1 And <=6` Wartość musi być z zakresu szkolnej skali ocen.
4. `<#2008-01-01#` Należy wprowadzić datę sprzed roku 2008.
5. `>=#1/1/2008#And <=#31/12/2008#` Należy wprowadzić datę z roku 2008 (zwróć uwagę na sposoby zapisu dat zastosowane w punktach 4. i 5. — możesz ich używać wymiennie).

W regule poprawności pola można również używać znaków symboli wieloznacznych. W naszym przykładzie użyto reguły poprawności `Like "WU*"` — przy zatwierdzaniu treści pola nastąpi sprawdzenie, czy dwie pierwsze litery to `WU` (pozostały ciąg znaków jest dowolnej długości i treści).

Komunikat o błędzie

Wprowadzony w tym miejscu tekst pojawi się jako treść komunikatu wyświetlanego przy próbie wprowadzenia wartości niezgodnej z określoną powyżej regułą poprawności, jak na rysunku 2.12.



Rysunek 2.12. Komunikat wyświetlany po wprowadzeniu niepoprawnych danych

Wymagane

Ta właściwość określa, czy wypełnienie pola jest wymagane. Jeśli wartość właściwości *Wymagane* ustawimy na *Tak*, w puste pola **nie może** być już wstawiana wartość *Null*, czyli wartość, która ma wskazywać brakującą lub nieznaną wartość pola, szczegółowo opisaną wcześniej. Pole *Wymagane* **musi** być wypełnione treścią.

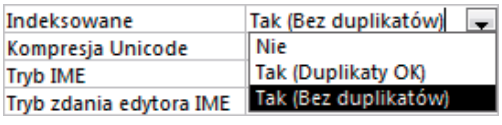
Przeanalizuj naszą tabelę *Studenci_dane* i zastanów się, które pola powinny mieć ustawienie *Tak* dla właściwości *Wymagane*.

Zerowa długość dozwolona

Właściwość ta określa, czy w polu można umieścić ciąg znaków o zerowej długości. Ciągiem znaków o zerowej długości jest ciąg: "".

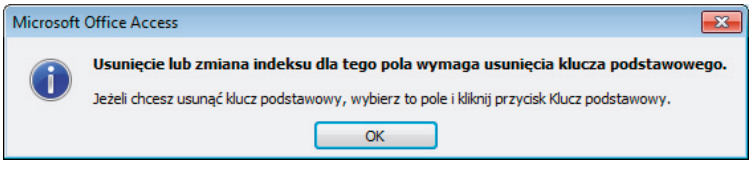
Indeksowane

Indeks znacznie przyspiesza wyszukiwanie i sortowanie danych, zwłaszcza w dużych bazach, ale może spowodować spowolnienie aktualizacji. Ma trzy opcje — pokazano je na rysunku 2.13. Wybranie *Tak (Bez duplikatów)* nie pozwoli na powtórzenie wartości w polu.



Rysunek 2.13. Dostępne opcje właściwości indeksowania pól

Dla pola kluczowego ta wartość jest ustawiana automatycznie, a próba jej zmiany wiąże się z koniecznością zrezygnowania z ustanowienia danego pola *polem kluczowym*. Pojawia się komunikat widoczny na rysunku 2.14.

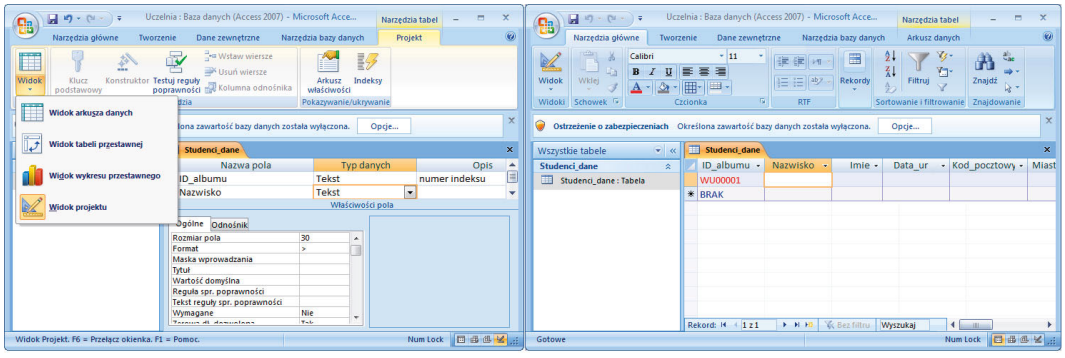


Rysunek 2.14. Komunikat dotyczący konieczności indeksowania pola kluczowego

Podczas projektowania tabeli możemy w dowolnej chwili skorzystać z krótkiej podpowiedzi lub szerszych wyjaśnień z menu *Pomocy* (*F1*).

W widoku *projektu* planujemy i ustalamy nazwy pól, typy danych oraz ich właściwości. W każdym momencie możemy przełączyć się do utworzonej tabeli w widoku *arkusza danych*, gdzie możemy rozpocząć wypełnianie utworzonej tabeli rekordami. Po dokonaniu zmian w projekcie musi nastąpić ich zapis w tabeli – dopiero teraz można wprowadzać dane czy testować reguły i właściwości. Zwróć uwagę, że paski narzędzi w obu widokach tabeli różnią się, co ilustruje rysunek 2.15. Przyciskiem *Widok* można dowolnie zmieniać oba widoki tabeli.

Jeśli po przeczytaniu informacji na temat typów danych oraz ich właściwości uważasz, że w Twoim projekcie tabeli *Studenci_dane* należy dokonać zmian, na przykład wprowadzić regułę poprawności czy uzupełnić komunikat o błędzie, zrób to teraz.



Rysunek 2.15. Tabela w widoku projektu (po lewej) i w widoku arkusza danych (po prawej)

Wprowadź do tabeli dane pięciu osób zgodnie z ustalonymi wcześniej typami pól. Sprawdź, czy masz możliwość wpisywania danych niezgodnych z założeniami.

Kliknięcie myszką w obrębie pola powoduje umieszczenie w nim punktu wprowadzania danych (kursora tekstowego). Jednocześnie następuje uaktywnienie rekordu, w którym znajduje się to pole. Aktywny rekord jest oznaczony kolorowym cieniowaniem (selektorem) widocznym na lewej krawędzi tabeli. Naciśnięcie jakiegokolwiek klawisza znakowego powoduje pojawienie się symbolu piszącego ołówka, co oznacza, że w aktywnym rekordzie zostały wprowadzone zmiany. Los tych zmian zależy od dalszego postępowania: przejście do innego rekordu powoduje zapisanie wprowadzonych zmian, a naciśnięcie klawisza *Esc* — przywrócenie poprzedniej postaci rekordu i rezygnację z dodania nowego rekordu.

Możesz zmieniać szerokości kolumn (pól) w tabeli, zmieniać ich kolejność, modyfikować projekt i zmieniać dane, pamiętaj jednak, że mogą pojawić się trudności z przeprojektowaniem tabeli z istniejącymi rekordami. Zawarte w tabeli dane mogą nie spełniać zmienionych przez nas kryteriów. Może to czasami oznaczać utratę wpisanych danych. Warto więc takie próby zmian poprzedzić wykonaniem kopii pliku bazy danych.

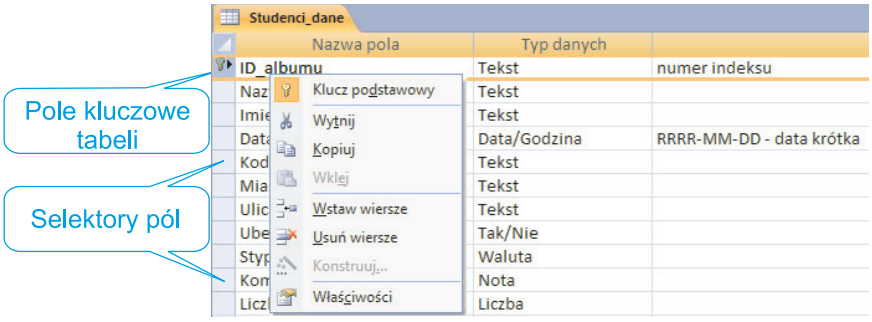
Usunięcie pola wraz z danymi jest nieodwracalne. Warto więc skopiować tabelę przed rozpoczęciem eksperymentów.

2.3.4. Pole kluczowe tabeli

Na rysunku 2.11 w widoku *projektu* obok nazwy pola *ID_albumu* widnieje *kluczyk*. Jest to symbol klucza podstawowego, którym w tabeli *Studenci_dane* jest pole *ID_albumu*.

Aby dodać do pola *klucz podstawowy (klucz własny)*, otwórz tabelę w widoku *projektu*. Zaznacz pole (lub pola), które chcesz zdefiniować jako klucz podstawowy, klikając selektor wiersza dla żądanego pola. Następnie kliknij przycisk *Klucz podstawowy* lub wybierz opcję *Klucz podstawowy* z menu podręcznego. Na selektorze pola pojawi się symbol klucza, jak na rysunku 2.16.

Aby zaznaczyć kilka pól, przytrzymaj klawisz *Ctrl*, a następnie kliknij selektor wiersza dla każdego pola.



Rysunek 2.16. Ustawienie klucza podstawowego tabeli

Definicja

Klucz podstawowy (własny) to jedno pole lub większa liczba pól (atrybutów, kolumn), których wartości jednoznacznie identyfikują każdy rekord w tabeli (wskazują na dany wpis w tabeli).

Pole kluczowe musi mieć unikatowy indeks (nie zezwala na powtórzenie wartości) i nie dopuszcza wartości *Null*. Pole kluczowe służy do wiązania (sprzężenia) tabeli z *kluczami obcymi* w innych tabelach.

Wartość pola kluczowego zawiera unikatową informację, która nie może pojawić się w żadnym innym rekordzie tabeli. Jeśli więc na tym samym kierunku studiów będzie dwóch studentów o tych samych imionach i nazwiskach, będą oni rozróżniani przez różne wartości pola kluczowego *ID_albumu*.

Tabela może zawierać jeszcze inne klucze, tzw. *klucze obce*, będące *kluczami własnymi innych tabel*.

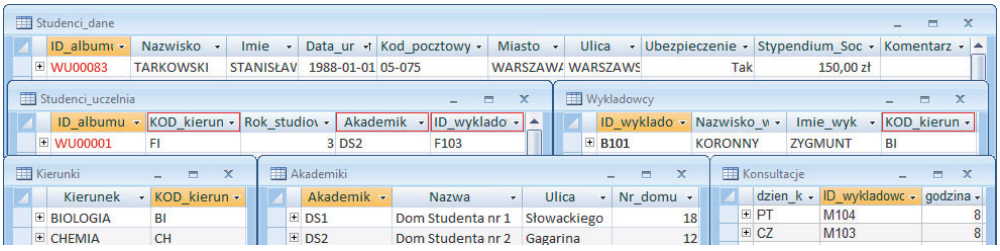
Definicja

Klucze obce:

- są kluczami własnymi innych tabel bazy;
- pozwalają spoić czyli połączyć relacjami dane z różnych tabel;
- wartości klucza obcego mogą się powtarzać.

Na płycie CD dołączonej do podręcznika zamieszczono plik *uczelnia.accdb*, w którym utworzono już podstawowe tabele bazy Uniwersytetu Wirtualnego. Otwórz na chwilę ten plik i zobacz tabele w widokach *arkusza danych* i *projektu*. Na rysunku 2.17 zestawione zostały wszystkie tabele tworzonej bazy. Można z łatwością określić, które pole pełni rolę *klucza własnego*, a które jest *kluczem obcym*.

Pomarańczowym kolorem oznaczono pola będące kluczami własnymi (podstawowymi). W każdej tabeli jest jeden taki klucz, czyli dla wszystkich tabel ustanowiono klucz podstawowy. We wszystkich tabelach pokazanych na rysunku 2.17 klucz podstawowy jest jednym polem (jedną kolumną) tabeli.



Rysunek 2.17. Klucze własne i obce w tabelach bazy Uczelnia

W tabeli *Wykladowcy* oprócz klucza własnego jest klucz obcy: *KOD_kierunku*. To klucz podstawowy tabeli *Kierunki*. Tabela *Studenci_uczelnia* ma aż trzy klucze obce: *KOD_kierunku*, *ID_wykladowcy* i *Akademik*. Na rysunku 2.17 klucze obce zaznaczono czerwoną ramką.

To właśnie dzięki polom kluczowym w relacyjnej bazie danych możemy budować *relacje* pomiędzy tabelami.

2.3.5. Definiowanie relacji

Definicja

Relacjami nazywamy związki (połączenia, powiązania, sprzężenia) między tabelami, definiowane za pomocą pola (lub kilku pól) występującego w obu powiązanych ze sobą tabelach.

Relacje umożliwiają jednoczesne odczytywanie informacji z wielu różnych tabel. Dzięki powiązaniu tabel relacjami możemy zaprojektować bazę danych tak, aby zminimalizować ilość nadmiarowych informacji.

Do połączenia dwu tabel można użyć pola kluczowego (własnego lub obcego) występującego w obu tabelach. Pola te muszą przechowywać ten sam typ danych, a w przypadku pól liczbowych muszą mieć również ten sam rozmiar (typ).

Istnieją trzy typy relacji:

- *Relacja jeden-do-jednego*, w której *jednemu* rekordowi z tabeli *A* odpowiada dokładnie *jeden* rekord z tabeli *B*.
- *Relacja jeden-do-wielu*, w której *jednemu* rekordowi z tabeli *A* odpowiada *wiele* rekordów z tabeli *B*.
- *Relacja wiele-do-wielu*, w której *jednemu* rekordowi z tabeli *A* odpowiada *wiele* rekordów z tabeli *B* oraz *jednemu* rekordowi z tabeli *B* odpowiada *wiele* rekordów z tabeli *A*.

We wszystkich typach relacji pole łączące w tabeli *A* (nadrzędnej) *musi być kluczem głównym* lub *polem indeksowym* jednoznacznie identyfikującym rekordy tej tabeli.

Korzystniej jest definiować relacje dla tabel, które nie zawierają jeszcze danych. W przeciwnym razie przechowywane w nich informacje mogą przeszkodzić i uniemożliwić zdefiniowanie relacji.

Pola, pomiędzy którymi ustanawiane są relacje, nie muszą mieć takich samych nazw. Mogą się one różnić. Przy tworzeniu relacji najważniejsze są typy danych i wartości pól, a nie ich nazwy.

Aby możliwe było ustanowienie relacji w bazie Uniwersytetu Wirtualnego, na płycie CD dołączonej do podręcznika zamieszczono plik *uczelnia_bez_relacji.accdb*, w którym usunięto istniejące już relacje między tabelami. Skopiuj ten plik na lokalny dysk i otwórz go.

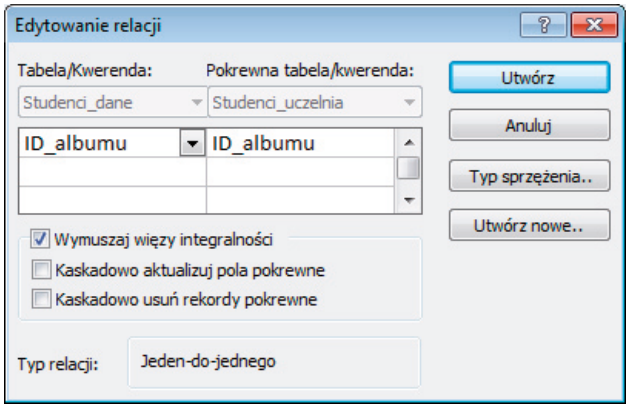
Relacja jeden-do-jednego (1-1)

Relacja *jeden-do-jednego* powstanie, gdy jednemu rekordowi danej tabeli przyporządkowany zostanie jeden rekord drugiej tabeli. Połączone relacją dane mogłyby znajdować się w jednej tabeli, ale zostały rozbite na dwie części ze względu na czytelność i przejrzystość bazy danych, łatwość obsługi mniejszych tabel oraz ewentualną poufność niektórych danych.

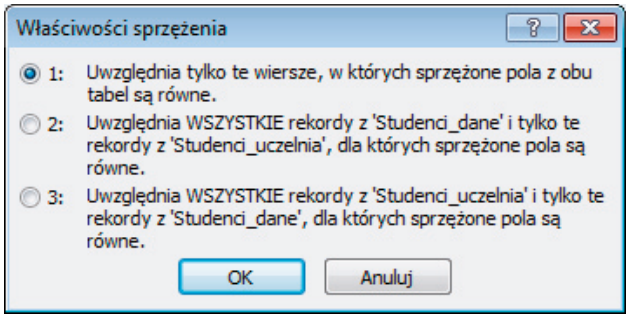
Zauważ, że w dwóch tabelach, *Studenci_dane* oraz *Studenci_uczelnia*, występuje taki sam klucz własny *ID_albumu*. Zsynchronizowanie tych dwóch tabel następuje poprzez to pole kluczowe. Jedna duża tabela została podzielona na dwie mniejsze. Każdy student ma wpisy w obu tabelach. Każdemu studentowi przysługuje dokładnie jeden wpis w każdej tabeli.

Aby ustanowić *relację* pomiędzy tabelą *A*: *Studenci_dane* i tabelą *B*: *Studenci_uczelnia*, należy skorzystać z pól kluczowych obu tych tabel, czyli *ID_albumu*, i wykonać następujące czynności:

1. Przejść do zakładki *Narzędzia bazy danych*.
2. Sprawdzić, czy pole łączące (*ID_albumu)* w tabeli *A* ma właściwość *Indeksowane* ustawioną na *Tak (Bez powtórzeń)*. Ponieważ są to pola kluczowe, mają tę właściwość ustawioną automatycznie.
3. Zamknąć tabele *A* i *B*, jeśli są otwarte.
4. W grupie narzędzi *Pokazywanie/ukrywanie* wybrać *Relacje*.
5. W okienku *Pokazywanie tabeli* wskazać i dodać tabele *A* i *B*, okienko zamknąć.
6. W oknie definiowania relacji połączyć tabele, chwytając pole *ID_albumu* w tabeli *A* i przenosząc je, jak kartę w pasjansie, do pola *ID_albumu* w tabeli *B*. Ważne jest, z której tabeli przenosimy pole — ta tabela staje się tabelą nadrzędną, czyli rodzicem. Podczas definiowania relacji należy określić, która tabela będzie tabelą nadrzędną (zwaną też rodzicem), a która podrzędną, pokrewną (zwaną również dzieckiem). Kierunek przenoszenia pola kluczowego nie jest dowolny i ma bardzo duże znaczenie, ponieważ ustala reguły integralności bazy. Tabela *Studenci_dane* jest *rodzicem*, a tabela *Studenci_uczelnia — dzieckiem*.
7. W oknie *Edytowanie relacji* zaznaczyć pole *Wymuszaj więzy integralności* (pojawi się informacja o rodzaju tworzonej relacji *jeden-do-jednego*, jak na rysunku 2.18).
8. Wybrać typ sprzężenia spośród opcji widocznych na rysunku 2.19 przedstawiającym okno *Właściwości sprzężenia*. Niech sprzężenie uwzględnia te wiersze, w których sprzężone pola z obu tabel są równe.



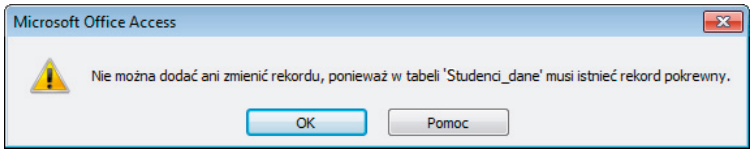
Rysunek 2.18. Ustawianie parametrów relacji w oknie edytowania relacji



Rysunek 2.19. Właściwości sprzężenia jeden-do-jednego

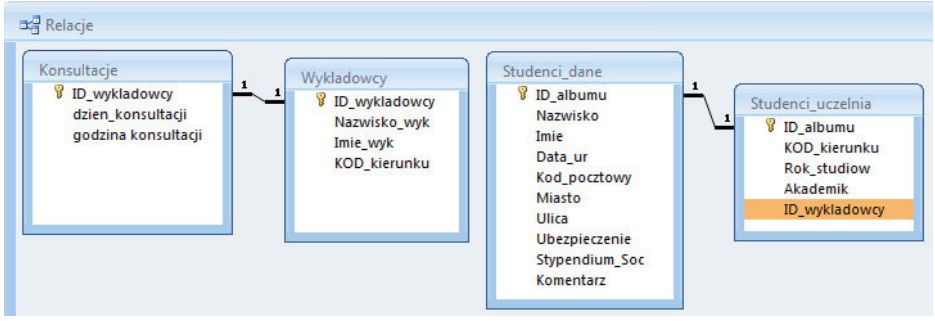
9. Utworzyć relację, klikając przycisk *Utwórz*.

Ustanowienie relacji z takimi właściwościami spowoduje, że możemy wypełnić tabele tylko w określonej kolejności — zacząć od wypełnienia tabeli rodzica (*Studenci_dane*), a dopiero później przejść do tabeli dziecka (*Studenci_uczelnia*). Przy próbie wypełnienia tabel w odwrotnej kolejności nastąpi naruszenie reguł integralności i nie będzie można dodać rekordu do tabeli. Pojawi się komunikat jak na rysunku 2.20. Aby dodać lub zmienić ten rekord, trzeba najpierw dodać do tabeli po stronie „jeden" (nadrzędnej) rekord zawierający tę samą wartość dla odpowiedniego pola.



Rysunek 2.20. Komunikat związany z hierarchią tabel w relacji i typem sprzężenia

Czynności powtarzamy dla tabel *Wykladowcy* i *Konsultacje*, dla których polem sprzęgającym tabele będzie również pole kluczowe obu tabel: *ID_wykladowcy*. W tej relacji tabelą nadrzędną jest tabela *Wykladowcy*, dlatego nie można dodać konsultacji wykładowcy przed jego wcześniejszym dodaniem do tabeli *Wykladowcy*. Efekt widzimy w oknie *Relacje*, gdzie pomiędzy tabelami powstały symboliczne połączenia (więzy) z informacją o rodzaju relacji (1-1), co ilustruje rysunek 2.21.



Rysunek 2.21. Ustanowione relacje jeden-do-jednego (1-1)

Oto znaczenie pozostałych poleceń wyświetlanych w oknie *Edytowanie relacji*, pokazanych na rysunku 2.19:

- *Wymuszaj więzy integralności* — tabele bazy danych Accessa mogą zostać połączone tak, aby przy wprowadzaniu i usuwaniu danych zachowane były *reguły integralności danych*. Na przykład tabele *Studenci_uczelnia* i *Wykladowcy* powinny być połączone w taki sposób, aby nie można było dodać *ID_wykladowcy* jako opiekuna studenta, jeśli takiego wykładowcy nie ma w tabeli *Wykladowcy*, zawierającej dane o wykładowcach, co ustrzeże naszą bazę przed wprowadzeniem błędnych danych.
- *Kaskadowo aktualizuj pola pokrewne* — wprowadzenie zmiany w polu wiążącym tabeli nadrzędnej spowoduje zmianę wartości w tym polu w rekordach tabeli podrzędnej odpowiadających zmienionemu rekordowi.
- *Kaskadowo usuń rekordy pokrewne* — usunięcie rekordu *X* z tabeli nadrzędnej spowoduje usunięcie rekordów z tabeli podrzędnej odpowiadających rekordowi *X*.

Jeśli na przykład zaznaczymy polecenia *kaskadowego uaktualniania* i *kaskadowego usuwania* w relacji wiążącej tabele *Akademiki* i *Studenci_uczelnia*, a następnie zmienimy kod akademika z *DS3* na *DS5* w tabeli *Akademiki*, to taka sama zmiana nastąpi we wszystkich rekordach w tabeli *Studenci_uczelnia*, gdzie wpisano w polu wiążącym *Akademik* wartość *DS3*. Jeśli natomiast usuniemy z tabeli *Wykladowcy* wykładowcę o kodzie *C103*, z tabeli *Konsultacje* usunięte zostaną wszystkie wpisy dotyczące tego rekordu (wykładowcy).

Aby zmodyfikować relację, należy w oknie *Relacje* dwukrotnie kliknąć linię łączącą obie tabele lub kliknąć linię prawym przyciskiem myszy i wybrać opcję *Edytuj relację…*. Na wstążce pojawi się zakładka z narzędziami związanymi z projektowaniem relacji między tabelami. Tam, w grupie *Narzędzia*, zamieszczono przycisk *Edytuj relacje*, który również otworzy okno *Edytowanie relacji*.

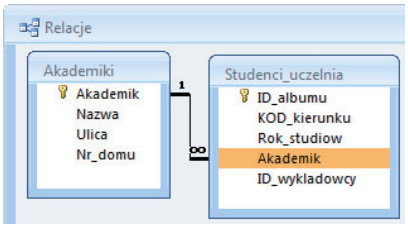
Aby usunąć relację, należy w oknie *Relacje* zaznaczyć linię łączącą tabele, a następnie usunąć ją klawiszem *Delete* (albo kliknąć tę linię prawym przyciskiem myszy i wybrać opcję *Usuń*).

Relacja jeden-do-wielu (1-∞)

Relacja *jeden-do-wielu* jest najczęściej stosowanym typem relacji, w której jednemu rekordowi z tabeli *A* odpowiada wiele rekordów z tabeli *B*, a każdemu rekordowi tabeli *B* jest przyporządkowany co najwyżej jeden rekord tabeli *A*.

W naszym przykładzie można ustanowić następujące relacje *jeden-do-wielu*:

- każdemu akademikowi możemy przyporządkować ***wielu*** studentów, natomiast każdy student zamieszkuje ***najwyżej*** w jednym akademiku, co ilustruje rysunek 2.22;
- każdemu wykładowcy możemy przyporządkować ***wielu*** studentów, których jest opiekunem, natomiast student ma ***jednego*** opiekuna;
- każdemu kierunkowi możemy przyporządkować ***wielu*** studentów, a każdy student studiuje na ***jednym*** kierunku;
- każdemu kierunkowi możemy przyporządkować ***wielu*** wykładowców, a każdy wykładowca uczy na ***jednym*** kierunku.



Rysunek 2.22. Ustanowiona relacja jeden-do-wielu (1-∞)

Wypełniając danymi tabele połączone tego typu relacją, powinniśmy pamiętać, że próba wykonania operacji, która mogłaby naruszyć reguły integralności dla powiązanych ze sobą tabel, skończy się niepowodzeniem. Sytuacja ta wystąpi na przykład wówczas, gdy jest dodawany lub zmieniany rekord z tabeli po stronie „wielu", a powiązanego z nim rekordu brak w tabeli po stronie „jeden". W przypadku relacji zilustrowanej na rysunku 2.22 oznacza to, że aby student mógł zamieszkać w nowym akademiku, trzeba najpierw dodać do tabeli po stronie „jeden" (tabela *Akademiki*) rekord zawierający dane nowego akademika, a dopiero później będzie można używać tego pola w tabeli *Studenci_uczelnia*.

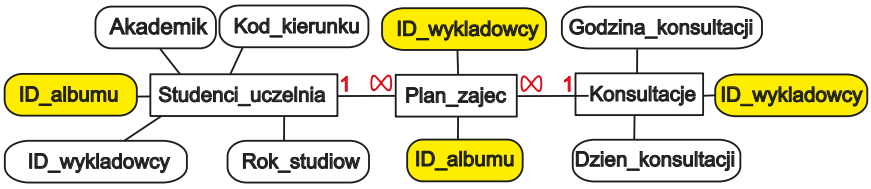
Relacja wiele-do-wielu (∞-∞)

W relacji *wiele-do-wielu jednemu* rekordowi z tabeli *A* odpowiada *wiele* rekordów z tabeli *B* oraz *jednemu* rekordowi z tabeli *B* odpowiada *wiele* rekordów z tabeli *A*.

Wyjaśnienie relacji *wiele-do-wielu* jest trochę bardziej skomplikowane. Rozważmy na przykład sytuację, w której każdy student może zapisać się na konsultacje do wielu wykładowców. Jakie zajdą relacje między studentami a wykładowcami? Każdy student ma zajęcia z wieloma wykładowcami, każdy wykładowca ma zajęcia z wieloma studentami.

Na pierwszy rzut oka moglibyśmy potraktować tę relację jak dwie różne relacje *jeden-do-wielu*. Jednak rozwiązanie to nie byłoby dobre, ponieważ wymagałoby dodania kluczy obcych do obu tabel (*Konsultacje* oraz *Studenci_uczelnia*), co w efekcie doprowadziłoby do wystąpienia powtarzających się wierszy tabeli z wieloma polami danych. Jest więc oczywiste, że w wyniku tego rozwiązania baza danych przechowywałaby mnóstwo niepotrzebnych danych. Dodatkowo należy zwrócić uwagę na fakt, że *ID_wykladowcy*, jako opiekuna, już się znajduje w tabeli *Studenci_uczelnia.*

Realizując relację *wiele-do-wielu*, należy do bazy danych **dodać nową, dodatkową tabelę**, tak aby podzielić relację na dwie relacje *jeden-do-wielu*. W naszym przypadku dodamy tabelę *Plan_zajec*, w której zaprojektujemy pola będące kluczami obcymi, czyli kluczami głównymi tabel *Konsultacje* i *Studenci_uczelnia*: *ID_wykladowcy* i *ID_albumu*. Powstała tabela *Plan_zajec* ma tylko te dwa pola. Każdy student będzie mógł dodać swoje *ID_albumu* i *ID_wykladowcy*, w którego zajęciach chce uczestniczyć. Diagram na rysunku 2.23 przedstawia opisaną procedurę w sposób graficzny.



Rysunek 2.23. Diagram relacji wiele-do-wielu w tabeli Plan_zajec

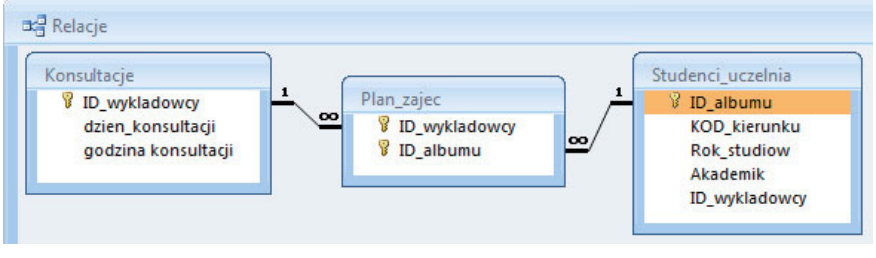
W tabeli *Plan_zajec* wystąpią powtórzone dane zarówno studentów, którzy wybierają po kilka zajęć, jak i wykładowców, których zajęcia wybierają studenci. Jednak nie powinien nigdy powtórzyć się zestaw *student-wykładowca*. Żeby to osiągnąć, można posłużyć się kluczem podstawowym, składającym się z dwóch pól (atrybutów). W oknie projektowym tabeli zaznacz selektory obu pól i ustaw dla nich klucz podstawowy, jak na rysunku 2.24. Teraz zdublowanie w tabeli zestawu ID studenta i wykładowcy nie będzie możliwe — pojawi się komunikat o konieczności zmiany powtarzających się danych.

Plan_zajec

Nazwa pola	Typ danych
ID_wykladowcy	Tekst
ID_albumu	Tekst

Rysunek 2.24. Projekt dodatkowej tabeli łączącej w relacji wiele-do-wielu

Po ustaleniu relacji między trzema tabelami: *Konsultacje*, *Plan_zajec* i *Studenci_uczelnia*, w oknie *Relacje* pojawi się obraz symbolizujący utworzoną relację, jak na rysunku 2.25. Jest to relacja *wiele-do-wielu*.

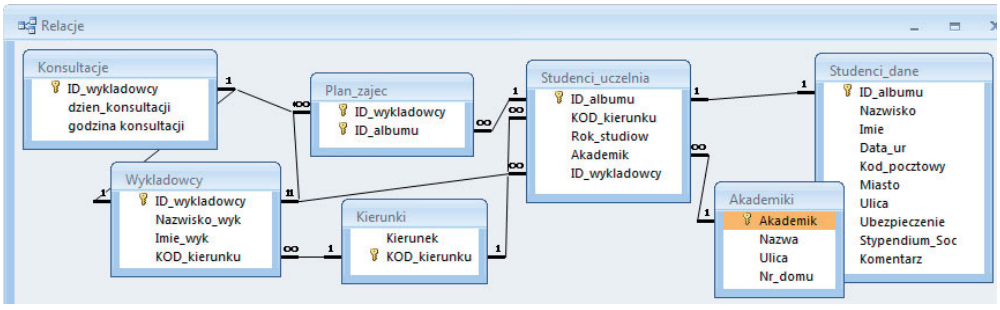


Rysunek 2.25. Ustanowiona relacja wiele-do-wielu (∞-∞)

W naszej bazie, zawierającej obecnie siedem tabel, po wyświetleniu w oknie *Relacje* wszystkich tabel zobaczymy powstałe relacje, jak na rysunku 2.26. Przyjrzyj się układowi zależności i połączeń. Rozpoznaj poszczególne typy relacji ustanowionych między tabelami.

Teraz, po połączeniu tabel relacjami, będziemy mogli odwoływać się do nich, jakby znajdowały się w jednym zbiorze spójnych danych, a nie w osobnych tabelkach.

Jeżeli wszystkie relacje zostały wykonane poprawnie, zmień nazwę swojej bazy danych z *uczelnia_bez_relacji.accdb* na *uczelnia.accdb*. Jeżeli nie masz takiego układu relacji jak na powyższej ilustracji, do dalszych ćwiczeń możesz użyć pliku *uczelnia.accdb* zamieszczonego na płycie CD dołączonej do podręcznika. Skopiuj plik na lokalny dysk.

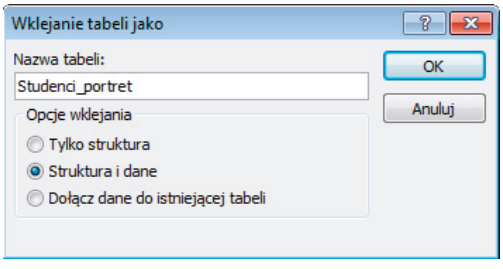


Rysunek 2.26. Relacje łączące tabele bazy Uczelnia

2.3.6. Operacje na tabelach bazy danych

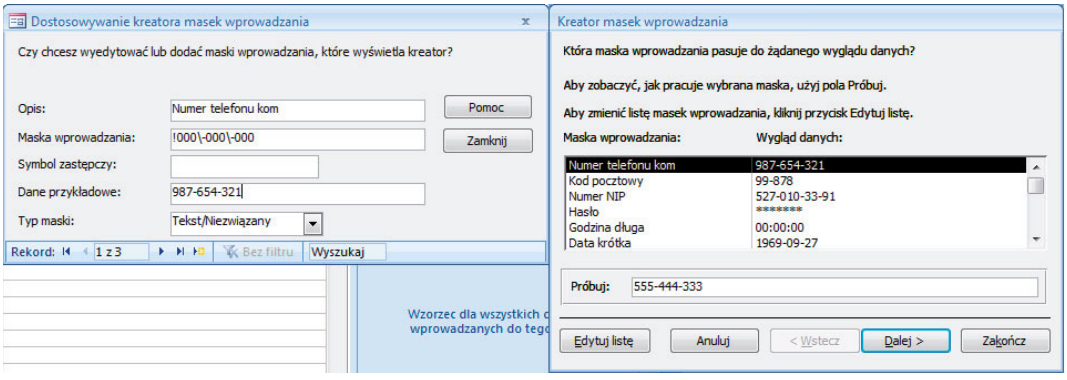
Operacje edycyjne

Na obiektach bazy danych można wykonywać operacje edycyjne — kopiować tabele, usuwać je, zmieniać ich nazwy. Wszystkie te operacje należy wykonywać na zamkniętych obiektach (w oknie bazy danych), korzystając z poleceń w menu kontekstowym, otwieranym prawym przyciskiem myszy. Zamknij więc wszystkie tabele i wykonaj kilka czynności:



Rysunek 2.27. Opcje wklejania kopii tabeli

- Wykonaj kopię tabeli *Studenci_dane* i nazwij ją *Studenci_portret*. Zauważ, że masz trzy różne opcje wykonania kopii tabeli: samej struktury tabeli (bez zawartych w niej danych), struktury wraz z danymi lub dołączenia danych do istniejącej tabeli. Wybór opcji wklejania kopii tabeli został zilustrowany na rysunku 2.27.
- Zmień nazwę utworzonej tabeli na *Studenci_portret_tel*. Wykorzystaj menu kontekstowe lub klawisz funkcyjny *F2*.
- Otwórz tabelę *Studenci_portret_tel* w widoku *projektu*.
- Usuń wszystkie pola oprócz pola kluczowego *ID_albumu*.
- Dodaj pole *Telefon* dla danych typu *Tekst*. Ustal rozmiar pola na 9 znaków. Z kreatora masek wprowadzania wybierz *Numer telefonu*. Ustal, że dane mają być przechowywane bez symbolu w masce. Maskę wprowadzania zmodyfikuj tak, aby pasowała do 9-cyfrowego numeru telefonu komórkowego. W tym celu poleceniem *Edytuj listę* otwórz okno *Dostosowywanie kreatora masek wprowadzania*. Ustaw powyższe parametry maski tak, jak to ilustruje rysunek 2.28. Po zamknięciu okna zmiany w ustawieniach kreatora masek zostaną zapisane. Zapisz tabelę i przejdź do widoku *arkusza danych*, a następnie wprowadź w nim kilka numerów telefonów.
- Powróć do widoku *projektu* i zmień nazwę pola *Telefon* na *Telefon_kom*.
- W tabeli *Studenci_uczelnia* w widoku *projektu* zmień nazwę pola *ID_wykladowcy* na *ID_opiekuna*.



Rysunek 2.28. Dostosowywanie kreatora masek wprowadzania

Wstawianie obiektów OLE

Technika OLE umożliwia *osadzanie* w polach tabeli *i dołączanie* do nich obiektów powstałych również w aplikacjach innych niż Access. Mogą to być na przykład dokumenty Worda lub arkusze Excela.

Osadzenie obiektu w polu tabeli polega na tym, że dany obiekt wraz z informacją o aplikacji, w jakiej został utworzony, umieszczany jest w polu tabeli.

Połączenie pola tabeli z obiektem polega zaś na tym, że w polu tabeli umieszczana jest, oprócz nazwy aplikacji tworzącej obiekt, ścieżka dostępu do obiektu zachowanego w postaci pliku.

Aby możliwe było dodanie obiektu OLE, w tabeli musi znajdować się kolumna typu *Obiekt OLE*. Dlatego dodamy teraz do tabeli *Studenci_portret_tel* w widoku *projektu* pole *Portret*, które będzie przechowywało dane typu *Obiekt OLE*. W widoku *arkusza danych* z menu podręcznego w polu tabeli należy wybrać *Wstaw obiekt…*. W oknie dialogowym można wybrać jeden z wymienionych obiektów i na bieżąco utworzyć obiekt OLE w skojarzonej z nim aplikacji. Arkusz, rysunek bądź równanie zostaną osadzone w polu tabeli. Można też wybrać polecenie *Utwórz z pliku*, a następnie wskazać plik, w którym znajduje się żądany obiekt (na przykład arkusz Excela). Po wskazaniu obiektu przyciskiem *OK* można potwierdzić jego wstawienie (*osadzenie*) w polu tabeli Accessa lub zaznaczyć pole *Połącz* i tym samym *połączyć* pole tabeli z obiektem zachowanym na dysku, w pliku pod wskazaną nazwą. Usunięcie lub przeniesienie powiązanego obiektu spowoduje nieprawidłową pracę bazy danych.

Na czym polega różnica między osadzeniem a połączeniem obiektu z polem tabeli? Pokażemy to na przykładzie. Przygotuj dwa pliki z rysunkami o wymiarach 100×100 pikseli, z których jeden przedstawia koło, a drugi kwadrat. Rysunków nie koloruj. Zapisz je w folderze *OLE* jako *kolo.bmp* i *kwadrat.bmp*. Plik z kołem osadź w polu *Portret* pierwszego studenta w tabeli *Studenci_portret_tel*, natomiast plik z kwadratem połącz z drugim studentem. Zapisz zmiany w tabeli. Teraz w obu plikach z rysunkami pokoloruj figury. Zapisz zmiany. Przejdź do tabeli i zobacz wstawione obiekty OLE. W pliku osadzonym koło nie jest pokolorowane, a w pliku połączonym kwadrat jest kolorowy.

Uzupełnij obrazki w trybie bezpośredniej edycji pola *Portret*. To trochę pracochłonne zajęcie. Można też wykorzystać kopiowany obrazek kolorowego okręgu i zmienić jego kolory. Miłej zabawy. Zauważ, że po wprowadzeniu obrazu nie jest on widoczny. Widać jedynie informację, że znajduje się tam obiekt typu bitmapa.

2.3.7. Zmiana sposobu prezentowania danych

Wygląd tabeli w widoku arkusza danych

Wygląd okna tabeli ma wpływ na sposób jej drukowania. W programie MS Access drukowanie tabeli jest zgodne ze sposobem jej wyświetlania w widoku *arkusza danych*.

Po otwarciu tabeli w widoku *arkusza danych* na karcie *Narzędzia główne*, w grupach narzędzi *Czcionka*, znajdują się narzędzia służące do zmiany sposobu prezentowania danych (ustawienia: czcionki, koloru wypełnienia automatycznego i alternatywnego, linii siatki). W grupie *Rekord* zamieszczono narzędzia do zmiany sposobu prezentowania kolumn i wierszy (pól i rekordów) — do modyfikacji wysokości wiersza, szerokości kolumny, ukrywania kolumn, blokowania kolumn oraz sprawdzania pisowni (*F7*).

Można również zmienić kolejność wyświetlania poszczególnych pól tabeli. W widoku *arkusza danych*, chwytając za nazwę pola (etykietę), przesuwamy pole w inne miejsce w tabeli. Zmiana ta nie ma wpływu na układ pól w widoku *projektu*. Natomiast jeśli — chwytając za selektor pola — zamienimy miejscami pola w widoku *projektu*, po zapisaniu zmian w tabeli nastąpi zamiana tych pól w widoku *arkusza danych*.

Zmiana kolejności wyświetlania pól tabeli, podobnie jak wszystkie inne operacje zmiany sposobu prezentowania danych, nie ma żadnego wpływu na strukturę tabeli.

Szybkie sortowanie

Dane wyświetlane w tabeli są domyślnie sortowane rosnąco według pól kluczowych (klucza własnego). Można jednak na czas pracy z tabelą zmienić kolejność sortowania. Należy wówczas:

1. Ustawić kursor w polu, według którego dane mają być posortowane (albo wcześniej zaznaczyć jedną kolumnę lub kilka kolumn tabeli).
2. W grupie *Sortowanie i filtrowanie*, na karcie *Narzędzia główne*, określić sposób porządkowania: *Sortuj rosnąco* lub *Sortuj malejąco*. Istnieje również możliwość usunięcia sortowania przyciskiem *Wyczyść wszystkie sortowania*.

Jeżeli podczas zamykania tabeli zapiszemy zmiany w jej układzie, to po ponownym otwarciu tabeli dane będą prezentowane zgodnie z ustawieniami i posortowane tak, jak przed jej zamknięciem.

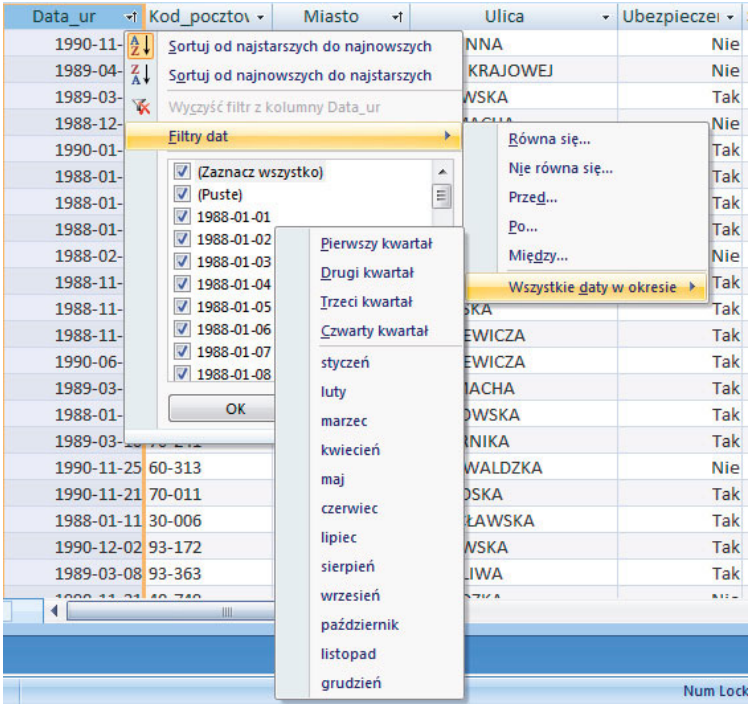
Filtrowanie danych

Filtry to narzędzia służące do wyszukiwania informacji. Uściślając tę definicję, można powiedzieć, że filtry służą do ograniczania informacji wyświetlanych w widoku *arkusza danych* tabeli do tych, które spełniają określone warunki (kryteria).

Przy użyciu filtru można znaleźć jeden rekord lub więcej określonych rekordów. Filtr ogranicza widok danych do określonych rekordów bez konieczności zmieniania projektu tabeli, kwerendy, formularza czy też raportu. Na przykład zamiast przeglądać kilkusetstronicową listę artykułów w poszukiwaniu produktów mieszczących się w przedziale cenowym od 10 do 50 zł, można zastosować filtr danych i wyświetlić tylko te rekordy tabeli *Produkty*, dla których wartość pola *Cena* mieści się w zakresie od 10 do 50 zł. Gdy zostanie zastosowany filtr, w widoku wyświetlą się tylko te rekordy, które zawierają wartości wybrane przez użytkownika. Pozostałe rekordy będą ukrywane do czasu usunięcia filtru.

W programie Microsoft Office Access 2007 filtry są wbudowane w każdy widok, w którym wyświetlają się dane. **Szybkie filtrowanie** danych w tabelach następuje w widoku *arkusza danych*. Dostępność filtrów zależy od typu danych w zaznaczonej kolumnie. Zatem inne filtry będą dostępne dla danych liczbowych, inne dla dat, a jeszcze inne dla tekstów.

W nagłówku kolumn tabeli zamieszczono filtry, które są dostępne po wskazaniu myszą strzałki znajdującej się obok nazwy pola. Ten sam efekt otrzymamy po wybraniu przycisku Filtruj znajdującego się w grupie *Sortowanie i filtrowanie* na karcie *Narzędzia główne*. Przykładowe filtry dla danych typu *Data/Godzina* ilustruje rysunek 2.29.



Rysunek 2.29. Opcje szybkiego filtrowania danych typu Data/Godzina w tabeli

Po kliknięciu przycisku Przełącz filtr (zastosuj filtr) wyświetlone zostaną tylko te rekordy, które w danym polu są równe zaznaczonej wartości. Można również wybrać przycisk Zaznaczenie, aby wyświetlone zostały tylko te rekordy, które spełniają jedno z zaproponowanych kryteriów wyboru odnoszących się do zaznaczonej wartości. Na szczególną

uwagę zasługuje tu możliwość wybrania filtrowania z tzw. wyłączeniem wyboru *Nie równa się „”*, co pozwala na wybranie wszystkich rekordów *oprócz* spełniających kryterium wyboru, na przykład: pokaż wszystkich oprócz zamieszkałych w Warszawie, pokaż wszystkich oprócz studiujących na kierunku biologia, pokaż wszystkich oprócz tych, którzy mają 2 z matematyki itp.

Aby zrezygnować z zastosowanego filtru, należy użyć przycisku Przełącz filtr, którego znaczenie zmienia się w zależności od tego, czy filtr w danej chwili jest zastosowany, czy też nie. Na pasku stanu tabeli zamieszczane są również aktualne informacje o zastosowanych filtrach. Tam też jest możliwość ich usunięcia.

Ćwiczenie 2.4. Do bazy danych należy dodać dwóch nowych wykładowców, którzy jeszcze nie określili terminu swoich konsultacji. Uzupełnij te brakujące dane, dodając w tabelach *Wykladowcy* i *Konsultacje* odpowiednie wpisy.

ID_wykladowcy	Nazwisko_wyk	Imie_wyk	KOD_kierunku
I104	ZAWIŁY	JANUSZ	IN
I105	CICHY	JAN	IN

Wykładowca *I104* ma mieć konsultacje we wtorek, natomiast *I105* — w piątek. Jednak nie może być to godzina konsultacji wybrana wcześniej przez któregokolwiek z wykładowców, którzy już ustalili swoje terminy zajęć.

Aby uniknąć przeglądania i notowania zajętych już terminów, posłużymy się filtrem. W widoku *arkusza danych* tabeli *Konsultacje* ustaw kursor w polu zawierającym dzień tygodnia *WT*. Po wydaniu polecenia *Zastosuj filtr* Przełącz filtr zostaną wyświetlone ID tylko tych wykładowców, którzy mają konsultacje we wtorek. Teraz z łatwością znajdziemy wolną godzinę konsultacji dla naszego *I104*. Po dopisaniu danych dla wykładowcy *I104* przywróć wyświetlanie wszystkich rekordów tabeli (usuń filtr Przełącz filtr). Wykonaj analogiczny wpis dotyczący konsultacji dla *I105*. Zapisz zmiany w tabeli i zamknij ją.

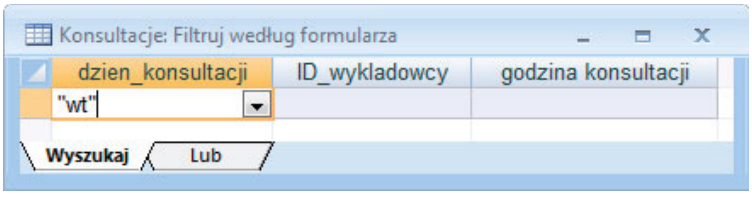
Szybkie filtrowanie można stosować na już przefiltrowanych danych. Na przykład w tabeli *Studenci_uczelnia* chcemy szybko wyszukać wszystkich mieszkańców akademika *DS3* studiujących na trzecim roku. Pierwszy wybór będzie dotyczył *DS3*, a następnie na wyszukanych danych należy ponownie zastosować filtr wyboru — tym razem dla trzeciego roku studiów. Zostanie wyświetlona lista ośmiorga studentów spełniających obydwa kryteria wyboru.

Podczas filtrowania wielu pól w jednym widoku filtry są łączone przy użyciu operatora *i* w następujący sposób:

Miasto = `Szczecin` ***i*** *miesiąc Data_ur* = `maj` *//wyszukaj wszystkich mieszkańców Szczecina //urodzonych w maju.*

Z odfiltrowanymi danymi można pracować tak jak z widokiem wszystkich niefiltrowanych danych — można je edytować, usuwać, dodawać i przechodzić do innych rekordów.

Jeśli nie wiadomo, jakie pozycje znajdują się w określonym polu, warto skorzystać z filtrowania według formularza będącego jedną z zaawansowanych opcji filtrowania danych. Pojawi się lista widoczna na rysunku 2.30, z której można wybrać potrzebne pole wyboru lub odczytać informację, jakiej godziny nie wybrał żaden wykładowca z naszej bazy. Po ustaleniu kryteriów należy przyciskiem Przełącz filtr zastosować filtrowanie.



Rysunek 2.30. Filtr według formularza

W przypadku filtrowania według formularza istnieje możliwość łączenia wyborów spójnikiem *lub*. Aby oprócz wtorku wybrać jeszcze inny dzień konsultacji, na przykład piątek, należy alternatywne kryterium podać na osobnej karcie.

Korzystając z opcji filtrów zaawansowanych, mamy również możliwość wybrania polecenia *Filtr/sortowanie zaawansowane…*, które otwiera okno projektu zaawansowanego wyszukiwania danych.

W siatce projektowej można uwzględnić wiele kryteriów wyszukiwania danych. Należy dla nich ustalić następujące opcje:

- *Pole* — nazwa pola, na które zostanie nałożone kryterium. Pole dodajemy, wybierając je z listy lub dwukrotnie klikając jego nazwę w oknie tabeli nad siatką projektową. Można również przeciągnąć nazwę pola z tabeli do siatki projektowej.
- *Sortuj* — sposób porządkowania wyświetlanych rekordów. Jeżeli wybrane rekordy mają być posortowane według określonego pola (lub określonych pól), należy wybrać sposób sortowania. Kolejność sortowania jest zgodna z kolejnością dodawania pól i kryteriów filtrowania.
- *Kryteria* — warunek (warunki), jaki musi spełnić każdy rekord, aby został wyświetlony na ekranie.

Wskazówka

Jeżeli często używamy zaawansowanego filtrowania, to — dla przyspieszenia pracy — możemy dołożyć do paska narzędzi *Szybki dostęp* przycisk *Filtr/sortowanie zaawansowane…*. W tym celu należy skorzystać z opcji *Więcej poleceń…* na liście opcji *Dostosuj pasek narzędzi Szybki dostęp…*. W polu *Wybierz polecenia z* wybierz *Karta Narzędzia główne*, a następnie na liście wskaż *Filtr/sortowanie zaawansowane…*. Przycisk *Dodaj>>* doda nowy przycisk do paska *Szybki dostęp*. Od tej chwili będzie on widoczny dla wszystkich dokumentów lub tylko dla tego jednego dokumentu, co zależy od ustawienia wyboru w oknie dostosowywania.

Zdefiniowane w ten sposób filtry można również zapisywać poleceniem *Zapisz jako kwerendę*, a następnie przywoływać na ekran poleceniem *Załaduj z kwerendy* (obydwa

polecenia są dostępne w menu podręcznym w czasie, gdy otwarte jest okno definiowania filtru). Zapisaną kwerendę można również otwierać, wyświetlając *Kwerendy* w oknie nawigacji.

Ćwiczenie 2.5. W tabeli *Studenci_dane* wyszukaj studentów, którzy mieszkają w Olsztynie i urodzili się przed 12 marca 1989 roku. Utworzony filtr zapisz jako kwerendę pod nazwą *Studenci_z_Olsztyna <1989-03-12*.

1. W bazie *uczelnia.accdb* otwórz tabelę *Studenci_dane* w widoku *arkusza danych*.
2. Przejdź do pola *Miasto* i odszukaj wartość *Olsztyn*. Aby wyszukać wszystkich studentów z Olsztyna, wybierz polecenie *Równa się „OLSZTYN”* z opcji dostępnych w filtrowaniu Zaznaczenie. Natychmiast otrzymasz listę wszystkich studentów spełniających wskazane kryterium, czyli studentów z Olsztyna. Teraz spośród wyświetlonych rekordów możesz wybierać tych studentów z Olsztyna, którzy spełniają kolejne kryteria wyboru. Niestety, nie możesz tworzyć złożonych kryteriów do danego pola. To jest możliwe tylko z użyciem narzędzia *Filtr/sortowanie zaawansowane…*, które jest już dostępne na pasku *Szybki dostęp*.
3. Przywróć wyświetlanie wszystkich rekordów tabeli poleceniem *Przełącz filtr* Przełącz filtr.
4. Korzystając z polecenia *Filtr/sortowanie zaawansowane…*, w pustej siatce projektowej ustal nowe warunki filtrowania danych. Pierwszym kryterium wyboru ma być fakt zamieszkania w Olsztynie, drugim data urodzenia. Dodaj więc pola *Miasto* oraz *Data_ur* do siatki projektowej. Możesz to wykonać na jeden z trzech sposobów: dwukrotnie klikając nazwy tych pól lub przeciągając te nazwy do siatki projektowej albo wybierając je z listy nazw pól w siatce projektowej. Teraz dodaj kryteria wyboru. W polu *Miasto* wpisz kryterium `OLSZTYN`, zaś w polu *Data_ur* wpisz `<1989-03-12`. Po opuszczeniu pola z kryteriami wyboru automatycznie zostaną dopisane znaki `##` i `""`, które otaczają wpisywaną datę oraz łańcuch znaków. W efekcie kryteria wyboru danych typu *Tekst* i *Data/Godzina* są następujące: `"OLSZTYN"` i `<#1989-03-12#`.
5. Zapisz utworzony filtr jako **kwerendę** pod nazwą *Studenci_z_Olsztyna <1989-03-12* (z menu podręcznego wybierz *Zapisz filtr/sortowanie*).
6. Zastosuj utworzony filtr, wybierając przycisk Przełącz filtr, lub z menu kontekstowego wybierz polecenie *Zastosuj filtr/sortowanie*. Zobacz wyświetlone w zestawieniu dane.
7. Przywróć wyświetlanie wszystkich danych.

W oknie nawigacji przełącz widok na *Wszystkie obiekty programu Access* i zauważ, że w grupie obiektów *Kwerendy* znajduje się utworzona przed chwilą kwerenda wybierająca studentów z Olsztyna urodzonych przed 12 marca 1989 roku.

Przy użyciu zaawansowanych filtrów możliwe jest wyszukiwanie danych z jednej tabeli. Można również stosować kryteria złożone, wykorzystując koniunkcję i alternatywę kry-

teriów (`And`, `Or`), oraz wyszukiwać dane według wielu kryteriów wyboru. Wyszukaj na przykład wszystkich studentów, którzy:

- mieszkają w Olsztynie lub w Warszawie;
- urodzili się pomiędzy 1 stycznia 1989 a 12 grudnia 1989 roku;
- mają nazwisko rozpoczynające się na litery K, L lub R;
- nie noszą nazwiska Kowalski;
- są ubezpieczeni i otrzymują stypendium socjalne.

2.3.8. Import, eksport, załączanie tabeli

Microsoft Office Access 2007 udostępnia narzędzia, dzięki którym możliwe jest korzystanie z danych przechowywanych nie tylko w innych bazach Accessa, ale również w plikach utworzonych przez inne aplikacje. Narzędzia te pozwalają również innym aplikacjom korzystać z danych Accessa.

Import

Do bazy danych Accessa można więc zaimportować dane przechowywane nie tylko w tabelach innych baz danych Accessa, ale i w innych systemach zarządzania bazami danych, arkuszach kalkulacyjnych czy plikach tekstowych. W tym przypadku Access tworzy kopię tych danych i umieszcza ją w tabeli, którą definiuje na podstawie analizy importowanych wartości. Na przykład dla danych liczbowych utworzy kolumnę (pole) typu *Liczba, a* dla danych tekstowych — kolumnę typu *Tekst*.

Aby zaimportować tabelę z danymi z innej bazy danych Accessa, należy w aktywnym oknie bazy danych z grupy *Importowanie* na zakładce *Dane zewnętrzne* wybrać polecenie *Access*. Teraz wystarczy określić źródło danych, czyli wskazać plik, z którego chcemy zaimportować tabelę, i określić sposób i miejsce przechowywania danych po zaimportowaniu. Są dwie możliwości importowania obiektów z innej bazy danych. Jedna polega na utworzeniu kopii obiektu w bieżącej bazie. Wykonywane operacje na zaimportowanych danych nie zostaną odzwierciedlone w źródłowej bazie danych. Druga możliwość łączy importowane obiekty ze źródłową bazą, tworząc tabelę połączoną. Zmiany wprowadzone w danych zaimportowanych zostaną uwzględnione w źródle, i odwrotnie. Po wybraniu pierwszej opcji importu należy w oknie *Importowanie obiektów* wybrać dowolny typ importowanych obiektów i kolejno wskazywać poszczególne obiekty. Tak samo bowiem importujemy zarówno tabelę, jak i pozostałe obiekty bazy danych MS Access.

Ćwiczenie 2.6. Z bazy danych *uczelnia_oceny.accdb*, znajdującej się na płycie CD dołączonej do podręcznika, zaimportuj tabelę *Studenci_oceny*, w której znajdują się oceny kilkudziesięciu studentów Uniwersytetu Wirtualnego. Tabelę tę zaimportuj do wcześniej skopiowanej z płyty CD bazy *uczelnia.accdb*. Po zaimportowaniu tabeli zmień jej nazwę na *Studenci_oceny_z_bazy*.

Ćwiczenie 2.7. Zaimportuj do bazy *uczelnia.accdb* te same dane na temat ocen, ale zamieszczone w tabeli arkusza kalkulacyjnego. Podobnie jak w ćwiczeniu 2.6, w aktywnym oknie bazy danych z grupy *Importowanie* na zakładce *Dane zewnętrzne* wybierz polece-

nie *Excel*. Teraz należy wskazać nazwę pliku, z którego zostanie zaimportowana tabela, czyli znajdujący się na płycie CD plik *studenci_oceny_z_arkusza.xlsx*, oraz określić sposób i miejsce przechowywania danych w bazie *uczelnia.accdb*. Zauważ, że pojawiła się dodatkowa opcja pozwalająca na dołączenie kopii rekordów z pobieranej tabeli do tabeli w bazie. Ty jednak chcesz importować tabelę. Po zatwierdzeniu wyboru zostanie otwarte okno *Kreator importu arkuszy*, w którym należy wybrać arkusz oraz zwrócić uwagę na to, czy dane pobierane z arkusza mają wiersz nagłówka w pierwszym wierszu danych. Jeśli tak, w drugim kroku kreatora zaznacz odpowiednie pole i przejdź do kroku trzeciego. Kreator rozpoznał prawidłowo wszystkie typy pól i je zaimportował. Istnieje możliwość rezygnacji z importu niektórych pól — wystarczy wskazać pole i zaznaczyć opcję *Nie importuj pola (pomiń)*. Wiesz już, jakie znaczenie ma indeksowanie. Sprawdź zatem, czy poprawnie ustalono indeksowanie dla pola *ID_albumu — Tak (Bez duplikatów)*. W ostatnim kroku musisz ustalić, czy tabela ma mieć klucz własny, czy też nie, i czy ten klucz ma być nowym, dodanym polem, czy może chcesz wybrać któreś z pól istniejących w tabeli. W naszym przypadku polem kluczowym będzie pole *ID_albumu*, które należy wybrać po zaznaczeniu opcji *wybieram własny klucz podstawowy*. Pozostało jeszcze podanie nazwy nowej tabeli *Studenci_oceny_z_arkusza*.

Aby uniknąć błędów importu danych z arkusza kalkulacyjnego, podczas wprowadzania informacji do arkusza pamiętaj, że:

- kolumny (pola) danych powinny mieć unikatowe nazwy nagłówków (nazwy pól);
- dane zawarte w kolumnie (polu) muszą być jednego typu;
- żaden wiersz wewnątrz tabeli nie może być pusty;
- żadne komórki nie mogą być scalone;
- puste arkusze należy usunąć.

Wskazówka

Podczas importu danych z arkusza Excela do programu MS Access możemy napotkać problemy. Może się na przykład zdarzyć, że podczas wprowadzania do arkusza danych liczbowych w jednej komórce wpiszemy `O` zamiast `0` albo słownie `dwa` zamiast `2`. Będzie to oznaczało, że przy imporcie danych kolumna z tymi danymi liczbowymi zostanie zaimportowana jako *Tekst*, a nie jako *Liczba*. Jak przekonwertować dane tekstowe na liczbowe? Włączamy widok projektu i zmieniamy typ danych z *Tekst* na *Liczba*. Zapisujemy zmiany w tabeli. Na skutek konwersji danych tekstowych na liczbowe pole zawierające tekst jest puste.

Czasami przy imporcie danych z Excela do MS Access otrzymujemy komunikat, że nie wszystkie dane zostały prawidłowo zaimportowane. Jak postępować w takim przypadku?

- Otwieramy tabelę, która zawiera błędy powstałe przy imporcie, oraz skoroszyt Excela, który importowaliśmy do Accessa.
- Na podstawie numeru wiersza odczytanego z tabeli *Błędy przy imporcie* lokalizujemy w arkuszu Excela komórkę, która spowodowała błędy. Dokonujemy korekty wpisanej

do niej wartości. Zapisujemy zmiany. Powyższą czynność przeprowadzamy dla wszystkich wierszy tabeli *Błędy przy imporcie*.

Ćwiczenie 2.8. Do bazy *uczelnia.accdb* zaimportuj te same dane na temat ocen, ale zamieszczone w pliku tekstowym *studenci_oceny_tekst.txt*, zapisanym na płycie CD. W grupie narzędzi *Importowanie* wybierz *Pliki tekstowe* (*.txt*, *.csv*, *.tab*, *.asc*) i zaimportuj plik tekstowy *studenci_oceny_tekst.txt*. W oknie *Kreator importu tekstu* pojawią się importowane dane, więc obserwując je, określ podstawowe opcje importu. W naszym przykładzie dane są rozdzielone średnikiem, wybierz więc opcję *Ograniczony - znaki takie jak przecinek czy tabulator rozdzielają pola*. Ogranicznikiem rozdzielającym pola jest średnik, tabela ma wiersz nagłówka, a zatem w następnym oknie kreatora zaznacz opcję *Pierwszy wiersz zawiera nazwy pól*. W dalszych krokach importu danych postępuj tak jak przy imporcie z Excela. Na koniec zatwierdź proponowaną nazwę tabeli *Studenci_oceny_tekst* i zakończ działanie kreatora importu.

Ćwiczenie 2.9. Po wykonaniu powyższych ćwiczeń w bazie danych znajdują się trzy tabele z ocenami studentów. Zobacz, jak wygląda ich projekt. W jaki sposób zostały rozpoznane dane pobrane z arkusza, a w jaki z pliku tekstowego? Porównaj je z oryginalną tabelą zaimportowaną z bazy danych. Zwróć uwagę na typy danych. Dwa pierwsze pola powinny być typu *Tekst*, pozostałe, zawierające oceny, to pola typu *Liczba*, o rozmiarze pola *Bajt*. Wykonaj korekty typów danych w tabelach zaimportowanych z arkusza i z pliku tekstowego.

Połącz relacją *jeden-do-jednego* (1-1) tabelę nadrzędną *Studenci_dane* z tabelą podrzędną *Studenci_oceny_z_bazy* (w oknie *Relacje* przycisk *Pokaż tabelę* daje możliwość dodania tabeli do relacji).

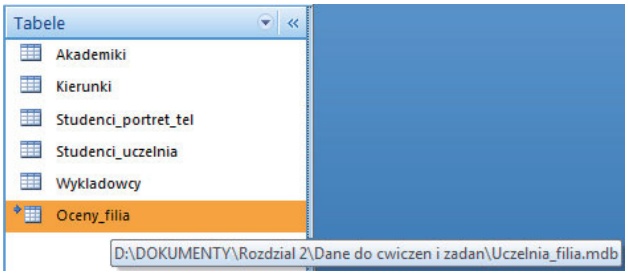
Zwróć uwagę na to, że w tabeli z ocenami powtórzone zostało pole *ID_kierunku* znajdujące się w tabeli *Studenci_uczelnia*. Czasami takie postępowanie jest podyktowane pewnymi względami. W tym przypadku studenci otrzymują oceny z różnych przedmiotów, zależnie od kierunku, na jakim studiują. Aby więc możliwa była kontrola nad prawidłowością wpisywania ocen wymaganych dla każdego studenta, została obok nich wyświetlona informacja o kierunku studiów. Można było to rozwiązać inaczej, na przykład na etapie projektu zaplanować, że w polu *ID_albumu* należy wpisać wartości, w których oprócz dwóch wymaganych liter `WU` (Uniwersytet Wirtualny) dodaje się jeszcze jedną wymaganą literkę oznaczającą kierunek studiów. Można również wpisać oceny studentów do różnych tabel, zgodnie z kierunkiem studiów.

Załączanie tabeli

Polecenie *Połącz ze źródłem danych, tworząc tabelę połączoną* pozwala umieścić w bazie MS Access tabelę wraz ze ścieżką dostępu do tabeli znajdującej się w bazie źródłowej. Wszystkie zmiany w tabeli bazy źródłowej są widoczne w bazie docelowej. W załączonej tabeli można modyfikować dane, ale nie można zmieniać jej struktury (projektu).

Zmiany wprowadzone w danych w tak utworzonej tabeli zostaną uwzględnione w źródle, i odwrotnie.

Aby załączyć tabelę do bazy danych z innej bazy danych Accessa, należy postępować tak jak przy imporcie tabeli. Jednak już w pierwszym kroku, po określeniu źródła danych, trzeba określić sposób i miejsce przechowywania danych po zaimportowaniu. Tu następuje połączenie ze źródłem danych, co powoduje utworzenie tabeli połączonej. Załączona tabela pojawi się na liście tabel. Obok jej ikony będzie widniała strzałka (rysunek 2.31).



Rysunek 2.31. Tabela załączona do bazy danych

Ćwiczenie 2.10. Załącz do bazy *uczelnia.accdb* tabelę *Oceny_filia* z pliku *uczelnia_filia.accdb* — wcześniej skopiuj go z płyty CD na dysk lokalny. W załączonej tabeli dokonaj zmiany ocen wybranego studenta. Zapisz zmiany w tabeli. Sprawdź, czy w źródłowej bazie *uczelnia_filia.accdb* również pojawiły się zmienione oceny.

Eksport

Operacją odwrotną do importu jest eksport danych z bazy Accessa do pliku utworzonego w innej aplikacji lub do innej bazy Accessa.

Aby wyeksportować dane z bazy danych Microsoft Office Access 2007 do innej bazy danych MS Access, należy wśród obiektów *Tabele* uaktywnić (wskazać) tę tabelę, którą chcemy wyeksportować. W grupie *Eksportowanie* na karcie *Dane zewnętrzne* zamieszczono przyciski przypisane do eksportu do różnych formatów plików wynikowych. Te same opcje eksportu otrzymamy w menu kontekstowym eksportowanej tabeli, w którym należy wybrać polecenie *Eksportuj…*.

W oknie *Eksportowanie* podajemy nazwę pliku, do którego chcemy wyeksportować dane (docelowy plik bazy musi istnieć). Pozostaje jeszcze zatwierdzenie domyślnej nazwy lub podanie nowej nazwy tabeli, do której dane mają być wyeksportowane, oraz zdecydowanie, czy eksport ma obejmować tylko definicje tabel (sam projekt), czy definicje i dane. Podobnie postępujemy przy eksporcie innych obiektów, na przykład kwerend.

Eksport danych do arkusza kalkulacyjnego różni się tym, że docelowy plik może nie istnieć i zostanie utworzony w momencie eksportu danych. Jeżeli jako plik docelowy wskażemy plik istniejący, eksport nastąpi do nowego arkusza w tym skoroszycie (pliku).

Przy eksporcie danych do pliku tekstowego domyślnie odznaczone są opcje eksportowania tabeli. Zatem ograniczamy działanie do wskazania miejsca i nazwy pliku tekstowego. Eksport danych do pliku tekstowego powoduje uruchomienie kreatora eksportu tekstu, w którym należy ustalić wszystko to, co musieliśmy wcześniej odczytać przy im-

porcie z pliku tekstowego, a mianowicie: czy dane będą ograniczone i jakiego znaku użyć do ograniczenia, czy w pierwszym wierszu dodać wiersz nagłówków pól i jak nazwać powstały plik z wyeksportowanymi danymi. Przy próbie zapisu danych do istniejącego pliku pojawi się pytanie o to, czy zastąpić istniejący plik. Nie można zatem drogą eksportu danych dopisać ich do danych istniejącego pliku tekstowego.

Ćwiczenie 2.11. Wyeksportuj dane z tabeli *Studenci_oceny_z_bazy* do bazy danych MS Access, arkusza kalkulacyjnego Excel oraz do pliku tekstowego. Pamiętaj o konieczności utworzenia pliku bazy danych MS Access przed eksportem tabeli. Sprawdź dwie możliwości eksportu do arkusza kalkulacyjnego istniejącego pliku Excela oraz do nowego skoroszytu.

W dalszej części podręcznika będziemy korzystać z tabeli z ocenami pobranej z bazy. Zmień nazwę tabeli *Studenci_oceny_z_bazy* na krótszą *Studenci_oceny*. Jeżeli nie udało Ci się wykonać wszystkich poleceń w bazie *uczelnia.accdb*, to na płycie dołączonej do podręcznika zamieszczono plik *uczelnia_OLE.accdb* z wszystkimi wykonanymi dotychczas ćwiczeniami. Skopiuj ten plik na dysk lokalny i zmień jego nazwę na *uczelnia.accdb*.

2.4. Kwerendy — zapytania

Istotą baz danych jest stworzenie możliwości efektywnego wyszukiwania informacji. Do tego celu służą **kwerendy**. Są one, obok tabel zawierających dane, podstawowymi **obiektami** definiującymi schematy wyszukiwania potrzebnych informacji. Program Access udostępnia dwa języki definiowania kwerend: język *SQL* (ang. *Structured Query Language*) oraz język *QBE* (ang. *Query By Example*).

Jeśli więc chcemy znaleźć i pobrać dane, które spełniają pewne warunki, tworzymy *kwerendę*, czasem nazywaną zapytaniem. Kwerenda może obejmować dane pochodzące z różnych tabel lub z wcześniej utworzonych kwerend (bądź jednych i drugich), wyszukiwać dane, aktualizować lub usuwać wiele rekordów jednocześnie, tworzyć nowe tabele. Pozwala także na wykonywanie obliczeń na danych oraz podsumowań danych.

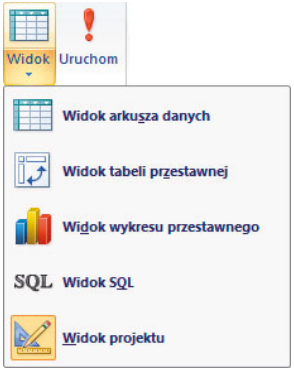
W naszej bazie danych znajduje się już jedna kwerenda, utworzona z zaawansowanego filtrowania danych, wybierająca studentów pochodzących z Olsztyna, urodzonych przed 12 marca 1989 roku. Przejdźmy więc do zakładki z kwerendami.

Po dwukrotnym kliknięciu myszą nazwy kwerendy następuje jej uruchomienie. Klawisz *Enter* lub polecenie *Otwórz* z menu kontekstowego tabeli również uruchamia kwerendy. W przypadku kwerendy *Studenci_z_Olsztyna <1989-03-12* nastąpi otwarcie kwerendy w postaci arkusza (tabelki) z danymi wyszukanymi zgodnie z kryteriami. Jest to zestaw danych dynamicznych, tworzony na czas uruchomienia kwerendy. Każda zmiana w tabeli, wyświetlonej po uruchomieniu kwerendy, oznacza zmianę w macierzystej tabeli. Sprawdź to, zmieniając datę urodzenia jednego ze studentów. Następnie zamknij kwerendę, przejdź do tabeli *Studenci_dane* i sprawdź, czy zmiana została zapisana w tej tabeli.

Kwerendę można również otworzyć w widoku jej *projektu*, wybierając polecenie *Widok projektu…* z menu kontekstowego kwerendy. Po takim otwarciu zobaczymy na Wstążce

znany zestaw poleceń zaawansowanego filtrowania danych — wybieraliśmy tu wcześniej pola i kryteria do filtrowania. Ta zakładka, *Projektowanie*, jest narzędziem kwerend.

Aby w widoku *projektu* uruchomić kwerendę, należy kliknąć przycisk Uruchom. Możliwe jest również wybranie polecenia *Uruchom* przy użyciu sekwencji następujących klawiszy: lewy *Alt* (który uaktywnia skróty klawiaturowe), następnie litera *R* (pozwala przejść do zakładki *Projektowanie*) i na koniec litera *H*.



Rysunek 2.32. Dostępne widoki kwerend

Kwerendy można oglądać jeszcze w innych widokach, pokazanych na rysunku 2.32.

My będziemy oglądać i projektować je w trzech widokach, wybieranych również z menu *Widok*:

- *Widok projektu* — wyświetla definicję zapytania w widoku siatki projektowej.
- *Widok arkusza danych* — wyświetla wynik wykonania zapytania w formie tabelki.
- *Widok SQL* — podaje definicję zapytania zapisaną w języku SQL.

Przeanalizujmy podstawowe typy kwerend na przykładach.

2.4.1. Kwerendy wybierające

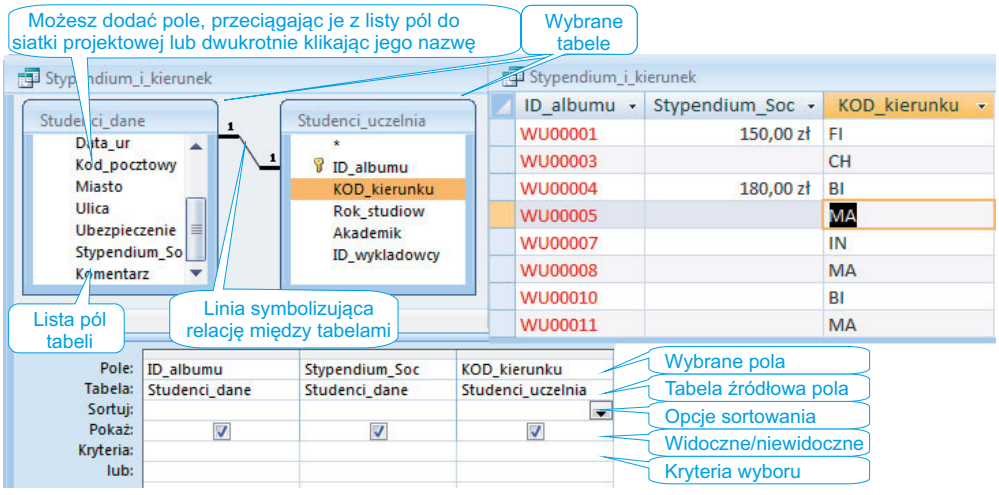
Najczęściej spotykanym rodzajem kwerendy jest kwerenda wybierająca. Pobiera ona dane z jednej tabeli lub z większej liczby tabel bądź z wcześniej zdefiniowanych i zapisanych kwerend, używając podanych kryteriów, a następnie wyświetla wybrane dane w żądanym porządku.

Kwerendę można utworzyć, używając kreatora kwerend lub samodzielnie zaprojektować ją w widoku projektu kwerendy. Oba narzędzia zamieszczono na zakładce *Tworzenie*, w grupie *Inne*.

W widoku projektu kwerendy przez dodanie do siatki projektowej pól zawierających dane z jednej tabeli lub z kilku tabel określamy poszczególne parametry wyszukiwania danych. Wypełnianie siatki projektowej przypomina zaawansowane filtrowanie danych. Podstawowa różnica polega na tym, że możemy korzystać z większej liczby tabel. Widok projektu kwerendy wybierającej zilustrowano na rysunku 2.33. Pomoże on nam w wykonaniu ćwiczenia 2.12.

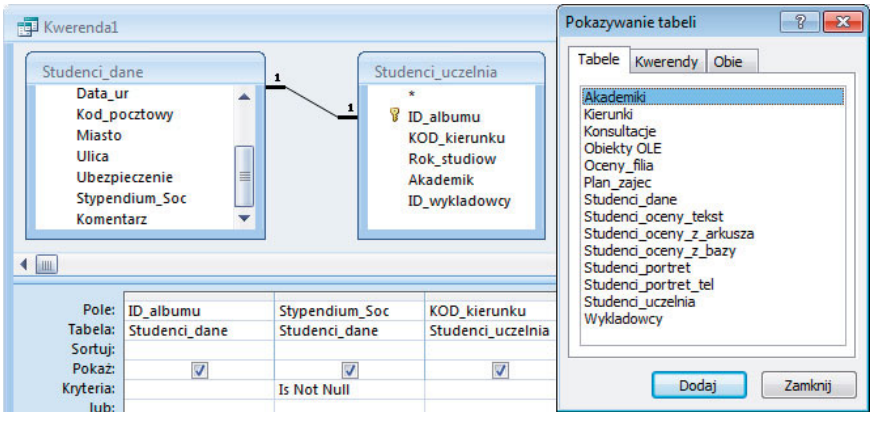
Ćwiczenie 2.12. Zaprojektuj kwerendę wybierającą studentów, którzy otrzymują stypendium. Obok numeru albumu wyświetl kierunek studiów oraz kwotę pobieranych świadczeń. Wybrane dane posortuj rosnąco według numeru indeksu.

Aby zaprojektować kwerendę, najpierw zdecyduj, które z tabel lub już istniejących kwerend będą dla niej źródłem danych. W naszym przykładzie będą to dwie tabele: *Studenci_dane* i *Studenci_uczelnia*. Przejdź do zakładki *Kwerendy* i z grupy *Inne* na karcie *Tworzenie*



Rysunek 2.33. Kwerenda z kryteriami wyboru wybierająca dane z dwóch tabel. Z lewej widok projektu, z prawej — arkusza danych

wybierz polecenie *Projekt kwerendy*. W oknie projektu kwerendy pojawi się, jak na rysunku 2.34, okno *Pokazywanie tabeli*. Dodaj niezbędne dwie tabele i zamknij okno. Następnie wybierz pola zawierające potrzebne informacje. Dodaj pola *ID_albumu*, *KOD_kierunku* oraz *Stypendium_Soc* do siatki projektowej, przeciągając je lub dwukrotnie klikając każde z nich. Możesz też wybrać pole i tabelę z rozwijanej listy bezpośrednio w siatce projektu. Zapisz utworzoną kwerendę pod nazwą *Stypendium_i_kierunek*. Uruchom kwerendę, klikając przycisk *Uruchom*. W dynamicznym zestawie danych pojawiły się wybrane pola tabeli. Zwróć uwagę, że w zestawie wybranych danych znaleźli się wszyscy studenci, również ci, którzy nie pobierają stypendium. Powróć do widoku *projektu*.



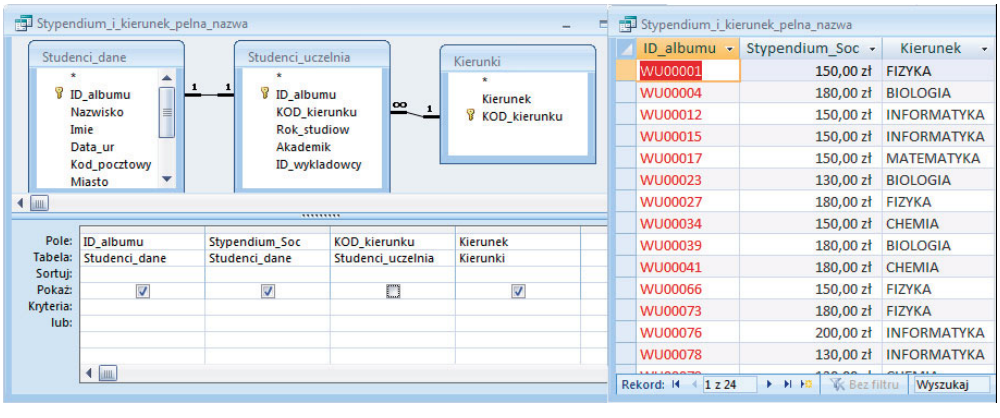
Rysunek 2.34. Okno wyboru tabel do projektu kwerendy

Gdyby wszyscy studenci pobierali stypendium w jednakowej wysokości, w kryteriach wyboru pola *Stypendium_Soc* można by podać tę wartość, na przykład `150`. Jednak po uruchomieniu kwerendy zniknęliby z zestawu nie tylko studenci bez stypendium, ale również ci, którzy pobierają stypendium inne niż 150 złotych. Ponieważ jest wiele róż-

nych kwot, korzystnym w tym przypadku kryterium wyboru jest odrzucenie wszystkich rekordów z pustym polem *Stypendium_Soc*. Kryterium *Is Null* spowoduje wypisanie wszystkich studentów, którzy nie pobierają stypendium (mają puste pole *Stypendium_Soc*), dlatego użyj zaprzeczenia i wpisz w kryterium wyboru *Is Not Null.* Zapisz kwerendę jako *Stypendium_i_kierunek* i uruchom ją. Powinien zostać wyświetlony wybrany zestaw danych tylko tych studentów, którzy pobierają stypendium.

Ćwiczenie 2.13. Zmodyfikuj utworzoną w ćwiczeniu 2.12 kwerendę wybierającą tak, aby zamiast kodu kierunku wypisywała pełną nazwę kierunków studiów każdego studenta pobierającego stypendium socjalne. Zmodyfikowaną kwerendę zapisz jako nową kwerendę i nadaj jej nazwę *Stypendium_i_kierunek_pelna_nazwa*.

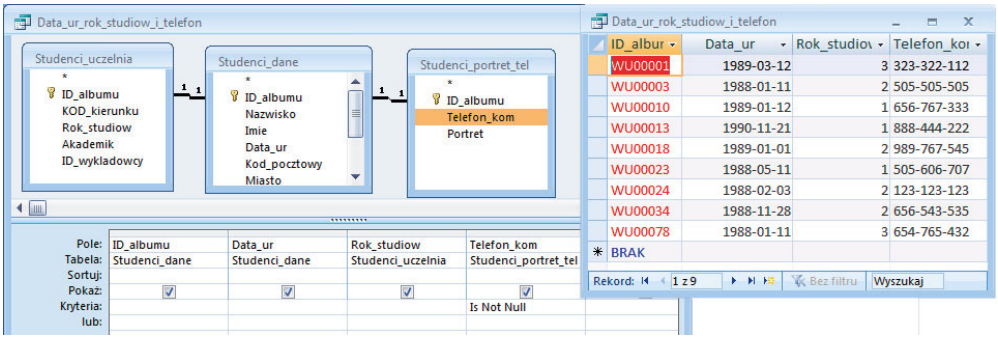
Do wykonania tego ćwiczenia użyj utworzonej wcześniej kwerendy *Stypendium_i_kierunek*. Otwórz ją w widoku *projektu*. Aby wyświetlić pełne nazwy kierunków zamiast ich kodów, potrzebna jest tabela *Kierunki*. Dołóż ją do tabel. W tym celu przyciskiem *Pokaż tabelę* (polecenie dostępne również w menu kontekstowym) otwórz listę dostępnych tabel i dodaj do projektu kwerendy tabelę *Kierunki*. Następnie pole *Kierunek* dodaj do siatki projektowej. Zapisz utworzoną kwerendę (*Przycisk pakietu Office/Zapisz jako…*) pod nazwą *Stypendium_i_kierunek_pelna_nazwa*. Uruchom ją. Zauważ, że niepotrzebnie wyświetlane są teraz dwie informacje o kierunku studiów: w postaci kodu i w postaci pełnej nazwy. W widoku *projektu* usuń zaznaczenie *Pokaż* dla pola *KOD_kierunku* i ponownie uruchom kwerendę. Teraz efekt jest taki jak na rysunku 2.35.



Rysunek 2.35. Okno projektowe korygowanej kwerendy i dynamiczny zestaw danych bez ukrytego pola KOD_kierunku

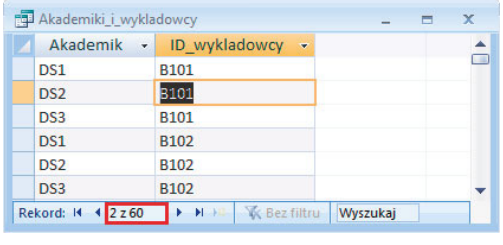
Przyjrzyj się uważnie utworzonej siatce projektowej oraz relacjom między tabelami danych źródłowych użytymi w kwerendzie. Zauważ, że tabela *Kierunki* jest sprzężona z tabelą *Studenci_uczelnia* przez pole *KOD_kierunku*. Nie ma więc potrzeby zamieszczania pola *KOD_kierunku* w siatce projektowej. Jeśli usuniemy to pole, efekt wyboru rekordów będzie dokładnie taki sam. W widoku *projektu* zaznacz więc w siatce projektowej pole *KOD_kierunku* i usuń je przyciskiem *Delete* z klawiatury lub używając menu podręcznego. Ponownie uruchom kwerendę. Nic się nie zmieniło. Zapisz zmiany i zamknij okno kwerendy.

Ćwiczenie 2.14. Zbierz dane o studentach posiadających telefon. Dane wybierz z trzech tabel, jak na rysunku 2.36. Wybranych studentów posortuj w kolejności od najmłodszego do najstarszego. Utworzoną kwerendę wybierającą zapisz pod nazwą *Data_ur_rok_studiow_i_telefon*. Który student jest ostatni na liście? Ilu studentów podało swoje numery telefonów?



Rysunek 2.36. Kwerenda wybierająca dane z trzech tabel do ćwiczenia 2.14

Przyjrzyj się zawartości dwóch tabel: *Akademiki* i *Wykladowcy*. Zawierają one, odpowiednio: 3 rekordy i 20 rekordów. Między nimi nie ma ustanowionej relacji. Co się stanie, jeśli utworzysz kwerendę, której wybranymi polami będą: *Akademik* i *ID_wykladowcy*? Zaprojektuj taką kwerendę, zapisz ją pod nazwą *Akademiki_i_wykladowcy*. Nie stosuj żadnych kryteriów wyboru. Uruchom kwerendę i zobacz, ile powstało par danych. Stało się tak właśnie ze względu na brak relacji między tymi dwiema tabelami. Powstały wszystkie możliwe kombinacje 3 akademików i 20 wykładowców, czyli 60 zestawów danych, co ilustruje rysunek 2.37.



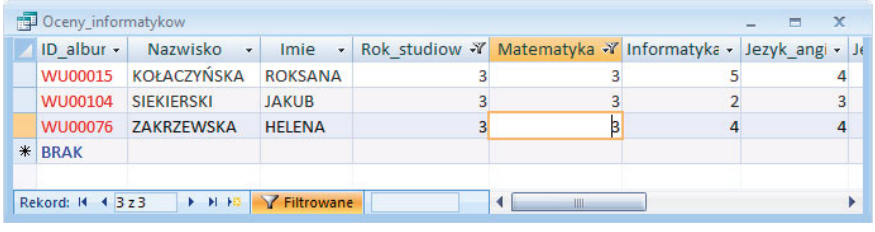
Rysunek 2.37. Kwerenda wybierająca rekordy z tabel niepowiązanych relacjami

Ćwiczenie 2.15. Zaprojektuj kwerendę, która będzie wybierała imiona i nazwiska studentów informatyki oraz ich oceny. Wybrane dane posortuj według nazwisk.

Do wykonania tego ćwiczenia użyj dwóch tabel: *Studenci_dane* i *Studenci_oceny*. Dodaj pola z nazwiskiem i imieniem, kierunkiem studiów oraz pola z ocenami. Ustal kryterium wyboru. W tej kwerendzie będzie to kryterium tekstowe, które sprowadza się do podania w komórce *Kryteria* pola *KOD_kierunku* odpowiedniego ciągu znaków, czyli `"IN"`. Posortuj wybrane dane według nazwisk. Zapisz utworzoną kwerendę jako *Oceny_informatykow*. Uruchom kwerendę i sprawdź jej działanie. Wyłącz pokazywanie kodu kierunku. Zobacz efekt działania kwerendy. Zauważ, że informatycy nie mają ocen ze wszystkich przedmiotów. Usuń z projektu te przedmioty nauczania. Aby zaznaczyć kolumnę (pole), ustaw wskaźnik myszy nad nazwą pola — przybierze on kształt czarnej strzałki. Zaznacz pole i usuń je z projektu kwerendy (*Delete*).

Dobrze byłoby również ująć w tym wykazie numer ID oraz rok studiów danego studenta, co wiąże się z dodaniem trzeciej tabeli, *Studenci_uczelnia*, w której znajduje się informacja o roku studiów. Uzupełnij to w widoku *projektu* kwerendy. Zapisz wprowadzone zmiany.

Po uruchomieniu kwerendy możesz na wyświetlonym, dynamicznym zestawie danych zastosować filtry. Którzy studenci trzeciego roku otrzymali najniższą ocenę z matematyki? Ustaw kursor w polu, w którym jest ocena `3` z matematyki, i zastosuj filtr dla zaznaczenia *Równa się 3*. Następnie zastosuj filtr dla pola *Rok_studiow*. Fragment zestawu przefiltrowanych w ten sposób danych tej kwerendy znajduje się na rysunku 2.38.


Oceny_informatykow

ID_albur	Nazwisko	Imie	Rok_studiow	Matematyka	Informatyka	Jezyk_angi
WU00015	KOŁACZYŃSKA	ROKSANA	3	3	5	4
WU00104	SIEKIERSKI	JAKUB	3	3	2	3
WU00076	ZAKRZEWSKA	HELENA	3	3	4	4
BRAK						

Rekord: 3 z 3 Filtrowane

Rysunek 2.38. Fragment zestawu przefiltrowanych danych kwerendy wybierającej oceny studentów informatyki

Definiowanie złożonych kryteriów wyboru

W przypadku wszystkich typów pól najprostszym kryterium jest podanie konkretnej wartości:

`"WARSZAWA"` — dla pola typu *Tekst*,

`50` — dla pola liczbowego (*Autonumerowanie*, *Liczba*, *Walutowy*),

`#1995-12-25#` lub `#1995/12/25#` — dla pola typu *Data/Godzina*,

`1` lub `0` (albo `tak` lub `nie`) — dla pola logicznego *Tak/Nie* (*Tak* odpowiada *1*, *Nie* odpowiada *0*).

Ćwiczenie 2.16. Zaprojektuj kwerendę, która wybierze wszystkich ubezpieczonych studentów. Jako wynik wyszukiwania podaj ID studenta oraz kierunek i rok studiów.

W widoku *projektu* wybierz odpowiednie tabele i cztery pola. W polu *Ubezpieczenie* wpisz kryterium, które pozwoli wybrać wszystkich ubezpieczonych studentów. Sprawdź działanie kwerendy. Jeśli wybór jest prawidłowy, wyłącz wyświetlanie wartości pola *Ubezpieczenie*, ponieważ nie ma potrzeby wyświetlania całej kolumny z napisem *tak*. Zapisz kwerendę pod nazwą *Ubezpieczeni*.

W kryteriach tekstowych można stosować *znaki wieloznaczne* (zastępcze), które zastępują wiele znaków lub jeden znak. Należą do nich `*`, `?` i `#`. Znak `*` zastępuje ciąg znaków dowolnej długości. Aby zrealizować kryterium wyboru wszystkich studentów, których nazwisko rozpoczyna się na literę J, należy w polu *Kryteria* wpisać `"J*"` lub `Like "J*"`. Wystarczy wpisać `"J*"`, a Access automatycznie uzupełni to kryterium do pełnej postaci, co zauważysz po ponownym otwarciu kwerendy w widoku *projektu*. Natomiast zapis `Like "*a"` spowoduje wybór tych rekordów, które w danym polu tekstowym na ostatniej pozycji ciągu znaków mają literę `"a"`. Tego zapisu często używa się jako kryte-

rium dla pola *Imie* do wyboru płci żeńskiej, zakładając, że zwykle w Polsce imiona kobiet kończą się na literę `a` (choć nie jest to regułą — coraz częściej nadawane są imiona obcego pochodzenia, na przykład imię żeńskie Inez nie spełnia tego warunku, a imię męskie Bonawentura jest błędnie traktowane jako imię żeńskie).

Oprócz `*` w kryteriach można użyć znaku `?`, zastępującego pojedynczy znak, oraz znaku `#` wpisywanego na miejscu cyfry. Przykładowo kryterium `"?????"` wymusza poszukiwanie wszystkich osób, których nazwisko składa się z 5 znaków, `"KO?KIEWICZ"` wyszuka takie nazwiska jak: KO**L**KIEWICZ, KO**R**KIEWICZ, KO**T**KIEWICZ, ale nie wybierze nazwiska KOKIEWICZ. W roli kryterium dla pola liczbowego można używać zarówno `"19##-01-01"`, jak i `"19??-01-01"`, w obu przypadkach otrzymamy wszystkich urodzonych 1 stycznia w latach 1900-1999. Aby wziąć pod uwagę również osoby z młodszych roczników, należałoby wpisać kryterium `"####-01-01"` lub `"????-01-01"`. Natomiast kryterium dla daty `"1989-??-??"` lub `"1989-##-##"` pozwala wybrać wszystkich urodzonych w 1989 roku. Można tych dwóch znaków, `?` i `#`, używać tutaj wymiennie, ponieważ dowolnymi znakami w polu typu *Data* na miejscu zastąpionym znakami `?` lub `#` są zawsze cyfry, więc nie popełnimy błędu i nastąpi poprawne wybranie rekordów zgodnie z kryteriami. Jednak jeśli wybór miałby dotyczyć cyfry w polu typu *Tekst*, to bezwzględnie użyjemy znaku `#`, dla odróżnienia jej od dowolnego znaku. Na przykład wpis `"*#"` w polu *Miasto* oznacza kryterium wyboru miast, które mają cyfrę na końcu nazwy, na przykład LUBICZ1.

W kryteriach wyboru można również podać zakres `[]`, w jakim musi znaleźć się poszukiwany rekord. Wpis `Like "[A-D]*"` oznacza, że wszystkie wybrane nazwy będą rozpoczynały się jedną z liter z podanego zakresu, więc A, B, C lub D. Wpis `Like "*[krs]"` oznacza, że wszystkie wybrane nazwy będą kończyły się jedną z liter podanych w zakresie, czyli k, r lub s. Korzystanie z zakresów znacznie przyspiesza definiowanie złożonych kryteriów, które można zastąpić złożoną alternatywą.

Oto znaczenie poszczególnych znaków:

- `*` — zastępuje dowolny ciąg znaków (również ciąg pusty);
- `?` — zastępuje jeden znak, który musi wystąpić w polu;
- `#` — zastępuje cyfrę, która musi wystąpić w polu;
- `[]` — zakres, do jakiego musi należeć dany znak;
- `[!]` — zakres, do jakiego nie może należeć dany znak. Wykrzyknik oznacza negację (na przykład `"[!act]*"` oznacza, że ciąg może zaczynać się na dowolną literę oprócz `a`, `c` lub `t`).

Określenie złożonego kryterium liczbowego wymaga użycia jednego z sześciu operatorów relacji: `>`, `>=`, `<`, `<=`, `=`, `<>` (różne).

Obok nieskomplikowanych warunków wyboru można tworzyć rozbudowane kryteria, łączące co najmniej dwa warunki. Są to: *koniunkcja kryteriów*, czyli wymóg spełnienia wszystkich warunków, oraz *alternatywa kryteriów*, czyli wymóg spełnienia co najmniej jednego warunku. Kryteria można łączyć operatorami logicznymi:

- `And` — koniunkcja, `X And Y` zwraca prawdę, gdy `X` i `Y` są prawdziwe;
- `Or` — alternatywa, `X Or Y` zwraca prawdę, gdy przynajmniej jedno z nich (`X` lub `Y`) jest prawdziwe;
- `Not` — zaprzeczenie, `Not X` zwraca prawdę, gdy `X` jest fałszywe;
- `In ("tekst1"; "tekst2"; "tekst3"; "tekst4")` — wyświetlenie rekordów, które w danym polu mają wartości należące do listy. To kryterium dotyczy danych tekstowych i zastępuje złożoną alternatywę;
- `Between` *wartość_początkowa* `And` *wartość_końcowa* — wyświetlenie wszystkich rekordów, które mają wartości z przedziału od wartości początkowej do wartości końcowej włącznie. Na przykład `Between 10 And 100` oznacza wybór rekordów mających we wskazanym polu wartość z przedziału `<10,100>`.

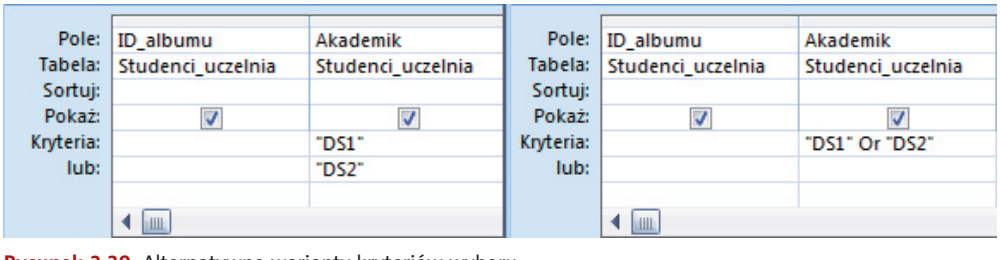
Wskazówka

Wielkość liter występujących w kryteriach nie ma znaczenia.

Ćwiczenie 2.17. W naszej przykładowej bazie danych utwórz dwie kwerendy wybierające. Pierwsza ma wybierać z tabeli *Studenci_dane* studentów urodzonych w pierwszym kwartale 1989 roku. Wykaz ma zawierać ID, imiona i nazwiska studentów. Na początku listy zamieść najstarszego studenta (`Zakrzewska`), a na końcu najmłodszego (`Olek`).

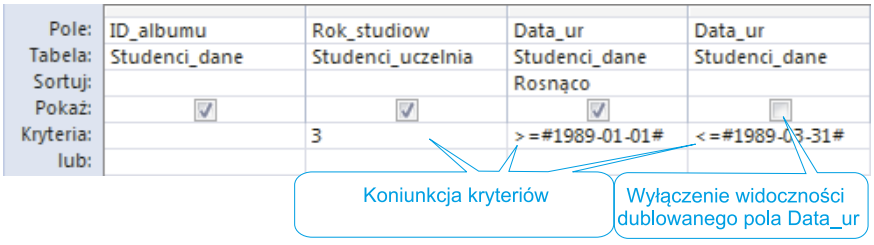
Datę w kryterium można wpisać w sposób następujący: `>=1989-01-01 And <=1989-03-31`. Po uruchomieniu kwerendy data automatycznie zostanie ujęta w znacznik `#data#` i kryterium przyjmie następującą postać: `>=#1989-01-01# And <=#1989-03-31#`. Utworzoną kwerendę zapisz pod nazwą *Urodzeni_w_pierwszym_kwartale_1989_roku*.

W następnej kwerendzie wybierz z tabeli *Studenci_uczelnia* studentów mieszkających w akademiku *DS1* lub *DS2*. To zadanie można zrealizować na dwa sposoby. Jeden to podanie alternatywy kryterium wyboru `"DS1" Or "DS2"`, drugi to wpisanie dwóch warunków do kolejnych pól kryteriów wyboru w siatce projektowej. Obie możliwości pokazano na rysunku 2.39. Kwerendę zapisz pod nazwą *Studenci_z_DS1_i_DS2*.



Rysunek 2.39. Alternatywne warianty kryteriów wyboru

Przy definiowaniu kryteriów należy pamiętać o następujących regułach:
- kryteria wprowadzone w polach wiersza *Kryteria* połączone są operatorem logicznym `And`, czyli **koniunkcją**, której przykład zobrazowano na rysunku 2.40;



Rysunek 2.40. Koniunkcja wyboru studentów trzeciego roku urodzonych w pierwszym kwartale 1989 roku

Wskazówka

Wszystkie wymienione powyżej kryteria mogą być stosowane jako reguły poprawności w definicjach pól (ich właściwości) tabeli i formularzy.

- kryteria dla jednego pola, wprowadzone w wierszach *Kryteria* i *lub* oraz w każdym następnym wierszu (kryteriów może być więcej niż dwa), połączone są operatorem logicznym `Or`, czyli **alternatywą**.

Definiowanie wyrażeń

Definicja

Wyrażenie to dowolna kombinacja operatorów matematycznych i logicznych, stałych, funkcji, nazw pól, formantów i właściwości, której wynikiem jest pojedyncza wartość. Wyrażenia służą do przeprowadzania obliczeń, operowania znakami lub testowania danych.

Kwerendy mogą wyświetlać pola tabel lub innych utworzonych wcześniej kwerend wybierających oraz **wyrażenia** arytmetyczne, tekstowe i inne. Wyrażenia takie definiowane są w siatce projektowej, w wierszu *Pole*.

Wyrażenie składa się z dwóch części: *tytułu* (nazwa pola) i *formuły* (reguła, według której obliczana jest wartość w tym polu), które zapisujemy, rozdzielając je dwukropkiem. Na formułę może składać się kombinacja pól tabeli źródłowej, funkcji Accessa, operatorów arytmetycznych (`+`, `-`, `/`, `*`) i innych. Ogólna postać wyrażenia jest następująca:

```
tytuł_wyrażenia: formuła
```

W wyrażeniach nazwy pól z tabeli źródłowej podajemy w nawiasach kwadratowych.

Ćwiczenie 2.18. Zaprojektuj kwerendę, która połączy zawartości pól tekstowych *Imie* i *Nazwisko* oddzielone pojedynczą spacją. Powstałą kwerendę zapisz pod nazwą *Dane_osobowe_studentow*.

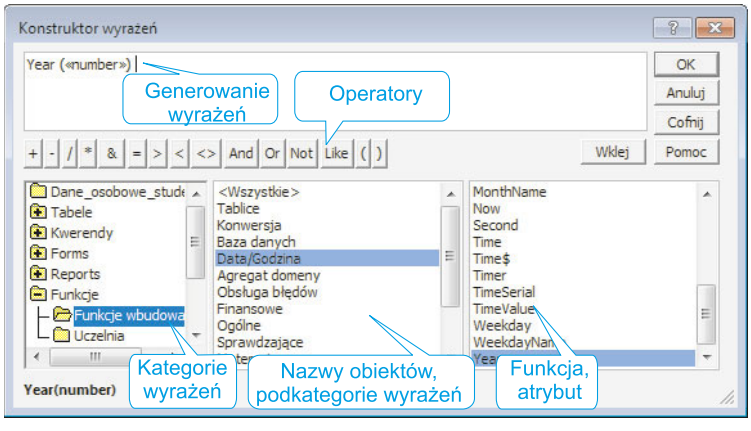
W oknie projektu nowej kwerendy dodaj tabelę *Studenci_dane*. Wybierz z niej trzy pola: *ID_albumu*, *Nazwisko* i *Imie*. W nowym polu siatki projektowej wpisz następujące wyrażenie: `[Imie] & " " & [Nazwisko]` i uruchom powstałą kwerendę. Zauważ, jak jest nazwane pole, które połączyło dwa pola tekstowe. Powróć do projektu. Aby lepiej widzieć powstałe wyrażenie, wybierz polecenie *Powiększenie…* i w otwartym okienku edycji wyrażenia zmień nazwę `Wyr1` na `Imie_i_nazwisko` (nie usuwaj dwukropka). Teraz wyrażenie

ma następującą postać: `Imie_i_nazwisko: [Imie] & " " & [Nazwisko]`. Posortuj listę według nazwisk i wyłącz pokazywanie nazwiska i imienia w osobnych polach. Ponownie uruchom kwerendę. Poszerz kolumnę, aby można było przeczytać dane połączone w jednym polu. Zapisz kwerendę pod nazwą *Dane_osobowe_studentow*.

Wskazówka

Aby widzieć całe wyrażenie, ustaw kursor w polu wyrażenia i wybierz polecenie *Powiększenie…* (*Shift+F2*).

W tworzeniu wyrażeń dużą pomoc oferuje *Konstruktor wyrażeń*, uruchamiany poleceniem *Konstruktor* znajdującym się w grupie narzędzi *Konfiguracja kwerendy*. Można go również uruchomić, wybierając opcję *Konstruuj…* z menu kontekstowego pustego pola lub pola z już istniejącym wyrażeniem. Polecenie otwiera okno *Konstruktor wyrażeń*, pokazane na rysunku 2.41.



Rysunek 2.41. Okno Konstruktor wyrażeń. Konstruowanie wyrażenia z wykorzystaniem wbudowanej funkcji

Konstruktor wyrażeń jest narzędziem, dzięki któremu można generować skomplikowane wyrażenia bez ryzyka popełnienia pomyłki literowej w nazwach pól lub błędu w składni funkcji.

Uwaga! Pola użyte w kwerendzie będą widziane w oknie *Konstruktor wyrażeń* dopiero po zapisaniu kwerendy.

Ćwiczenie 2.19. Zmodyfikuj utworzoną wcześniej kwerendę *Ubezpieczeni* tak, aby wyświetlić rok urodzenia każdego ubezpieczonego studenta. Wykorzystaj do tego *Konstruktor wyrażeń* oraz funkcję wbudowaną `Year` — zwracającą rok z pola zawierającego datę. Kwerendę zapisz pod nazwą *Rok_urodzenia_studentow*.

Otwórz w widoku *projektu* kwerendę *Ubezpieczeni*. Do siatki projektowej dodaj pole *Data_ur*. Zapisz kwerendę pod nazwą *Rok_urodzenia_studentow*. Zapis jest konieczny, aby było widać dodane przed chwilą pole. W następnym, pustym polu wybierz polecenie *Konstruuj…* i wstaw funkcję `Year`, zgodnie z rysunkiem 2.41. W oknie generowania wyrażenia pojawi się treść: `Year («number»)`. Teraz zamiast `«number»`, czyli w miejscu parametru funkcji, wstaw pole *Data_ur*, wybierając je w dolnej części okna *Konstruktor wyrażeń*, z bieżącej kwerendy, poleceniem *Wklej* lub dwukrotnie klikając nazwę pola.

W wyrażeniu pojawi się nazwa pola w kwadratowym nawiasie, a całe wyrażenie powinno wyglądać następująco: `Year([Data_ur])`. Oczywiście można wpisywać nazwy samodzielnie, bez wybierania z listy, jednak niesie to ze sobą niebezpieczeństwo niedokładnego podania nazwy pól lub funkcji, co w konsekwencji doprowadzi do błędu w obliczeniach. Uruchom utworzoną kwerendę i sprawdź jej działanie. Kwerenda działa prawidłowo, jednak utworzone pole otrzymało nazwę *`Wyr1`*. Powróć do widoku *projektu* i zmień nazwę pola *`Wyr1`* na `Rok:` (nie zapomnij o dwukropku). Wyrażenie powinno wyglądać następująco: `Rok: Year([Data_ur])`. Zapisz zmiany.

Kilka przydatnych funkcji zamieszczonych w tabeli 2.7 przyspieszy i ułatwi pracę. Pamiętaj, że pomoc otrzymasz po wpisaniu w indeksie haseł odpowiedniej nazwy funkcji.

Tabela 2.7. Przydatne wbudowane funkcje MS Access

`Date()`, `Now()`	Podanie bieżącej daty lub daty i czasu
`DateAdd(...)`, `DateDiff(...)`	Obliczanie różnic i sum dat w dniach, latach itp.
`Day(...)`, `Year(...)`, `Month(...)`, `Minute(...)`	Wyodrębnienie z daty jej części
`IsDate(...)`, `IsNull(...)`	Sprawdzenie, czy dany obiekt jest, odpowiednio, datą, wartością `Null` itp.
`Left(...)`, `Right(...)`, `Mid(...)`	Wybieranie fragmentu napisu z tekstu. `(Left([ID_albumu],2)` zwróci `WU`
`LTrim(...)`	Usuwanie spacji
`Nz(…)`	Konwersja wartości `Null` na `0` lub `""`
`Choose(...)`	Wybieranie z listy wartości elementu o podanym numerze
`IIf(...)`	Wybranie spośród dwu wartości, zależnie od warunku logicznego
`Format(...)`	Przekształcenie podanej wartości w napis według zadanego formatu

Wśród funkcji, których parametrem są data i czas, znajduje się funkcja `Date()`, zwracająca aktualną datę. Zaprojektuj kwerendę, która obliczy wiek studenta według działania: *Wiek = rok (dzisiejsza data) – rok (data urodzenia)*. Odpowiednie wyrażenie utwórz wykorzystując *Konstruktor wyrażeń*. Oprócz wieku wyświetl *ID_studenta* oraz kierunek studiów. Utworzoną kwerendę zapisz pod nazwą *Wiek_studentow*.

W tym przykładzie parametrem funkcji `Year` jest funkcja `Date()`. Wyrażenie będzie więc miało następującą postać: `Wiek: Year(Date())-Year([Data_ur])`.

Ćwiczenie 2.20. Zaprojektuj kwerendę, która obliczy i wyświetli średnią ocen studentów informatyki. Otrzymany zestaw danych wyświetl w kolejności malejącej średniej. Kwerendę zapisz pod nazwą *Srednia_ocen_informatykow*. Do utworzenia wyrażenia wykorzystaj *Konstruktor wyrażeń*, który ułatwi i przyspieszy podanie nazw wszystkich pól zawierających oceny z poszczególnych przedmiotów. Oprócz średniej, obliczonej z dwoma miejscami dziesiętnymi, wyświetl rok studiów, kod kierunku oraz imię i nazwisko studenta zdefiniowane w kwerendzie *Dane_osobowe_studentow*.

Aby obliczyć średnią wartość danych zawartych w kilku polach, napisz wyrażenie zawierające sumę tych pól podzieloną przez liczbę przedmiotów sumowanych. W widoku *projektu* kwerendy dodaj tabelę *Studenci_uczelnia* oraz kwerendy *Dane_osobowe_studentow* i *Oceny_informatykow*. Dodaj niezbędne pola do siatki projektowej (*ID_albumu*, *Imie_i_nazwisko*, *Rok_studiow*, *Kod_kierunku*, *Matematyka*, *Informatyka*, *Jezyk_angielski*, *Jezyk_niemiecki*, *Przedsiebiorczosc*, *Filozofia*, *WF*). Zapisz kwerendę (ważne, aby pola były widoczne). W nowym polu poleceniem *Konstruuj…* uruchom *Konstruktor wyrażeń*. Utwórz wyrażenie obliczające średnią ocen: `Srednia_ocen:([Matematyka]+[Informatyka]+[Jezyk_angielski]+[Jezyk_niemiecki]+[Przedsiebiorczosc]+[Filozofia]+[WF])/7`, ustal sposób sortowania i wyłącz widok pól z ocenami. Kwerenda po uruchomieniu wyświetla obliczone średnie. Jeżeli zamiast liczb wyświetliły się krzyżyki `######`, oznacza to, że kolumna jest za wąska i należy ją poszerzyć. A co z tak dużą liczbą cyfr po przecinku? Sposób wyświetlania wyniku należy ustalić we właściwościach pola, dostępnych w menu kontekstowym pola, w siatce projektowej, w widoku *projektu* kwerendy. W widoku *projektu*, w grupie *Pokazywanie/ukrywanie*, również jest dostępna opcja wyświetlenia *Arkusz właściwości*. Dopilnuj jednak, żeby aktywnym polem było to z wyrażeniem obliczającym średnią. W arkuszu właściwości ustal *Format* liczby na *Standardowy* i *2* miejsca dziesiętne. Zapisz kwerendę i uruchom ją.

Po zamknięciu i ponownym otwarciu tej kwerendy w widoku *projektu* nie ma nazw przedmiotów, jest tylko wyrażenie obliczające średnią. Wyłączona została widoczność pól, a ponieważ tabele są połączone ze sobą relacjami, nie ma potrzeby zamieszczania nazw przedmiotów w projekcie tej kwerendy. Wstawiając pola do wyrażenia, można było pobierać je bezpośrednio z kwerendy *Oceny_informatykow*.

Ćwiczenie 2.21. Oblicz średnie ocen wszystkich studentów, biorąc pod uwagę sześć przedmiotów, z których wszyscy studenci mają oceny (matematyka, język angielski i język niemiecki, przedsiębiorczość, filozofia i wychowanie fizyczne). Wykonaj obliczenia na grupach rekordów i wyświetl tylko niezbędne dane, jak na rysunku 2.42. Utworzoną kwerendę zapisz pod nazwą *Srednia_ocen*.

Srednia_ocen

ID_albumu	Imie_i_nazwisko	KOD_kierunku	Rok_studiow	Srednia_ocen
WU00001	ADRIANNA MAKOWSKA	FI	3	3,83
WU00003	OLIWIA RUMIŃSKA	CH	2	3,33
WU00004	DOMINIK SOCHATZKI	BI	1	4,00
WU00005	ALEKSANDRA KWIATKOWSKA	MA	2	4,33
WU00007	MICHAŁ MIKOŁAJEWICZ	IN	2	4,50

Rekord: 3 z 71 Bez filtru Wyszukaj

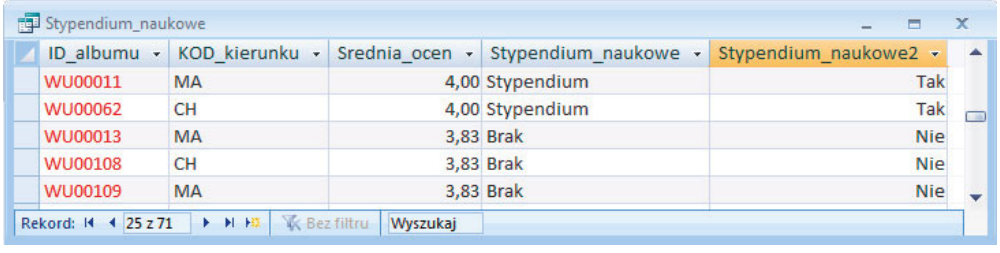
Rysunek 2.42. Kwerenda wybierająca dane z polem obliczeniowym

Ćwiczenie 2.22. Zaprojektuj kwerendę, która na podstawie średnich ocen obliczonych w ćwiczeniu 2.21 wypisze obok ID studenta i kierunku studiów średnią ocen wraz z informacją, czy student otrzyma stypendium naukowe, czy też nie. Kryterium przyznania stypendium to średnia ocen, która wynosi **minimum 4,0**. Jeśli studentowi przysługuje

stypendium, w polu wpisz słowo *Stypendium*, w przeciwnym razie w polu pojawi się napis *Brak*. Utworzoną kwerendę zapisz pod nazwą *Stypendium_naukowe*.

W widoku *projektu* nowej kwerendy użyj — tym razem zamiast tabeli — utworzonej wcześniej kwerendy *Srednia_ocen*. Wybierz z niej trzy niezbędne pola: *ID_albumu*, *KOD_kierunku* oraz *Srednia_ocen*. Zapisz kwerendę. W nowym polu utwórz wyrażenie, wykorzystując okno *Konstruktor wyrażeń*. Wśród funkcji wbudowanych, zgrupowanych w bloku funkcji przepływu sterowania programu, znajduje się funkcja wyboru `IIf`, która w oknie *Konstruktor wyrażeń* pojawi się jako: `IIf(«expr»; «truepart»; «falsepart»)`. Należy ją rozumieć następująco: `IIf(warunek;wartość_gdy_prawda,wartość_gdy_fałsz)`. W oknie wyrażenia wpisz brakującą nazwę pola oraz parametry funkcji: `Stypendium_naukowe: IIf([Srednia_ocen]>=4;"Stypendium";"Brak")`. Sprawdź poprawność działania kwerendy.

W kolejnym polu, które nazwij *Stypendium_naukowe2*, ponownie wykorzystaj funkcję warunkową. Podaj w niej wartości liczbowe `1` i `0`, odpowiednio: dla spełnionego i niespełnionego warunku przydzielenia stypendium. W oknie wyrażenia wpisz nazwę pola oraz parametry funkcji: `Stypendium_naukowe2: IIf([Srednia_ocen]>=4;1;0)` i zobacz wynik działania kwerendy. Powróć do widoku *projektu* i zmień właściwości dla tego pola, ustalając format pola najpierw jako typ *Prawda/Fałsz*, później jako typ *Tak/Nie*. Zobacz, jakie wartości liczbowe odpowiadają wartościom *Tak/Nie*. Ustal sposób sortowania według średniej ocen. Wynik działania tej kwerendy został zilustrowany na rysunku 2.43.


Stypendium_naukowe

ID_albumu	KOD_kierunku	Srednia_ocen	Stypendium_naukowe	Stypendium_naukowe2
WU00011	MA	4,00	Stypendium	Tak
WU00062	CH	4,00	Stypendium	Tak
WU00013	MA	3,83	Brak	Nie
WU00108	CH	3,83	Brak	Nie
WU00109	MA	3,83	Brak	Nie

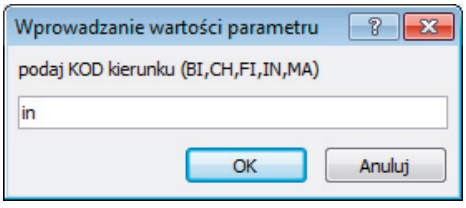
Rysunek 2.43. Wynik działania kwerendy z wyrażeniem zawierającym funkcję warunkową IIf

Kwerendy parametryczne

Zapytania mogą być tak konstruowane, aby wartości określające kryterium wyboru podawane były dopiero w momencie uruchamiania kwerendy, bez każdorazowej zmiany kryteriów w siatce projektowej. Takie rozwiązanie wymaga, aby w polu z kryteriami lub w wyrażeniu użyć parametru, podając jego nazwę w nawiasach kwadratowych `[]`. Parametrów może być kilka.

Ćwiczenie 2.23. Zaprojektuj kwerendę, która będzie wyświetlała ID i miejscowość zamieszkania wszystkich studentów z danego kierunku studiów. Przy każdym uruchomieniu kwerendy będzie można po podaniu kodu kierunku wyświetlić dane jego studentów. Utworzoną kwerendę zapisz pod nazwą *Studenci_wg_kierunku_z_parametrem*.

W widoku *projektu* dodaj tabele z danymi adresowymi i uczelnianymi studentów. Wybierz niezbędne pola. W kryterium wyboru pola *KOD_kierunku* podaj kryterium zawierające parametr. Wystarczy wpisać w nawiasach kwadratowych treść komunikatu `[podaj KOD kierunku]`, który pojawi się na ekranie w oknie wprowadzania parametru. Zapisz kwerendę i sprawdź jej działanie. Można ułatwić sobie wypełnienie pola przez podanie podpowiedzi z wypisanymi kodami kierunków, jak na rysunku 2.44. Kryterium przyjmie wówczas postać: `[podaj KOD kierunku (BI,CH,FI,IN,MA)]`. Nie ma konieczności wpisywania parametrów wielkimi literami — Access radzi sobie zarówno z małymi, jak i z wielkimi literami.



Rysunek 2.44. Okno wprowadzania parametrów kwerendy

Ćwiczenie 2.24. W kwerendach można stosować wiele parametrów. Przeprojektuj powstałą w ćwiczeniu 2.23 kwerendę tak, aby wybierała studentów z danego kierunku oraz mieszkających w danym mieście (określonym parametrem). Zauważ, że kolejność podawania parametrów jest zgodna z kolejnością pól w siatce projektowej. Dodaj drugie miasto do kryterium wyboru, jak na górze rysunku 2.45. Uruchom zapytanie i wybierz studentów informatyki mieszkających w Szczecinie lub Toruniu. Utworzoną kwerendę zapisz pod nazwą *Studenci_wg_kierunku_z_wieloma_parametrami*.

Pole:	ID_albumu	KOD_kierunku	Miasto
Tabela:	Studenci_dane	Studenci_uczelnia	Studenci_dane
Sortuj:			
Pokaż:	☑	☑	☑
Kryteria:		[podaj KOD kierunku (BI,CH,FI,IN,MA)]	[podaj 1. miasto zamieszkania] Or [podaj 2. miasto zamieszkania]
lub:			

Pole:	ID_albumu	KOD_kierunku	Miasto
Tabela:	Studenci_dane	Studenci_uczelnia	Studenci_dane
Sortuj:			
Pokaż:	☑	☑	☑
Kryteria:		[podaj KOD kierunku (BI,CH,FI,IN,MA)]	[podaj 1. miasto zamieszkania]
lub:			[podaj 2. miasto zamieszkania]

Rysunek 2.45. Kwerenda z wieloma parametrami

Zwróć uwagę, że na rysunku 2.45 użyta została alternatywa dwóch warunków parametrycznych. Zastanów się, dlaczego dolne rozwiązanie jest niepoprawne dla tego przykładu. Jakie rekordy otrzymasz w wyniku działania tej kwerendy, jeśli jako parametry podasz kolejno: `IN`, `Szczecin`, `Toruń`?

Ćwiczenie 2.25. Na podstawie kwerendy *Srednia_ocen* utwórz nową kwerendę parametryczną, która będzie wybierała studentów ze średnią ocen między dwiema wartościami podawanymi przy uruchamianiu zapytania w oknie z komunikatem *Podaj minimalną średnią* oraz *Podaj maksymalną średnią*. Utworzoną kwerendę zapisz pod nazwą *Srednia_ocen_z_parametrem*. Wskazówką niech w tym miejscu będzie zapis kryteriów. Podawane przy uruchomieniu oceny muszą zastąpić parametry, a zatem: `>=[Podaj minimalną średnią] And <=[Podaj maksymalną średnią]`.

2.4.2. Kwerendy funkcjonalne — modyfikujące

Oprócz kwerend wybierających można tworzyć kwerendy funkcjonalne. Lista kwerend, pokazana na rysunku 2.46, jest dostępna w widoku *projektu* kwerendy, w grupie narzędzi *Typ kwerendy*. Typy kwerend są dostępne również w menu kontekstowym obszaru tabel.



Rysunek 2.46. Typy kwerend

Przy ikonach czterech typów kwerend widoczny jest wykrzyknik, co oznacza, że wykonanie tych kwerend skutkuje trwałymi zmianami w bazie danych. Każdorazowe ich uruchomienie może powodować przeliczanie wartości lub usuwanie danych, jeśli spełnione zostaną określone kryteria. Trzeba więc zachować szczególną ostrożność. Należy również czytać komunikaty pojawiające się na ekranie przy próbie uruchomienia kwerend funkcjonalnych. Warto też wykonać kopię bazy lub przynajmniej tabel, aby nie stracić cennych danych. Dlatego przed przystąpieniem do dalszych ćwiczeń wykonaj kopię bazy *Uczelnia* i zapisz ją pod nazwą *Uczelnia_kopia*.

Kwerenda aktualizująca

Utwórz kwerendę opartą na tabelach lub kwerendach zawierających rekordy, które mają zostać zaktualizowane.

Aby wykonać zadanie, skorzystaj z tabeli *Studenci_dane*. Wykonaj jej kopię i nazwij ją *Studenci_dane_aktualizacja*. Ponieważ każde uruchomienie kwerendy aktualizującej oznacza przeliczenie wartości, trzeba być bardzo ostrożnym przy jej uruchamianiu. To właśnie dlatego kwerendy tego typu mają ikonę opatrzoną wykrzyknikiem symbolizującym niebezpieczeństwo. Zaktualizujemy dane w kopii tabeli, zwiększając stawki obecnych stypendiów socjalnych o 20%. Podwyżka dotyczy jedynie tych osób, których stypendium jest mniejsze niż 200 złotych.

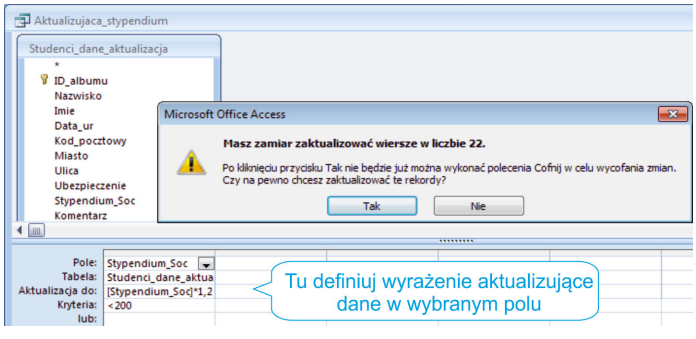
Utwórz nową kwerendę w widoku *projektu*. Wybierz tabelę *Studenci_dane_aktualizacja* i przeciągnij z listy pól tabeli do siatki projektu pole *Stypendium_Soc*, które będziesz

aktualizować lub dla którego chcesz ustawić kryteria. Wpisz `<200` w komórce *Kryteria*. Obecnie jest to kwerenda wybierająca, która wyświetli z tabeli *Studenci_dane_aktualizacja* wszystkie stypendia niższe niż 200 złotych.

Aby możliwe było przeliczenie tych kwot, zmień typ kwerendy na aktualizującą. W grupie narzędzi *Typ kwerendy* wybierz przycisk *Aktualizuj*. Typ kwerendy możesz również wybrać z menu kontekstowego obszaru tabel w projekcie kwerendy.

Po zmianie typu kwerendy w siatce projektowej została dodana komórka *Aktualizacja do:*. Można do niej wpisać wyrażenie lub wartość, która ma być wykorzystana do zmiany zawartości pola. Wpiszemy tu wyrażenie, które zwiększy wartość obecnego stypendium o 20%. Wpisz `[Stypendium_Soc]*1,2`. Zapisz kwerendę pod nazwą *Aktualizujaca_stypendium*. Zauważ, że kwerenda wyróżnia się inną ikoną: Aktualizujaca_stypendium. Wykrzyknik w ikonie ma nie tylko odróżniać tę kwerendę od kwerend wybierających, ale i ostrzegać przed przypadkowym lub nieprzemyślanym jej użyciem.

Uwaga! Każdorazowe uruchomienie tej kwerendy spowoduje kolejne przeliczenie kwoty stypendiów spełniających kryterium. Dlatego uruchomienie kwerendy aktualizującej jest poprzedzone ostrzeżeniem i zapytaniem, czy na pewno chcesz zaktualizować dane, co ilustruje rysunek 2.47. Po wykonaniu kwerendy sprawdź wartości stypendiów w zaktualizowanej tabeli.



Rysunek 2.47. Kwerenda aktualizująca dane

Wskazówka

Użycie kwerendy aktualizującej jest niezmiernie wygodne i pomocne w przeliczaniu istniejących lub podawaniu nowych wartości. Jeśli jednak nie zamierzamy w przyszłości użyć takiej kwerendy, po jej zastosowaniu można — a nawet należy — po prostu ją usunąć.

Kwerenda tworząca tabelę

Utwórz kwerendę, wybierając tabele lub kwerendy zawierające te rekordy, które mają się znaleźć w nowej tabeli.

Rozwiązanie zadania tego typu przebiega w dwóch etapach. Etap pierwszy to utworzenie kwerendy wybierającej, a następnie przekształcenie jej w *kwerendę tworzącą nową tabelę*.

Po podaniu nazwy powstałej tabeli w zakładce z tabelami pojawi się nowy zestaw danych — nowa tabela. Do ćwiczenia użyjemy więc już zdefiniowanej wcześniej kwerendy.

Ćwiczenie 2.26. Wykorzystaj kwerendę *Oceny_informatykow* do utworzenia nowej tabeli, która będzie zawierać dane z ocenami studentów informatyki.

W zakładce zawierającej kwerendy odszukaj kwerendę *Oceny_informatykow* i wykonaj jej kopię, a następnie zmień nazwę kopii na *Oceny_informatykow_tworzaca_tabele*. Otwórz ją w widoku *projektu*. Zmień jej typ na kwerendę tworzącą nową tabelę, wybierając ten typ z listy typów kwerend, widocznej na rysunku 2.46. Podaj nazwę nowej tabeli, która powstanie po uruchomieniu kwerendy: *Oceny_informatykow_tabela_z_kwerendy*. Uruchom przekształconą kwerendę. Przeczytaj komunikat i potwierdź chęć utworzenia tabeli z 14 rekordami. Zamknij okno kwerendy, zapisując zmiany, i sprawdź w zakładce z tabelami, czy powstała nowa tabela i jakie zawiera rekordy.

Kwerenda dołączająca

W bazie danych *uczelnia_filia.accdb* znajduje się tabela *Wykladowcy-filia*. Zamieszczono tam dane wykładowców, którzy uczą języków obcych. Utworzymy kwerendę, która dołączy dane tych wykładowców do tabeli *Wykladowcy*. W tym celu należy zaimportować tabelę *Wykladowcy-filia*. Po zaimportowaniu obejrzyj jej zawartość. Zwróć uwagę, że tabela ta została utworzona zgodnie z projektem tabeli *Wykladowcy*. Aby dołączyć tabelę do istniejącej tabeli, użyj *kwerendy dołączającej*. Zanim przystąpisz do tworzenia kwerendy dołączającej, utwórz kopię tabeli *Wykladowcy* (*Wykladowcy_kopia*). Będzie potrzebna do dalszej analizy.

Kwerenda dołączająca **dodaje na koniec tabeli** rekordy pochodzące z innej tabeli (lub kilku tabel). Jest to kwerenda z grupy kwerend funkcjonalnych. Jej ikona również jest wyróżniona znakiem wykrzyknika. Każdorazowe uruchomienie kwerendy dołączającej będzie próbą ponownego dodania rekordów do tabeli.

Utwórz nową kwerendę w widoku *projektu*. Wybierz tabelę, którą chcesz dołączyć, czyli zaimportowaną tabelę *Wykladowcy-filia*. Wybierz gwiazdkę z listy pól tabeli, symbolizującą dołączanie wszystkich pól. Teraz zmień typ kwerendy z wybierającej na dołączającą. Wybierz z listy lub wpisz nazwę tabeli, do której mają być dołączone rekordy po uruchomieniu kwerendy, czyli *Wykladowcy*. Utworzoną kwerendę zapisz pod nazwą *Dolaczajaca_tabele_wykladowcy-filia*. Teraz uruchom kwerendę dołączającą. W odpowiedzi na komunikat o próbie dodania rekordów do tabeli potwierdź chęć ich dodania.

Na przygotowanej wcześniej tabeli *Wykladowcy_kopia* sprawdź, jak będzie się zachowywała kwerenda przy kilku uruchomieniach. Jednak wcześniej w widoku projektu kwerendy zmień tabelę *Wykladowcy* na *Wykladowcy_kopia* przez ponowne wybranie typu kwerendy dołączającej przyciskiem *Dołącz*. Uchroni to dane potrzebne do dalszych ćwiczeń. Czy istniejące już w tabeli dane będą nadpisywane, czy podjęta zostanie próba ponownego dodania tych samych wykładowców? Czy podczas wykonywania tego typu funkcjonalnej kwerendy następuje jakaś analiza danych, czy tylko mechaniczne dopi-

sanie danych na końcu tabeli? Czy reguła poprawności i właściwość indeksowania bez powtórzeń pozwolą na zdublowanie danych? Wyciągnij wnioski.

Wykonaj drugą część tego ćwiczenia. W bazie *uczelnia_filia.accdb* znajduje się tabela *Konsultacje-filia*, w której zamieszczone zostały terminy konsultacji dołączonych wykładowców. Utwórz kwerendę dołączającą te dane do tabeli *Konsultacje*. Kwerendę zapisz pod nazwą *Dolaczajaca_konsultacje_filia*. Dodaj konsultacje pracowników z filii. Jeśli wcześniej nie zostały dodane dane wykładowców, nastąpi odmowa dodania konsultacji tych pracowników. Dlaczego? Wyjaśnienie ma związek z projektowaniem relacji między tabelami. W przypadku tabel połączonych relacją *jeden-do-jednego* najpierw należy wpisać dane do tabeli nadrzędnej (*Wykladowcy*), dopiero później można wpisać dane do tabeli podrzędnej (*Konsultacje*).

Kwerenda usuwająca

Jeśli rekordy zostaną usunięte za pomocą kwerendy usuwającej, która usuwa z jednej tabeli lub z kilku tabel wiersze spełniające określone kryteria, czynności tej nie można cofnąć. Dlatego też przed uruchomieniem kwerendy należy obejrzeć dane, które zostały przeznaczone do usunięcia. Aby to uczynić, trzeba kliknąć przycisk *Widok* na pasku narzędzi i obejrzeć kwerendę w widoku *arkusza danych*.

Wskazówka

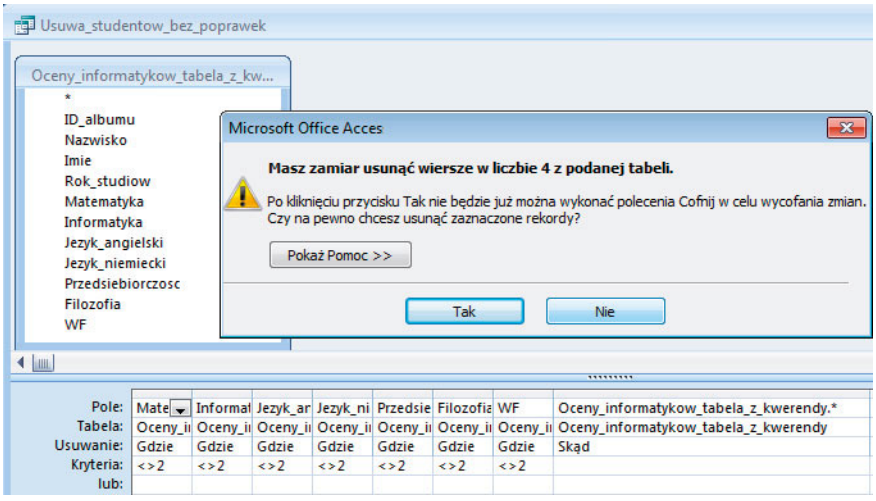
Zawsze należy sporządzać zapasowe kopie danych. Jeśli zostaną usunięte niewłaściwe rekordy, można je odzyskać z kopii zapasowej.

Ćwiczenie 2.27. Z tabeli utworzonej w ćwiczeniu 2.26 usuń dane tych studentów, którzy nie mają żadnej poprawki, czyli otrzymali ze wszystkich przedmiotów ocenę różną od niedostatecznej (2).

Praca ponownie będzie dwuetapowa: najpierw utworzymy kwerendę wybierającą, a następnie przekształcimy ją w funkcjonalną. Utwórz zatem nową kwerendę w widoku *projektu*. Wybierz tabelę *Oceny_informatykow_tabela_z_kwerendy*. Dodaj nazwy wszystkich przedmiotów do wiersza *Pole* w siatce projektu i określ kryteria wyboru (usuwania), wpisując <>2 dla każdego przedmiotu. Uruchom kwerendę wybierającą wszystkie te rekordy, które za chwilę zostaną usunięte (trwale) z tabeli. Powinny pokazać się tylko oceny pozytywne czterech studentów, którzy za chwilę zostaną usunięci.

Powróć do widoku *projektu* i w grupie *Typ kwerendy* wybierz przycisk *Usuń*. W siatce projektu kwerendy zostanie dodany wiersz *Usuwanie* i automatycznie pojawi się słowo *Gdzie*. Przeciągnij gwiazdkę (*) z listy pól tabeli do siatki projektu kwerendy. W komórce *Usuwanie* pojawi się słowo *Skąd*. Projekt kwerendy pokazano na rysunku 2.48. Kwerendę zapisz pod nazwą *Usuwa_studentow_bez_poprawek*.

Kliknij przycisk *Uruchom*, aby usunąć wybrane rekordy. Potwierdź jeszcze w oknie komunikatu chęć usunięcia czterech rekordów z tabeli, jak na rysunku 2.48. Po usunięciu danych sprawdź zawartość tabeli *Oceny_informatykow_tabela_z_kwerendy*. Wszyscy studenci mają oceny niedostateczne. Zmień nazwę tabeli na *Poprawki_informatykow*.



Rysunek 2.48. Kwerenda usuwająca dane z tabeli

W niektórych przypadkach uruchomienie kwerendy usuwającej może spowodować usunięcie rekordów z tabel związanych relacją, nawet jeśli nie zostały one uwzględnione w kwerendzie. Może się tak stać, jeśli tabela znajdująca się w kwerendzie jest stroną „jeden" relacji *jeden-do-wielu* i dla tej relacji została włączona opcja usuwania kaskadowego. Wówczas, usuwając rekordy w tabeli po stronie „jeden" relacji, usuwa się również rekordy z tabeli po stronie „wiele".

Kwerenda krzyżowa

Oblicz sumę, zlicz lub wykonaj inne działania, a następnie pogrupuj wyniki według dwóch typów informacji zawartych w tabeli (jak na rysunku 2.49), tworząc krzyżowe zestawienie podsumowanych lub zliczonych danych.



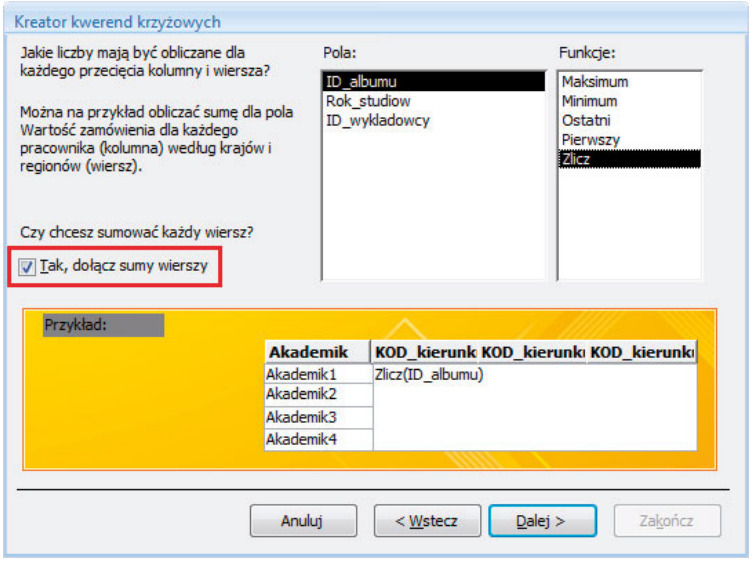
Rysunek 2.49. Kwerenda krzyżowa

W tym przykładzie utworzona została kwerenda zliczająca studentów z danych kierunków mieszkających w poszczególnych akademikach — domach studentów.

Aby utworzyć kwerendę krzyżową, posłużymy się narzędziem *Kreator kwerend krzyżowych*, które wybierzemy z zestawu kreatorów po naciśnięciu przycisku *Kreator kwerend* z grupy *Inne*.

- *W pierwszym kroku* kreatora wybierz tabelę, na podstawie której wykonasz zestawienie krzyżowe danych. W naszym przykładzie jest to tabela *Studenci_uczelnia*.
- *W drugim kroku* kreatora wskaż, które wartości mają być nagłówkami wierszy. W naszym przykładzie będą to symbole akademików, wybierz więc pole *Akademik*.

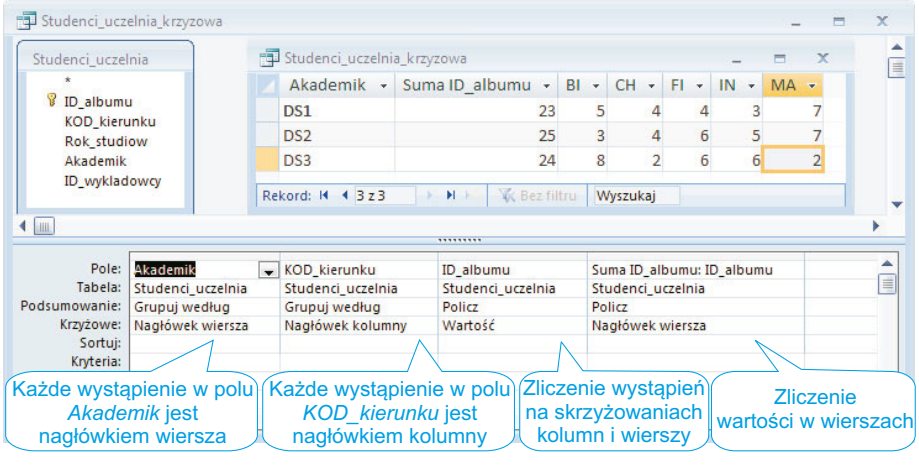
- *W trzecim kroku* kreatora wskaż, które wartości mają być nagłówkami kolumn. W naszym przykładzie będą to symbole kierunków studiów, wybierz więc pole *KOD_kierunku*.
- W kolejnym kroku ustal, jakie liczby mają być obliczane dla każdego przecięcia kolumny i wiersza. Dokonaj wyboru pola oraz funkcji. Tu możesz zaznaczyć opcję zliczania wszystkich wartości w każdym wierszu. Dla naszego przykładu ustal opcje zgodnie z rysunkiem 2.50.



Rysunek 2.50. Kreator kwerend krzyżowych: wybór pola i funkcji

- W ostatnim kroku zatwierdź lub podaj nazwę utworzonej kwerendy *Studenci_uczelnia_krzyzowa* i zakończ pracę kreatora.

Jak wygląda utworzona kwerenda w widoku *projektu*? Oba widoki, zarówno arkusza danych, jak i projektu, pokazano na rysunku 2.51, gdzie zostały zamieszczone wyjaśnienia.



Rysunek 2.51. Kwerenda krzyżowa w widoku arkusza danych i projektu

Z kwerendy krzyżowej korzysta się między innymi przy obliczaniu podsumowań, wartości średnich, zliczaniu wystąpień danych, wybieraniu wartości najmniejszych, największych, pierwszych lub ostatnich. Tak tworzone kombinacje danych, w których jedne dane stanowią nagłówki wierszy, a inne nagłówki kolumn, nie są możliwe w macierzystej tabeli.

Wykonaj ten projekt ponownie, zamieniając miejscami kolumny z wierszami. Powstałą kwerendę zapisz pod nazwą *Studenci_uczelnia_krzyzowa_odwrotnie*.

Ćwiczenie 2.28. Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która **na podstawie kwerendy** *Srednia_ocen* utworzy zestawienie średnich ocen na wszystkich kierunkach i rocznikach studiów. Zwróci też w kolumnie *Suma Srednia_ocen* najwyższą średnią w każdym roczniku, tak jak to pokazano na rysunku 2.52. Kwerendę zapisz pod nazwą *Srednia_ocen_krzyzowa*.

Srednia_ocen_krzyzowa

Rok_studiow	Suma Srednia_ocen	BI	CH	FI	IN	MA
1	4,83	4,17	3,83	4,83	4,83	4,33
2	4,50	4,00	4,33	4,50	4,50	4,33
3	4,33	4,33	4,00	4,17	4,17	4,17

Rekord: 1 z 3 | Bez filtru | Wyszukaj

Rysunek 2.52. Wynik działania kwerendy z ćwiczenia 2.28

Na rysunku przedstawiono wartości liczbowe (standardowe) z dwoma miejscami dziesiętnymi. Właściwości należy ustawiać w widoku *projektu* zarówno dla zliczania wystąpień na skrzyżowaniach wierszy i kolumn, jak i dla podsumowań wierszy. Zwróć uwagę, że pomimo wyboru funkcji *Maksimum* w tabeli wyświetlana jest kolumna *Suma Srednia_ocen*.

Ćwiczenie 2.29. Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która na podstawie tabeli *Studenci_uczelnia* utworzy zestawienie liczby osób studiujących na danym kierunku, z rozbiciem na poszczególne roczniki. Pokaże też sumę studentów na każdym kierunku, tak jak to zaprezentowano na rysunku 2.53. Kwerendę zapisz pod nazwą *Studenci_uczelnia_kierunek_rok_krzyzowa*.

Studenci_uczelnia_kierunek_rok_krzyzowa

KOD_kierun	Suma ID_albumu	1	2	3
BI	16	7	4	5
CH	10	3	5	2
FI	16	5	6	5
IN	14	3	5	6
MA	16	5	6	5

Rekord: 1 z 5 | Bez filtru | Wyszukaj

Rysunek 2.53. Wynik działania kwerendy z ćwiczenia 2.29

Ćwiczenie 2.30. Korzystając z kreatora kwerend krzyżowych, zaprojektuj kwerendę, która na podstawie kwerendy *Stypendium_i_kierunek* utworzy zestawienie ilościowe studentów każdego kierunku, rozróżniając wszystkie możliwe kwoty wypłacanego stypendium. Zlicz liczbę studentów z wszystkich kierunków razem dla każdej kwoty, tak jak to pokazano na rysunku 2.54. Powstałą kwerendę zapisz pod nazwą *Stypendium_i_kierunek_krzyzowa*.



Rysunek 2.54. Wynik działania kwerendy z ćwiczenia 2.30

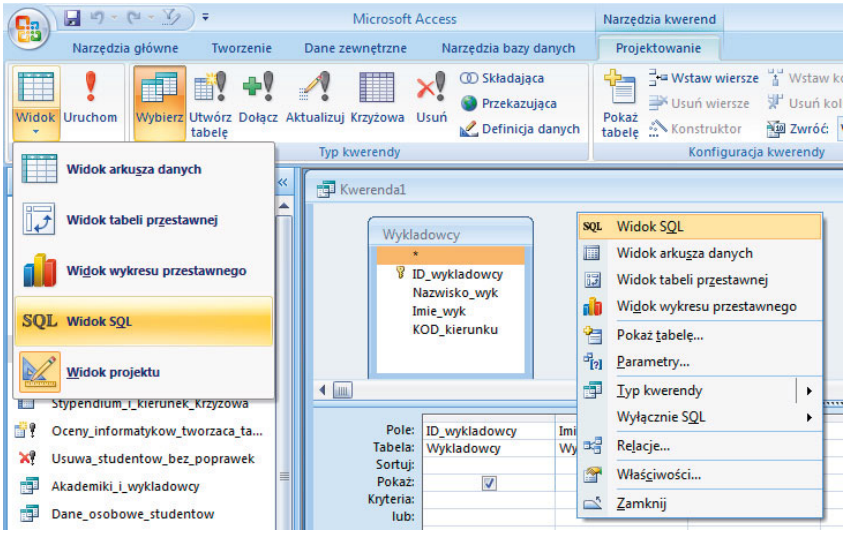
2.4.3. Język zapytań SQL

Zapytania w bazach danych zazwyczaj tworzy się za pomocą języka SQL (jego dialektów). Jednak w Accessie dominującą metodą ułatwiającą tworzenie kwerend jest korzystanie z szablonu graficznego, w którym zapytanie tworzone jest za pomocą graficznych elementów, przy użyciu myszki, jak to robiliśmy we wszystkich dotychczasowych ćwiczeniach. Należy jednak pamiętać, że jest to tylko forma prezentacji. Zapytanie istnieje w postaci kodu SQL, a niektóre zapytania można stworzyć tylko przy użyciu języka SQL — szablon graficzny jest wtedy nieprzydatny. Każde zapytanie można zapisać bezpośrednio przy użyciu tego języka.

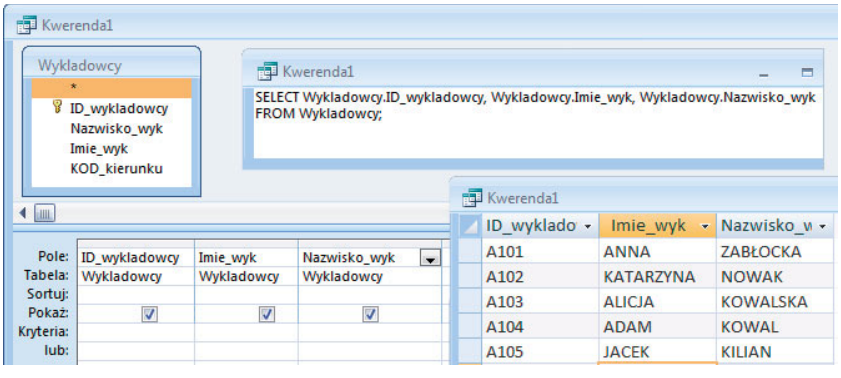
Kwerenda oparta na jednej tabeli

Ponieważ każda kwerenda utworzona przy użyciu graficznego szablonu ma swój zapis w **kodzie SQL**, zobaczmy, jak wygląda taki kod dla kwerendy wybierającej dane z tabeli *Wykladowcy*. W widoku *projektu* wybierz z niej pola z ID oraz imionami i nazwiskami wszystkich wykładowców. Teraz zobacz jej kod w języku zapytań. W tym celu kliknij prawym przyciskiem myszy w wolnym obszarze tabel kwerendy. W oknie dialogowym wybierz opcję *Widok SQL*. Ten sam efekt otrzymasz, wybierając *Widok SQL* z listy rozwijanej przycisku *Widok* zamieszczonego w grupie *Widoki*, co ilustruje rysunek 2.55.

Po wywołaniu widoku SQL pojawi się okno z zapisem kwerendy w języku SQL. Treść kodu w tym oknie możesz czytać i modyfikować. Wszystkie trzy widoki tej kwerendy wybierającej zgromadzone zostały na rysunku 2.56.



Rysunek 2.55. Sposoby przejścia z widoku projektu kwerendy do widoku SQL



Rysunek 2.56. Kwerenda wybierająca w trzech widokach: projektu, arkusza danych i kodu SQL

SQL to język deklaratywny, bardzo wysokiego poziomu, jednak zrozumienie podstaw składni SQL nie będzie stanowić dla Ciebie wielkiego problemu. Omawiana prosta kwerenda wybierająca ma następującą postać:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Imie_wyk, Wykladowcy.Nazwisko_wyk
FROM Wykladowcy;
```

Język SQL jest językiem strukturalnym. Przeanalizujmy treść powyższego kodu.

`SELECT` jest instrukcją służącą do wybierania rekordów i pól z tabeli lub wielu tabel. Klauzula `FROM` wyznacza źródłowe tabele. Podstawowa składnia instrukcji `SELECT` wygląda następująco:

```
SELECT pola FROM tabela;
```

Nazwę pola poprzedzamy nazwą tabeli i kropką. Listę pól oddzielamy przecinkami. Pola zostaną wyświetlone w kwerendzie w takiej kolejności, w jakiej podamy je w zapisie SQL. Jeśli chcesz wyświetlić wszystkie pola, w miejscu *pola* wpisz *. Zapis *tabela* ozna-

cza nazwę pojedynczej tabeli lub określa zbiór połączonych tabel (traktowanych dzięki relacjom jak jedna tabela).

W oknie widoku SQL dokonaj zmiany, dopisując jeszcze pole z kodem kierunku. Kwerenda będzie więc miała postać:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Imie_wyk, Wykladowcy.
Nazwisko_wyk, Wykladowcy.KOD_kierunku
FROM Wykladowcy;
```

Równoważny powyższemu jest zapis:

```
SELECT Wykladowcy.*
FROM Wykladowcy;
```

Zamiast wymieniać wszystkie pola z tabeli *Wykladowcy*, użyto znaku *.

Kwerenda wybierająca dane z kilku tabel

Zobaczmy, jak wygląda kod SQL dla kwerendy wybierającej dane z dwóch tabel, na przykład *Studenci_dane* i *Studenci_uczelnia*. W widoku *projektu* wybierzmy z nich pola z ID, imieniem i nazwiskiem oraz kierunkiem studiów. Interesuje nas zapis połączenia tabel relacją. W języku SQL wygląda on następująco:

```
SELECT Studenci_dane.ID_albumu, Studenci_dane.Nazwisko, Studenci_dane.Imie,
Studenci_uczelnia.KOD_kierunku
FROM Studenci_dane INNER JOIN Studenci_uczelnia ON Studenci_dane.ID_albumu
= Studenci_uczelnia.ID_albumu;
```

Instrukcja `SELECT` jest już dla nas czytelna i jasna. W klauzuli `FROM` zastosowano operator złączenia w celu pobrania danych z dwóch tabel. Do najczęściej stosowanych operatorów należą:

`INNER JOIN` — złączenie wewnętrzne, wynikające z ustanowionych relacji; w wyniku znajdą się tylko odpowiadające sobie rekordy z obu tabel.

`CROSS JOIN` — złączenie krzyżowe; wynikiem jest zestawienie kombinacji wszystkich rekordów z obu tabel (tak jak to miało miejsce w przypadku wyboru danych z tabel, między którymi nie ustanowiono relacji).

Klauzulę `FROM` z naszego przykładu można zatem tłumaczyć następująco: wybierz rekordy występujące w obu tabelach (*FROM_Studenci_dane INNER JOIN Studenci_uczelnia*), których *ID_albumu* są sobie równe (te pola są bowiem połączone relacją). Jednak możliwe jest również następujące polecenie: wybierz wszystkie rekordy występujące w tabeli *Studenci_dane* i tylko te rekordy z tabeli *Studenci_uczelnia* (*FROM Studenci_dane LEFT INNER JOIN Studenci_uczelnia*), których *ID_albumu* są sobie równe. I analogiczne polecenie: wybierz wszystkie rekordy występujące w tabeli *Studenci_uczelnia* i tylko te rekordy z tabeli *Studenci_dane* (*FROM Studenci_dane RIGHT INNER JOIN Studenci_uczelnia*), których *ID_albumu* są sobie równe. Tego typu sprzężenia nazywamy *sprzężeniem lewostronnym* lub *prawostronnym*. W widoku projektu kwerendy zmianę typu sprzężenia można wykonać w oknie *Właściwości sprzężenia*, które jest dostępne po dwukrotnym kliknięciu na linii symbolizującej sprzężenie tabel.

Sortowanie

Powróćmy do widoku *projektu* i dodajmy opcję sortowania danych według identyfikatora wykładowcy. Kod SQL jest teraz następujący:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Imie_wyk, Wykladowcy.
Nazwisko_wyk, Wykladowcy.KOD_kierunku
FROM Wykladowcy
ORDER BY Wykladowcy.ID_wykladowcy;
```

Gdy za klauzulą `ORDER BY` (sortuj według) umieścimy zapis `DESC` (ang. *descending* — malejąco), posortujemy nazwiska malejąco:

```
ORDER BY Wykladowcy.ID_wykladowcy DESC;
```

Analogicznie `ASC` (ang. *ascending* — rosnąco) posortuje dane w porządku rosnącym. Brak na końcu klauzuli `ORDER BY` jednej z opcji `ASC/DESC` spowoduje domyślne posortowanie w porządku rosnącym, jak to miało miejsce w naszym przykładzie.

Kryterium wyboru

Kolejnym etapem będzie wprowadzenie kryterium wyboru. Wybierzmy więc z naszej bazy tylko wykładowców informatyki. W oknie projektu dodamy do kwerendy kryterium wyboru `="IN"`. Kod SQL ma teraz postać:

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Imie_wyk, Wykladowcy.
Nazwisko_wyk, Wykladowcy.KOD_kierunku
FROM Wykladowcy
WHERE (((Wykladowcy.KOD_kierunku)="IN"))
ORDER BY Wykladowcy.ID_wykladowcy DESC;
```

W celu wprowadzenia ograniczeń kryteriów dla wyświetlanych rekordów zastosowana została klauzula `WHERE`. Za pomocą klauzuli `WHERE` można filtrować rekordy, czyli decydować o tym, które rekordy zostaną włączone do wyniku zapytania. Po słowie `WHERE` podaje się wyrażenie, które musi być prawdziwe, aby rekord pojawił się w wyniku zapytania (został wybrany). Istotne jest zachowanie kolejności, ponieważ klauzula `WHERE` wybiera rekordy z tabel wymienionych wcześniej w klauzuli `FROM`, które spełniają podane kryteria. Bez użycia klauzuli `WHERE` zwracane byłyby wszystkie rekordy tabeli (wszyscy wykładowcy, wszyscy studenci), tak jak w poprzednich przykładach. Klauzula `WHERE` jest więc konieczna, gdy chcemy wprowadzić kryteria wyboru. Składnia kwerendy z kryterium wyboru jest następująca:

```
SELECT pola FROM tabela WHERE wyrażenie;
```

Wyrażenie (predykat) zastosowane po słowie `WHERE` musi być wyrażeniem logicznym. Predykat może być pojedynczym kryterium, alternatywą lub koniunkcją kilku kryteriów wyboru. Najczęściej predykatem jest po prostu kryterium wyboru pól rekordów. Gdy jest ono prawdziwe (spełnione), powoduje wyświetlenie rekordów spełniających to kryterium. Można więc przedstawić składnię kwerendy z określonym kryterium w taki sposób:

```
SELECT pola FROM tabela WHERE predykat;
```

Oto kilka przykładów zastosowania kryteriów w widoku SQL:

- Alternatywa kryteriów: wypisz wykładowców, którzy w polu *KOD_kierunku* mają wpis równy *IN* lub *MA*.

```
WHERE (((Wykladowcy.KOD_kierunku)="IN" Or (Wykladowcy.KOD_kierunku)="MA"));
```

- Koniunkcja kryteriów: wypisz ID studentów trzeciego roku informatyki.

```
SELECT Studenci_uczelnia.ID_albumu
FROM Studenci_uczelnia
WHERE (((Studenci_uczelnia.Rok_studiow)=3) AND ((Studenci_uczelnia.KOD_
kierunku)="IN"));
```

- Alternatywa i koniunkcja: wypisz ID studentów trzeciego roku informatyki i matematyki.

```
WHERE (((Studenci_uczelnia.Rok_studiow)=3) AND ((Studenci_uczelnia.KOD_
kierunku)="IN" Or (Studenci_uczelnia.KOD_kierunku)="MA"));
```

- Kryterium ograniczone podanymi wartościami:

```
WHERE (((Studenci_uczelnia.Srednia_ocen)>=3 AND (Studenci_uczelnia.
Srednia_ocen)<=4));
```

- Krótszy zapis powyższej koniunkcji z podaniem zakresów:

```
WHERE (((Studenci_uczelnia.Srednia_ocen) Between 3 AND 4));
```

Pola obliczeniowe

Na przykładzie kwerendy obliczającej nową stawkę stypendium socjalnego przeanalizujmy zasadę tworzenia nowego pola, którego wartości są wynikiem obliczeń. Do kwoty stypendium dodajemy każdemu 10 złotych — oczywiście pod warunkiem, że student stypendium otrzymuje. Oto zapis kwerendy:

```
SELECT Studenci_dane.ID_albumu, Studenci_dane.Stypendium_Soc, [Stypendium_
Soc]+10 AS Nowa_stawka
FROM Studenci_dane
WHERE (((Studenci_dane.Stypendium_Soc) Is Not Null));
```

W wykazie pól pojawiło się nowe pole, powstałe z wyrażenia, które do pola *Stypendium_Soc* dodaje wartość `10`. `AS` oznacza, że w widoku *arkusza danych* wyświetlona zostanie kolumna (pole) o podanej nazwie *Nowa_stawka*. Zwróć również uwagę na wprowadzone kryterium `WHERE … Is Not Null`.

Kwerendy parametryczne

Parametry podajemy w kryteriach, więc nie będzie dla nikogo zaskoczeniem, że w kodzie SQL parametr z treścią komunikatu *podaj kod kierunku* pojawi się w nawiasach kwadratowych w klauzuli `WHERE`.

```
SELECT Wykladowcy.ID_wykladowcy, Wykladowcy.Nazwisko_wyk, Wykladowcy.KOD_
kierunku
FROM Wykladowcy
WHERE (((Wykladowcy.KOD_kierunku)=[podaj kod kierunku]));
```

Jeśli jednak chcesz użyć kilku parametrów, dla każdego z nich, nawet gdy dotyczy tego samego pola, musisz podać nazwę tabeli, nazwę pola i — opcjonalnie — treść komunikatu. A zatem rozbudowując poprzedni przykład tak, aby podać dwie wartości parametrów przy uruchamianiu kwerendy, musisz napisać:

```
WHERE (((Wykladowcy.KOD_kierunku)=[podaj kod kierunku] Or (Wykladowcy.
KOD_kierunku)=[podaj kod drugiego kierunku]));
```

Kwerenda ta wypisze dane wykładowców z dwóch kierunków, przy czym konkretne kierunki podaje użytkownik (przy uruchomieniu kwerendy).

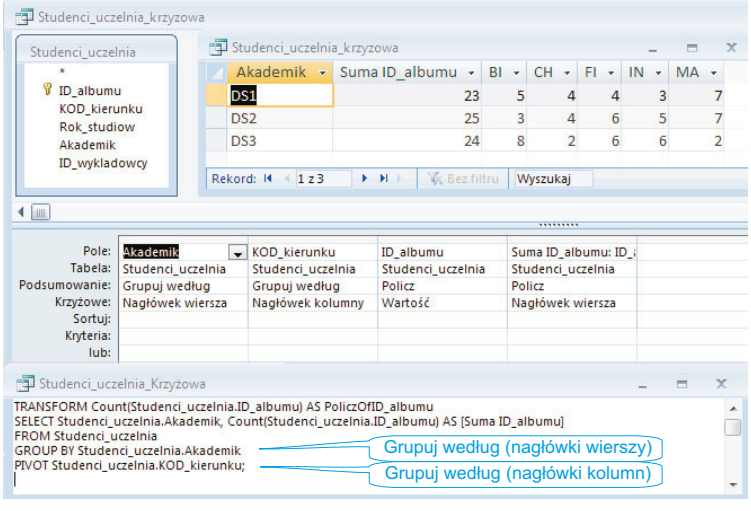
Kwerenda tworząca nową tabelę

Kod SQL kwerendy tworzącej tabelę różni się od kodu kwerendy wybierającej obecnością klauzuli `INTO`, która informuje, jak ma się nazywać nowa tabela, powstała po uruchomieniu kwerendy. Zapis kodu kwerendy tworzącej z tabeli *Studenci_uczelnia* nową tabelę, która zawiera dwie wybrane kolumny danych, wygląda zatem następująco:

```
SELECT Studenci_uczelnia.ID_albumu, Studenci_uczelnia.KOD_kierunku INTO
[Studenci_ID+KOD]
FROM Studenci_uczelnia;
```

Kwerenda krzyżowa

Jako przykład do analizy posłuży utworzona wcześniej kwerenda krzyżowa *Studenci_uczelnia_krzyzowa*. Została ona utworzona w oknie kreatora kwerend krzyżowych i samo przełączenie do widoku *projektu* może wywołać lekkie zaskoczenie. Na rysunku 2.57 zestawiono wszystkie trzy widoki tej skomplikowanej kwerendy: widok *projektu*, *arkusza danych* i *SQL*.



Rysunek 2.57. Kwerenda krzyżowa w trzech widokach: projektu, arkusza danych i SQL

Przyjrzyjmy się uważniej kolejnym liniom kodu tej kwerendy krzyżowej.

```
TRANSFORM Count(Studenci_uczelnia.ID_albumu) AS PoliczOfID_albumu
```

Operacje zliczenia (`PoliczOf`) będą wykonywane na podstawie pola *ID_albumu* z tabeli *Studenci_uczelnia*.

```
SELECT Studenci_uczelnia.Akademik, Count(Studenci_uczelnia.ID_albumu) AS
[Suma ID_albumu]
```

W kolumnie, która pojawi się jako *Suma ID_albumu*, nastąpi zliczenie wszystkich wartości dla każdego wiersza *Akademik*. Dla każdego wystąpienia w polu *Akademik* nastąpi zliczenie identyfikatorów w polu *ID_albumu*.

```
FROM Studenci_uczelnia
```

To jest chyba jasne.

```
GROUP BY Studenci_uczelnia.Akademik
```

Grupuj według wystąpień w polu *Akademik*. Każde wystąpienie będzie nowym wierszem (*DS1*, *DS2*, *DS3*).

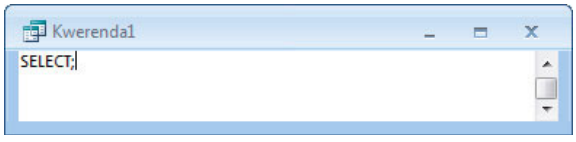
```
PIVOT Studenci_uczelnia.KOD_kierunku
```

Grupuj według wystąpień w polu *KOD_kierunku*. Każde wystąpienie będzie nową kolumną (*BI*, *CH*, *FI*, *IN*, *MA*).

Tworzenie kwerend za pomocą instrukcji języka SQL

Przejdźmy do tworzenia kwerend za pomocą instrukcji języka SQL.

W celu utworzenia kwerendy za pomocą instrukcji języka SQL należy w widoku *projektu* nowej kwerendy zamknąć pojawiające się standardowo okno *Pokazywanie tabel*, a następnie przejść do widoku SQL kwerendy. Zostanie otwarte okno, jak na rysunku 2.58, w którym zdefiniujemy kwerendę za pomocą instrukcji języka SQL. W oknie wpisane jest tylko słowo `SELECT` i średnik, który kończy instrukcję SQL.



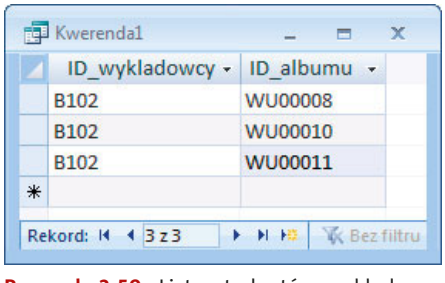
Rysunek 2.58. Okno tworzenia kwerendy w widoku kodu SQL

Po napisaniu kwerendy można ją uruchomić lub otworzyć w widoku *projektu*. Zmienimy więc kolejność tworzenia zapytania. Najpierw będziemy pisać kwerendy w języku SQL, a potem obserwować efekty w widoku *projektu* kwerendy.

Ćwiczenie 2.31. Utwórz zapytanie prezentujące na podstawie tabeli *Plan_zajec* wykaz studentów (ich ID) wybranego wykładowcy. Efekt działania kwerendy powinien być taki jak na rysunku 2.59, na którym wyświetlona jest lista studentów wykładowcy *B102*.

W pierwszej linii wpisz nazwy pól oddzielone przecinkiem. Pamiętaj o składni:

```
SELECT tabela.pole1, tabela.pole2
```



Rysunek 2.59. Lista studentów wykładowcy B102

Określ, skąd mają być pobierane dane:

```
FROM tabela
```

Ustal kryterium wyboru. W miejscu kryterium wpisz parametr:

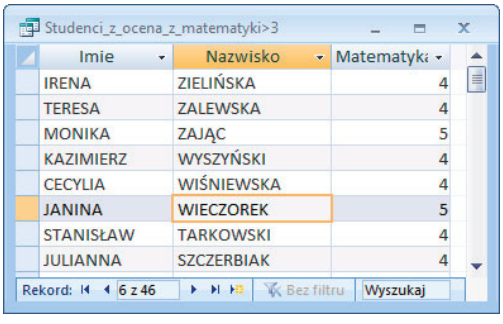
```
WHERE (((tabela.pole)=[komunikat dla parametru]));
```

Oto gotowa treść kwerendy zapisana w języku SQL. Porównaj ją ze swoim rozwiązaniem:

```
SELECT Plan_zajec.ID_wykladowcy, Plan_zajec.ID_albumu
FROM Plan_zajec
WHERE (((Plan_zajec.ID_wykladowcy)=[podaj ID wykladowcy]));
```

Kwerendę zapisz pod nazwą *Wykaz_studentow_na_zajecia_wykladowcy_z_parametrem*.

Ćwiczenie 2.32. Utwórz zapytanie wybierające studentów (z podaniem ich imion i nazwisk), którzy uzyskali z matematyki ocenę wyższą od 3. Wynik działania kwerendy *Studenci_z_ocena_z_matematyki>3* zilustrowano na rysunku 2.60.



Rysunek 2.60. Wykaz 46 rekordów spełniających kryterium z ćwiczenia 2.32

Jakie pola musisz wybrać do działania tej kwerendy? Podaj nazwy pól poprzedzone kropką i nazwą tabeli.

Określ, skąd mają być pobierane dane. Pamiętaj, że te dwie tabele są połączone relacją. Ogólna składnia jest następująca:

```
FROM tabela1 INNER JOIN tabela2 ON tabela1.pole = tabela2.pole
```

Ustal kryterium wyboru. W miejscu *kryterium* wpisz parametr:

```
WHERE (((tabela.pole)kryterium));
```

Wydaj polecenie posortowania danych według nazwisk w porządku malejącym. W miejscu *porzadek* wpisz odpowiednie polecenie:

```
ORDER BY tabela.pole porzadek;
```

Oto gotowa treść kwerendy zapisana w języku SQL. Porównaj ją ze swoim rozwiązaniem:

```
SELECT Studenci_dane.Imie, Studenci_dane.Nazwisko, Studenci_oceny.
Matematyka
FROM Studenci_dane INNER JOIN Studenci_oceny ON Studenci_dane.ID_albumu =
Studenci_oceny.ID_albumu
WHERE (((Studenci_oceny.Matematyka)>3));
ORDER BY Studenci_dane.Nazwisko DESC;
```

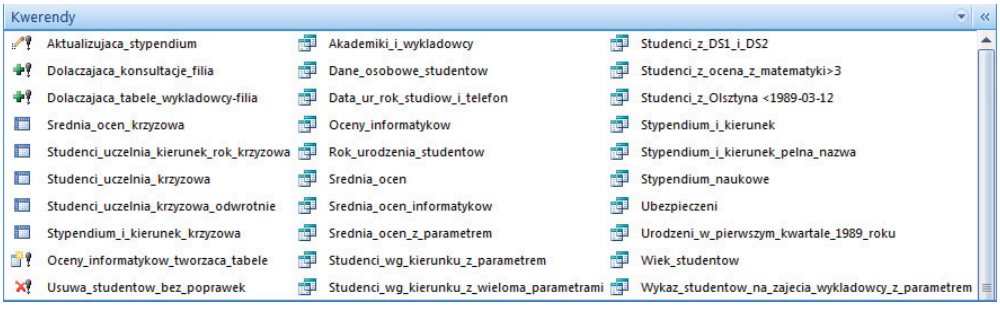
Zapisz utworzoną kwerendę.

Warunki zapisywane w podwójnych nawiasach (()) nie są warunkami języka SQL, lecz warunkami języka wyrażeń Office Access (ściślej — języka Visual Basic for Applications).

Wskazówka

Po wykonaniu wszystkich ćwiczeń z punktu *2.4*, *Kwerendy — zapytania*, w zakładce z kwerendami powinny pojawić się kwerendy pokazane na rysunku 2.61.

2.5. Formularze



Rysunek 2.61. Lista kwerend utworzonych w ramach ćwiczeń z punktu 2.4, Kwerendy — zapytania

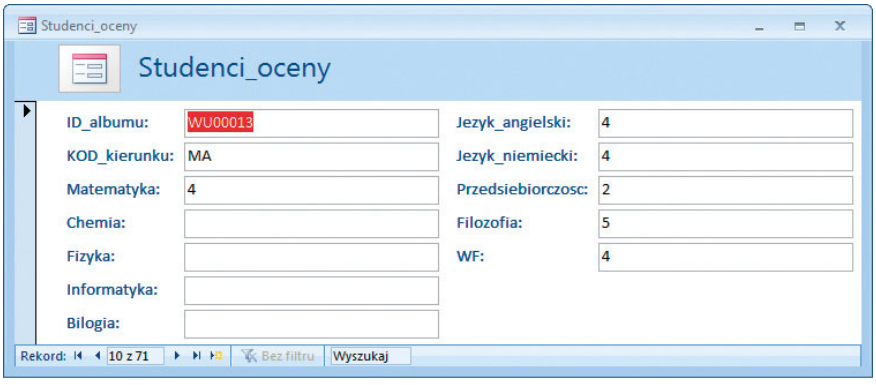
Formularze służą do graficznej prezentacji danych zawartych w tabeli lub kwerendzie oraz do wprowadzania danych do tabeli źródłowej poprzez graficzny interfejs użytkownika. Jeśli więc chcesz wyświetlać, wprowadzać lub zmieniać dane w tabelach, utwórz formularz. Głównymi składowymi formularza są obiekty graficzne, które są powiązane z danymi przechowywanymi w bazie danych.

Przy otwieraniu formularza program Microsoft Office Access 2007 pobiera dane z jednej tabeli lub z kilku tabel i wyświetla je na ekranie. Układ danych na ekranie możesz wybrać z kreatora formularzy lub zdefiniować samodzielnie, projektując formularz.

Przed przystąpieniem do ćwiczeń w oknie obiektów wyświetl *Formularze*. Obecnie lista formularzy jest pusta.

2.5.1. Autoformularze

Najprostszym sposobem opracowania formularza, który pozwoli przeglądać, dodawać, wyszukiwać, edytować i usuwać dane tabeli, jest utworzenie tzw. autoformularza. W tym celu wystarczy wskazać tabelę, dla której ma automatycznie powstać formularz, i z zakładki *Tworzenie*, z grupy *Formularze*, wybrać polecenie *Formularz*. Powróć więc do obiektów *Tabele* i utwórz formularz do tabeli *Studenci_oceny*. Widok powstałego automatycznie formularza pokazano na rysunku 2.62. Zapisz formularz pod nazwą *Studenci_oceny*. Zwróć uwagę na pasek stanu formularza, gdzie znajdują się: nawigacja między rekordami tabeli, wyszukiwanie oraz możliwość przejścia do okna dodawania nowego rekordu. Zauważ, że w widoku formularza możliwe jest filtrowanie danych.



Rysunek 2.62. Utworzony automatycznie formularz do tabeli Studenci_oceny

Oprócz standardowego formularza można również automatycznie utworzyć tzw. formularz dzielony. Łączy on dwa widoki w jednym oknie: widok tabeli i widok formularza. Utwórz więc do tabeli *Studenci_oceny* taki formularz i zapisz pod nazwą *Studenci_oceny_dzielony*. Ostatnim automatycznym formularzem jest formularz pokazujący *Wiele elementów*. Pozwala on oglądać wiele rekordów, przeznaczając dla każdego jeden wiersz. Można go porównać z widokiem przejrzystej tabeli. Utwórz również taki automatyczny formularz do tabeli *Studenci_oceny* i nazwij go *Studenci_oceny_wiele_elementow*.

Powróć do widoku *Formularzy*. Lista zapełnia się. Powinny tam już być trzy utworzone przed chwilą automatyczne formularze do tabeli *Studenci_oceny*. Możesz je wyświetlić.

2.5.2. Projektowanie formularzy

Ćwiczenie 2.33. Wygeneruj prosty formularz do czytania i wprowadzania danych w spisie kierunków. Skorzystaj z możliwości generowania formularzy za pomocą kreatora. Zapisz formularz pod nazwą *Kierunki*.

Aby utworzyć nowy formularz, przejdź do zakładki *Tworzenie*, gdzie zamieszczono grupę *Formularze*. Tam pod przyciskiem *Więcej formularzy…* ukryto *Kreator formularzy*. W pierwszym kroku kreatora należy wybrać z listy tabelę *Kierunki* będącą źródłem pól dla formularza i dodać oba pola znajdujące się w tej tabeli. W drugim kroku wybierz kolumnowy układ formularza. Zobacz ofertę stylów i wybierz standardowy *Access 2007*. Podaj nazwę dla utworzonego formularza i zakończ pracę kreatora. Nie było to trudne zadanie, prawda?

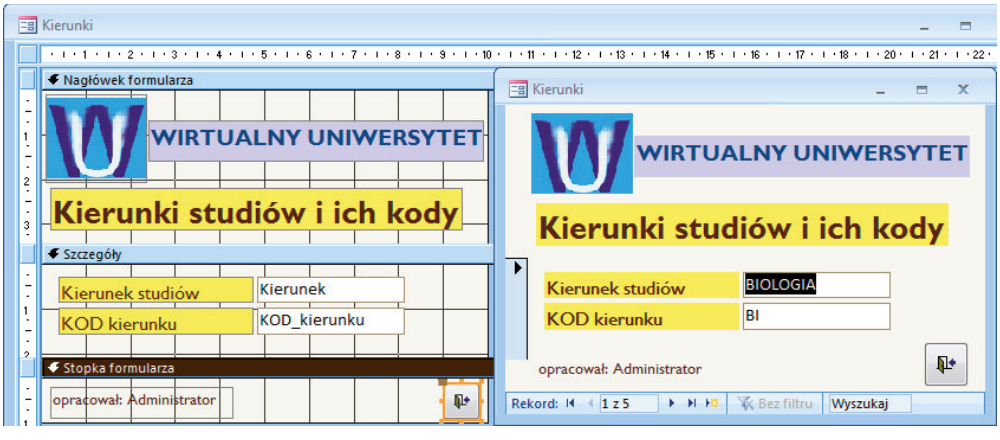
Podczas pracy z formularzem można przełączać pomiędzy trzema widokami: *formularza*, *układu* i *projektu*. Funkcja ta jest dostępna z poziomu paska narzędzi lub z menu podręcznego. Zobacz swój formularz w tych trzech widokach. Przejdź do widoku układu. Automatycznie zmieniły się *Narzędzia układów formularzy*. Dodany został pasek *Formatowanie* — można tu zmienić wygląd formularza. Zobacz gotowe projekty zgromadzone w grupie *Autoformatowanie*.

Zwróć uwagę na to, że otrzymany formularz jest czytelny, przejrzysty i prosty w nawigacji. Jego wygląd — szatę graficzną, kształty czy położenie elementów — zmienisz w widoku *projektu*. Możesz również zamieścić nowe pola tekstowe (etykiety), przyciski akcji czy **podformularze**.

Na formularzu można umieszczać **formanty**, czyli elementy prezentujące dane oraz pewne elementy sterujące. Wybiera się je z zakładki *Projektowanie*, na której zamieszczono grupę *Formanty* (w poprzednich wersjach pakietu Office formanty zgromadzone były w tzw. *przyborniku*, czasami nazywanym *piórnikiem*). Formanty mają różne przeznaczenie, a ich wstępne rozszyfrowanie nastąpi po zbliżeniu wskaźnika myszy do każdego z nich.

W widoku *projektu* również można dokonywać zmiany wyglądu elementów graficznych projektu formularza. Służą do tego narzędzia zgromadzone w oddzielnych grupach, ułatwiające między innymi dodawanie kolorów, efektów specjalnych, czcionek i linii.

W naszym prostym formularzu mamy jedynie dwa pola, których nazwy brzmią tak samo jak pola w tabeli *Kierunki*, co w formularzu nie wygląda najlepiej. Otwórzmy więc formularz w widoku *projektu* i zmieńmy kilka elementów oraz dodajmy nowe, tak jak w przykładzie przedstawionym na rysunku 2.63.

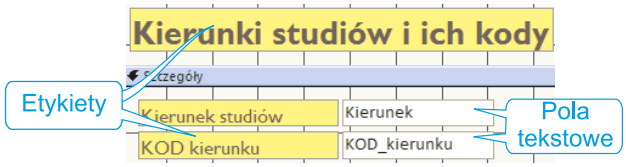


Rysunek 2.63. Propozycja zmiany interfejsu graficznego prostego formularza utworzonego w ćwiczeniu 2.33

Zauważ, że każdy formularz (oczywiście w widoku *formularza*) prezentuje wybrane dane na temat jednego rekordu. W tabelach widzieliśmy jednocześnie wiele rekordów.

Każdy formularz w widoku *projektu* podzielony jest na trzy sekcje: *nagłówek*, *szczegóły* i *stopkę*.

W każdej części formularza mogą wystąpić *formanty*. Formant to pojedynczy element używany do tworzenia formularza. W naszym przykładzie występują dwa główne rodzaje formantów: *pola tekstowe* i *etykiety*, których znaczenie zilustrowano na rysunku 2.64. Pole tekstowe służy do wyświetlania wartości pobieranej z pola tabeli lub kwerendy źródłowej. Etykieta zawiera tekst, na przykład: tytuł, podpis, komentarz.



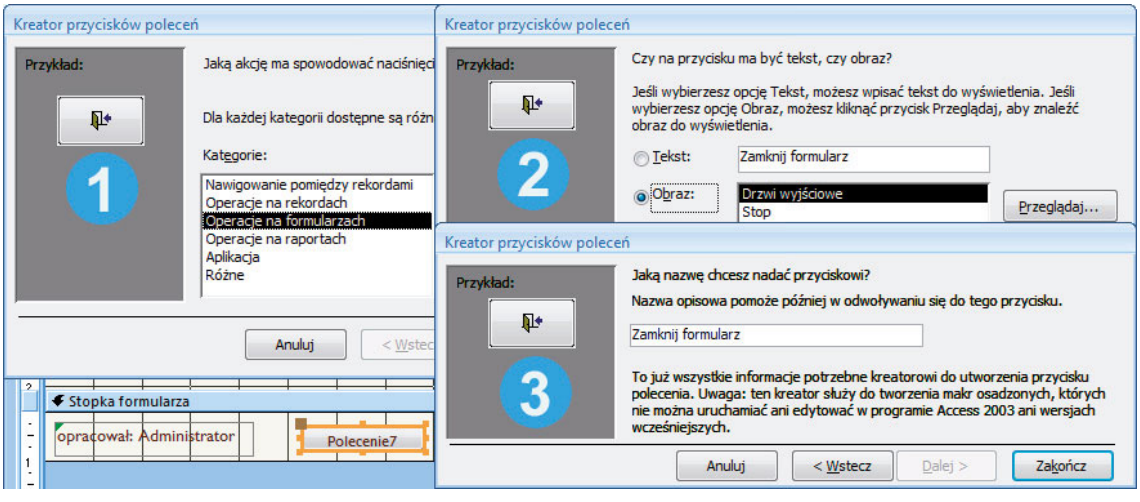
Rysunek 2.64. Przykłady formantów w formularzu

W nagłówku formularza przedstawiane są informacje, które pozostają niezmienne dla każdego rekordu, takie jak tytuł formularza. Aby przejść do sekcji nagłówka, należy powiększyć jego rozmiar. W tym celu trzeba chwycić krawędź między nagłówkiem a szczegółami i rozciągnąć obszar przeznaczony dla nagłówka. Można też dokładnie podać jego wysokość — wystarczy po wyborze z menu kontekstowego opcji *Właściwości* wpisać w oknie odpowiednie parametry.

W sekcji *Nagłówek formularza* dodaj dwie etykiety z tekstem. Po wpisaniu tekstu zaznacz pole etykiety i ustal wielkość czcionki, położenie tekstu i kolory wypełnienia. Dodaj obraz z pliku, na przykład logo *UW_logo.bmp*, które znajdziesz na płycie CD dołączonej do podręcznika (możesz również narysować własne logo). Przy wstawianiu obrazu zwróć uwagę na typy plików graficznych, które można wstawić do formularza Accessa.

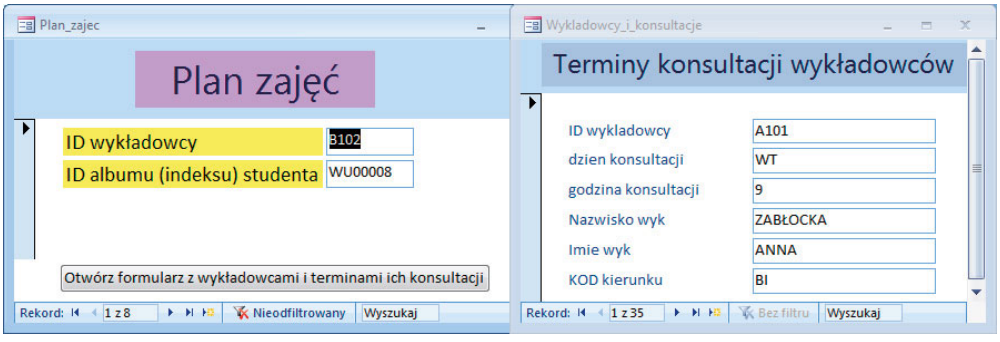
W sekcji *Szczegóły* zmień tekst etykiety pola oraz wygląd wszystkich zamieszczonych w tej sekcji formantów. Wszystkich zmian możesz dokonać na wskazanym obiekcie lub grupie obiektów. Możesz również przejść do *Właściwości* każdego formantu, wybierając odpowiednie polecenie z jego menu podręcznego, i tam wpisać parametry, które chcesz ustalić i zmienić.

W sekcji *Stopka formularza* dodaj etykietę z informacją o autorze oraz przycisk poleceń mający zamykać okno formularza. Wstawienie przycisku poleceń uruchomi *Kreator przycisków poleceń*. Rysunek 2.65 pokazuje kolejne kroki postępowania z kreatorem.



Rysunek 2.65. Dodanie przycisku poleceń w formularzu za pomocą przycisków kreatora poleceń

Ćwiczenie 2.34. W tym ćwiczeniu dodasz do prostego formularza, który ma tworzyć plan lekcji oraz umożliwić przeglądanie go, przycisk uruchamiający inny formularz, w którym podasz dane wykładowców wraz z ich ID i terminem konsultacji, jak to pokazano na rysunku 2.66.



Rysunek 2.66. Formularz uruchamiający (w razie potrzeby) inny formularz

W tym celu najpierw przygotuj formularz *Wykladowcy_i_konsultacje*, który z tabel *Wykladowcy* i *Konsultacje* wybierze i wyświetli dane jak na rysunku. Zmień nazwy etykiet. Aby zmienić kolejność wyświetlanych pól, należy zaznaczyć pole tekstowe wraz z etykietą (klikając myszką tak, jak zaznacza się wiersz w tabeli), a następnie aktywne pola przesunąć za pomocą strzałek sterujących ruchem kursora w górę lub w dół. Uporządkuj kolejność danych. Zapisz i zamknij formularz.

Następnie wygeneruj właściwy formularz *Plan_zajec* zawierający pola z tabeli *Plan_zajec*. W widoku *projektu* w sekcji *Stopka formularza* dodaj przycisk poleceń, któremu w kreatorze przycisków poleceń przypisz akcję: *Uruchom formularz*, a następnie wskaż nazwę formularza *Wykladowcy_i_ konsultacje*. W sekcji nagłówka dodaj etykietę *Plan zajęć* i ustal dla niej formatowanie. Tak utworzony formularz zapisz pod nazwą *Plan_zajec*. Otwiera on inny formularz, zawierający dane o wykładowcach i godzinach ich konsultacji, który w razie potrzeby można w każdej chwili uruchomić. Sprawdź działanie formularza i przycisku.

To ćwiczenie z pewnością przekonuje do stosowania czytelnych i przyjaznych formularzy, które w tym przypadku ułatwiają wpisywanie danych do planu, pozwalają na przeglądanie już istniejących wpisów i powodują zminimalizowanie ewentualnych błędów. Wpisywanie danych bezpośrednio do tabeli *Plan_zajec* było kłopotliwe i mało czytelne dla użytkownika. Trudnością było również przełączanie się pomiędzy tabelami w celu znalezienia wszystkich danych o wykładowcach. Teraz wywoływany w formularzu drugi formularz pokazuje wszystkie zestawione z dwóch tabel dane o wykładowcach. Dodawanie danych do tabeli *Plan_zajec* poprzez formularz *Plan_zajec* odbywa się zgodnie z regułami i zasadą indeksowania pól kluczowych, bez powtórzeń ustalonych wcześniej dla tej tabeli.

Formularze pozwalają ponadto na prezentowanie danych z tabel i kwerend na wykresach, sortowanie i filtrowanie danych oraz na stosowanie makropoleceń i pól obliczeniowych, jak to miało miejsce w kwerendach. Formularze można również drukować.

2.6. Raporty

Jeśli chcesz analizować swoje dane lub zaprezentować je w postaci drukowanej, utwórz raport. Możesz na przykład wydrukować raport grupujący dane i obliczający podsumowania. Projektując raport, masz wpływ na jego szatę graficzną, podobnie jak w przypadku formularza. W raporcie można grupować dane według określonego kryterium, sortować je oraz wykonywać na nich obliczenia.

W raportach możesz więc między innymi utworzyć etykiety adresowe, podsumowania, zaprezentować wyliczone sumy na wykresach. Możesz również pokazać wiele rekordów.

Podobnie jak w formularzu, w raportach używamy formantów, dzięki którym tworzymy łącze między raportem a jego źródłem danych.

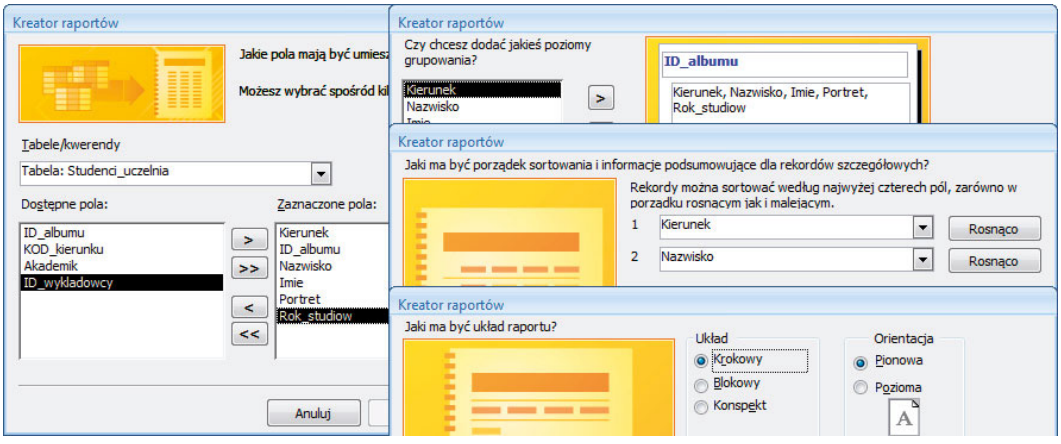
Ćwiczenie 2.35. Korzystając z kreatora raportów, przygotuj prosty raport przeznaczony do wydruku, zawierający informacje o studentach. W raporcie uwzględnij dane pochodzące z czterech tabel: *Kierunki* (pole *Kierunek*), *Studenci_dane* (pola *ID_albumu*, *Nazwisko*, *Imie*), *Studenci_portret_tel* (pole *Portret*) oraz *Studenci_uczelnia* (pole *Rok_studiow*).

Uwaga! Zanim wykonasz to ćwiczenie, musisz połączyć relacją niezwiązaną tabelę *Studenci_portret_tel*. W tym celu powróć do zakładki z tabelami i w oknie relacji ustal relację *jeden-do-jednego* dla tabel *Studenci_dane* i *Studenci_portret_tel*. W oknie relacji dodaj tabelę *Studenci_portret_tel*, a następnie przeciągnij pole z *ID_albumu* z tabeli *Studenci_dane* do tabeli *Studenci_portret_tel* (nie odwrotnie, gdyż nadrzędną jest tabela *Studenci_dane*). Zamknij okno relacji, zapisując zmiany.

W okienku nawigacji wyświetl obiekty *Raporty*. Lista raportów jest jeszcze pusta. Rozpocznij tworzenie raportu, uruchamiając *Kreator raportów*. W pierwszym kroku wybierz kolejno tabele i pola niezbędne do przedstawienia raportu. W kroku drugim możesz wybrać pole, według którego będą grupowane dane. Może to być pole z ID albumu studenta. Krok trzeci to możliwość sortowania danych według określonych kryteriów. Czwarty krok to wybór sposobu wyświetlania danych. Tok postępowania w pierwszych czterech krokach został zilustrowany na rysunku 2.67. W kroku piątym wybierz odpowiedni styl, a następnie nazwę dla raportu *Studenci_dane*. Zakończ pracę kreatora.

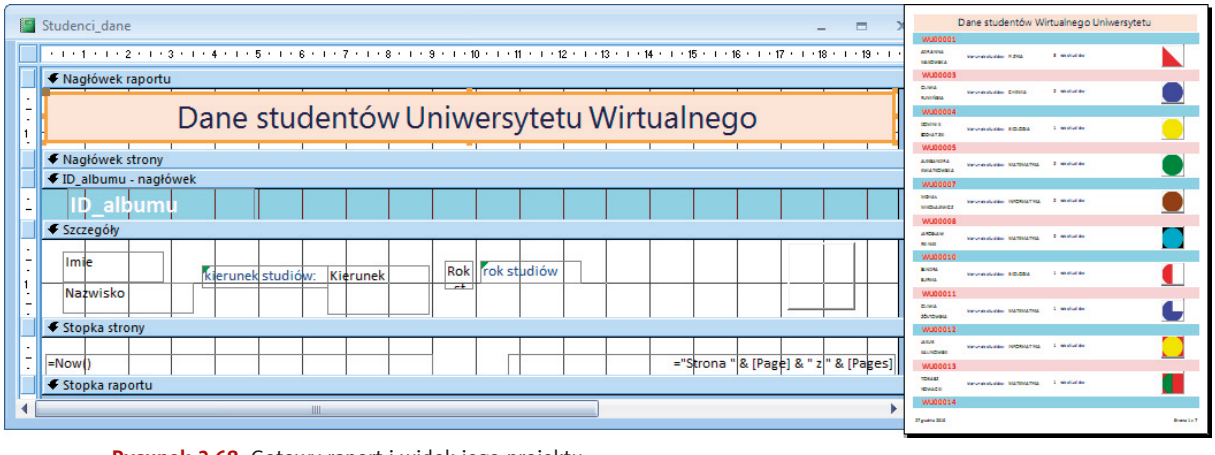
Poszczególne formanty w obszarze raportu zostały rozmieszczone automatycznie. Po raz pierwszy możesz też zobaczyć obiekt OLE, czyli wstawioną do tabeli *Studenci_portret_tel* bitmapę. Jest to kolorowy, niepowtarzalny znak, który ma służyć studentowi jako znak kodowania jego prac. W prawdziwej bazie danych w tym miejscu obiektem OLE byłoby cyfrowe zdjęcie studenta.

W projekcie raportu pojawiły się kolejne obszary będące nagłówkiem oraz stopką strony i raportu. W nagłówku raportu dodaj notkę informującą, że są to dane studentów Uniwersytetu Wirtualnego. W stopce strony raportu automatycznie zamieszczone zostały informacje o dacie generowania raportu oraz numeracja stron. To potrzebne informacje. W celu oddzielenia tej części raportu od danych można zmienić kolor wypełnienia tła. Przejdź do widoku *układu*. Widok ten pozwala na wizualizację raportu. Na karcie



Rysunek 2.67. Kreator raportów

Rozmieszczanie usuń sterowanie układem. Domyślnie włączony został bowiem układ *Tabelaryczny*. Teraz możesz usuwać i przemieszczać formanty. Zmieniając rozmiar obrazu w pierwszym rekordzie raportu, zmieniasz rozmiary obrazów we wszystkich pozostałych rekordach. Z sekcji nagłówka raportu usuń zamieszczone tam etykiety. W sekcji *Szczegóły* poprzestawiaj dowolnie zamieszczone tam formanty. Dobierz odpowiedni rozmiar i położenie dla obiektu OLE. Powróć do widoku *projektu*. W sekcji *Szczegóły* dodaj etykiety *Rok studiów*, *Kierunek studiów* oraz *Średnia ocen:*. Takie informacje są czytelniejsze i bardziej przypominają wizytówkę studenta. Zmień rodzaj i wielkość czcionki oraz kolor wypełnienia dla sekcji nagłówka, w której znajduje się numer ID studenta (wybrany w kreatorze poziom grupowania). Poprzesuwaj formanty i dopasuj ich wielkości tak, aby zmieściły się ich treści. Dokonaj zmian podobnych do pokazanych na rysunku 2.68.

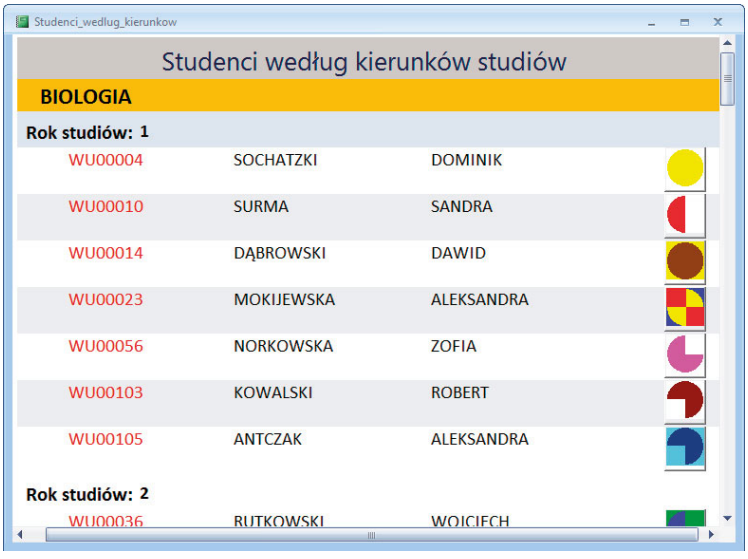


Rysunek 2.68. Gotowy raport i widok jego projektu

Zwróć uwagę na kolor czcionki ID albumu. Mimo że w widoku *projektu* dla tego formantu zmienisz kolor, to i tak w raporcie pojawi się czerwony napis. Jest to związane z właściwościami pola tekstowego macierzystej tabeli, która we właściwości *Format* miała kolor zdefiniowany w następujący sposób: `>@[Czerwony];"BRAK"@[Niebieski]`.

Po zmianach na stronie wydruku mieści się teraz dwa razy więcej wizytówek niż przed korektą. Na rysunku zamieszczono podgląd wydruku pierwszej strony raportu w pomniejszeniu. Cały raport znajduje się na płycie CD dołączonej do podręcznika, w pliku *raport_studenci_dane.pdf*.

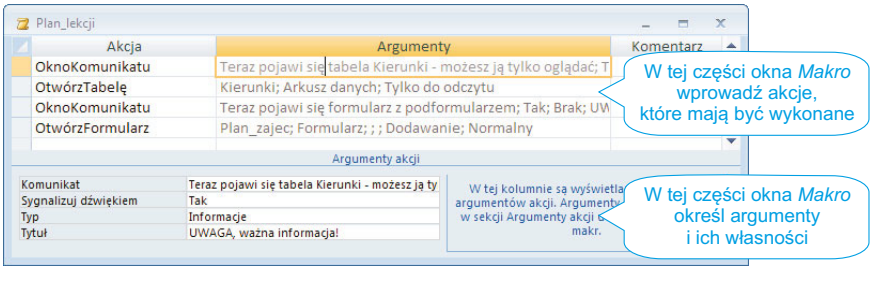
Przygotuj raport *Studenci_wedlug_kierunkow*, który przedstawi zestawienie wszystkich studentów pogrupowanych według kierunku studiów oraz z podziałem na poszczególne roczniki. Przykład takiego zestawienia przedstawiono na rysunku 2.69. Wydruk raportu zamieszczono na płycie CD dołączonej do podręcznika, w pliku *raport_studenci_wedlug_kierunkow.pdf*.



Rysunek 2.69. Raport z wykazem studentów pogrupowanych według kierunku studiów, z podziałem na lata studiów

2.7. Makropolecenia

Makropolecenie (lub krócej: makro) jest *akcją* lub *zestawem akcji*. Każda z nich wykonuje ustaloną operację (czynność, zadanie), taką jak otwarcie formularza lub wydrukowanie raportu. Makra pomagają zautomatyzować często wykonywane zadania, potrafią także w określonym porządku wywoływać kolejne sekwencyjne czynności. I takie właśnie makro przygotujemy. Na rysunku 2.70 pokazano krótki opis okna projektowania makr.

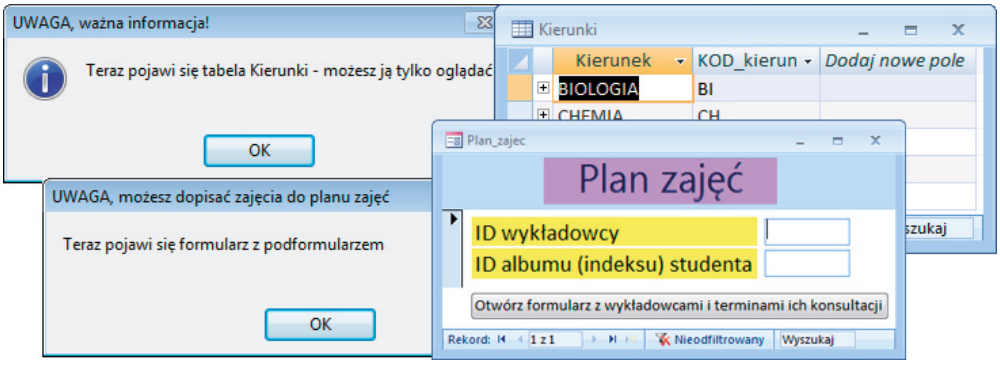


Rysunek 2.70. Okno tworzenia makropolecenia

Ćwiczenie 2.36. Zdefiniuj makropolecenie złożone z sekwencji czterech akcji. Oto zadania tego makra:

1. Wyświetli okno komunikatu z treścią: *Teraz pojawi się tabela Kierunki — możesz ją tylko oglądać*. Okno ma mieć tytuł: *UWAGA, ważna informacja!*
2. Otworzy tabelę *Kierunki* z trybem danych *tylko do odczytu*.
3. Wyświetli okno komunikatu z treścią: *Teraz pojawi się formularz z podformularzem*. Okno ma mieć tytuł: *UWAGA, możesz dopisać zajęcia do planu zajęć*.
4. Otworzy okno formularza *Plan_zajec* w zwykłym trybie edycji.

W okienku nawigacji nie zamieszczono osobno obiektów *Makra*, wybierz więc *Wszystkie obiekty programu Access*. Kategoria *Makra* zostanie automatycznie dodana do okienka nawigacji po utworzeniu pierwszego makropolecenia. Zatem do dzieła. Na karcie *Tworzenie*, w grupie *Inne*, znajduje się przycisk *Makro* umożliwiający wstawianie makr. Proces tworzenia makropolecenia polega na wybieraniu z listy *Akcja* poszczególnych czynności do wykonania i wypełnieniu odpowiednich argumentów akcji zgodnie z poleceniami w ćwiczeniu i podpowiedzią zamieszczoną na rysunku 2.70. Po ustaleniu kolejnych akcji zapisz makro pod nazwą *Plan_lekcji* i uruchom je. Zdefiniowane akcje będą teraz wykonywane kolejno, jak na rysunku 2.71. Utworzone w ten sposób makro składa się z serii akcji. Program Microsoft Access wykonuje te akcje za każdym razem, gdy wykonywane jest makro *Plan_lekcji*. Zauważ, że w okienku nawigacji dodano obiekty *Makra*. Zamknij więc utworzone makro i uruchom je, wybierając z listy obiektów *Makra*. Kolejno wykonywane są zdefiniowane akcje.



Rysunek 2.71. Uruchomione makro złożone z czterech akcji

Po otwarciu pliku bazy danych, w którym zdefiniowano makro, najczęściej pojawia się komunikat z ostrzeżeniem o zabezpieczeniach. Informuje on nas o tym, że określona zawartość bazy danych została zablokowana. W opcjach (wyświetlanych przy komunikacie) bazy danych pochodzącej z wiarygodnego źródła należy włączyć zablokowane makra.

2.8. Ochrona i bezpieczeństwo danych

Nasza baza danych przeznaczona jest dla kilku grup użytkowników.

Grupa pierwsza — administrator lub kilku administratorów z pełnymi uprawnieniami do edycji, projektowania, usuwania danych czy tworzenia nowych formularzy i raportów.

Grupa druga — przeszkolona przez administratora grupa osób pracujących w dziekanatach poszczególnych kierunków studiów. Osoby te wprowadzają dane i pobierają niezbędne do pracy informacje.

Grupa trzecia — grupa wykładowców, czyli użytkowników, którzy zostali dodani do bazy przez upoważnionych nadrzędnych użytkowników (może to być użytkownik grupy pierwszej lub drugiej). Grupa ta ma prawo do wprowadzania, czytania i edycji własnych danych oraz wprowadzania ocen z danego przedmiotu.

Grupa czwarta — grupa studentów, którzy mogą czytać i edytować swoje dane osobowe oraz widzą swoje oceny. Mają prawo zobaczyć terminy konsultacji wykładowców i w określonym przedziale czasowym wpisać się na poszczególne zajęcia do określonych wykładowców.

Jak widać, stworzone przykładowe grupy muszą mieć **odpowiednie uprawnienia** do odpowiednich obiektów, na dodatek **z różnymi do nich prawami**, na przykład tylko do odczytu. Podstawą ograniczeń jest tutaj nie tylko ustawa o ochronie danych osobowych, ale również obawa o utratę czy zniszczenie zgromadzonych danych. Czy w programie Access 2007 jest możliwe tworzenie grup użytkowników bazy i nadawanie im odrębnych uprawnień? Niestety, dla plików zapisanych w nowym formacie *.accdb* nie ma takiej możliwości. Jest to jeszcze możliwe jedynie dla plików zapisanych w starszym formacie *.mdb*.

W programie Office Access 2007 wprowadzono nowe rozszerzenia plików:

ACCDB Rozszerzenie nowego formatu pliku programu Office Access 2007. Zastępuje ono rozszerzenie pliku *MDB*.

ACCDE Rozszerzenie plików programu Office Access 2007 dostępnych w trybie **tylko do wykonywania**. Pliki *ACCDE* nie zawierają kodu źródłowego języka *VBA* (ang. *Visual Basic for Applications*). Użytkownik pliku *ACCDE* może uruchomić kod *VBA*, ale nie może go zmodyfikować. Rozszerzenie pliku *ACCDE* zastępuje rozszerzenie *MDE*.

ACCDT Rozszerzenie plików **szablonów** bazy danych programu Access.

ACCDR Nowe rozszerzenie plików, które umożliwia otwarcie bazy danych w trybie czasu wykonywania. Wystarczy zmienić rozszerzenie pliku bazy danych z *.accdb* na *.accdr*, aby utworzyć **zablokowaną** wersję bazy danych programu Office Access 2007. Aby przywrócić pełne funkcje, rozszerzenie pliku należy zmienić z powrotem na *.accdb*.

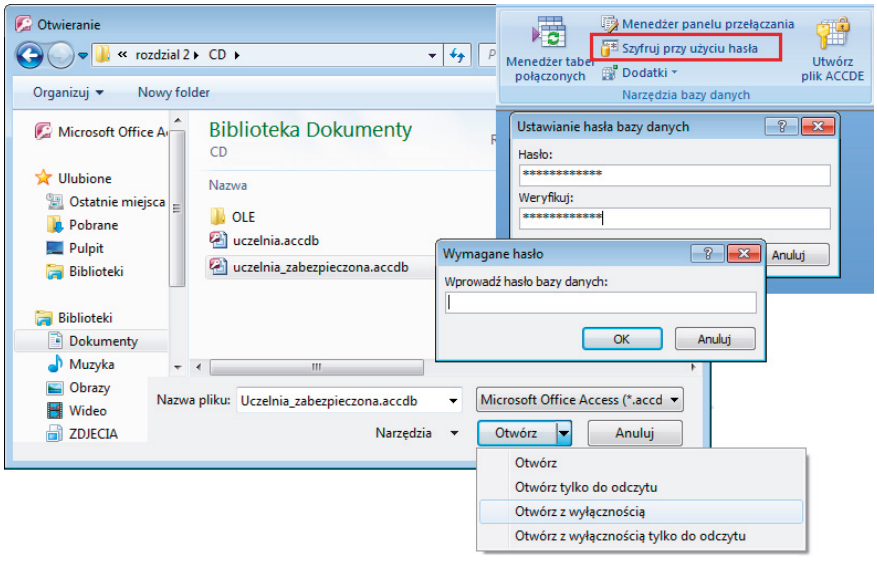
2.8.1. Ochrona bazy danych hasłem

Jeśli zostanie ustanowione hasło dla bazy danych, wszyscy użytkownicy muszą je wprowadzać przed otwarciem pliku bazy danych. Zastosowanie hasła bazy danych jest łatwym sposobem zapobiegania otwieraniu bazy danych *przez niepożądanych użytkowników*. Jednak po otwarciu bazy danych każdy ma pełny dostęp do wszystkich zgromadzonych danych, ponieważ nie zostały podjęte żadne inne środki bezpieczeństwa. Aby rozgraniczyć użytkowników i zróżnicować ich prawa, trzeba w bazie zdefiniować zabezpieczenie na poziomie użytkownika.

Ćwiczenie 2.37. Zamknij bazę danych *uczelnia.accdb* i utwórz jej kopię *uczelnia_zabezpieczona.accdb*. Zabezpiecz tę kopię bazy hasłem. W tym celu musisz otworzyć plik bazy w *trybie wyłączności*. Z menu *Przycisk pakietu Office* wybierz polecenie *Otwórz*. W oknie *Otwieranie* wskaż plik bazy i kliknij strzałkę na prawo od przycisku *Otwórz*, a następnie wybierz polecenie *Otwórz z wyłącznością*.

Następnie na karcie *Narzędzia bazy danych*, w grupie *Narzędzia bazy danych*, wskaż polecenie *Szyfruj przy użyciu hasła*. W okienku *Ustawianie hasła bazy danych*, w polu *Hasło:*, wpisz hasło, następnie powtórz je w polu *Weryfikuj:*. Należy używać **silnych haseł**, w których występują małe i wielkie litery, cyfry i symbole. W słabych hasłach nie występują połączenia tych elementów. Silne hasło to na przykład `V3dgA!t5`. Słabe hasło: `Ania1990`. Hasła powinny mieć co najmniej 8 znaków. Lepsze są jednak hasła liczące co najmniej 14 znaków.

Następnym razem, gdy otworzysz bazę danych, pojawi się okno dialogowe z żądaniem podania hasła, jak na rysunku 2.72. Każdy, kto zna hasło, może otworzyć bazę i mieć dostęp do jej zasobów. W dowolnym momencie możesz odszyfrować bazę. Pamiętaj jednak o uruchomieniu jej w trybie z wyłącznością.



Rysunek 2.72. Otwieranie pliku bazy danych z wyłącznością i zabezpieczanie go hasłem

I jeszcze jedno — zapomnianego hasła nie można odtworzyć, a w konsekwencji nie można otworzyć chronionej nim bazy danych. Utwórz więc hasło trudne dla włamywacza, ale łatwe dla użytkownika.

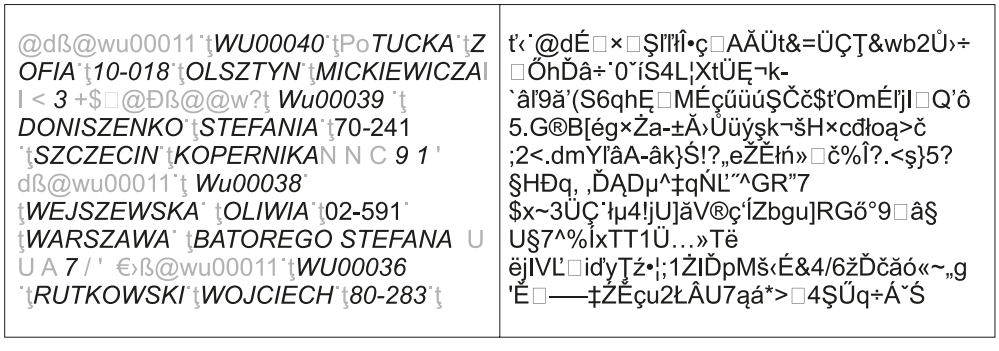
2.8.2. Kodowanie i dekodowanie bazy danych

W narzędziu szyfrowania dostępnym w programie Office Access 2007 połączono i ulepszono funkcje dwóch starszych narzędzi — *haseł* i *kodowania* bazy danych. Zaszyfrowanie bazy danych za pomocą hasła powoduje, że danych nie można odczytać innymi narzędziami i użytkownicy muszą wprowadzić hasło, aby móc używać bazy danych. W programie Access 2007 zastosowano algorytm szyfrowania silniejszy niż we wcześniejszych wersjach programu Access.

Kodowanie (zaszyfrowanie) bazy danych uniemożliwia jej odczytanie przy użyciu programu narzędziowego lub edytora tekstu. Dekodowanie (odszyfrowanie) bazy danych przywraca jej stan pierwotny. W programie Office Access 2007 wraz z zabezpieczeniem bazy danych hasłem następuje jej zakodowanie (zaszyfrowanie), co uniemożliwia odczytanie danych.

Efekt działania szyfrowania bazy danych pokazano na rysunku 2.73. W lewym oknie widnieje tekst bazy danych odczytany w Notatniku. Po prawej stronie znajduje się ten sam plik odczytany po zaszyfrowaniu. Przed zaszyfrowaniem tekst na pierwszy rzut oka wydaje się nieczytelny, jednak łatwo wskazać zawartość rekordów w tabeli bazy i odczytać tajne dane osobowe czy pracownicze.

Zaszyfrowanie bazy danych powoduje zakodowanie danych w tabelach i utrudnia nieupoważnionym użytkownikom odczytywanie danych.



Rysunek 2.73. Zawartość bazy odczytana w Notatniku przed zakodowaniem i po nim

2.9. Zadania

Zadanie 2.1. Firma przewozowa ma kilka pojazdów transportowych, zatrudnia również kierowców. Rejestrowane są wszystkie kursy poszczególnych pojazdów. Dane gromadzone były w folderze *Firma_transportowa*, w trzech plikach tekstowych: *pojazdy.txt*, *kierowcy.txt* i *trasy.txt*. Folder wraz z plikami zamieszczono na płycie CD dołączonej do podręcznika.

W pliku *pojazdy.txt* zapisane są dane dotyczące samochodów firmy przewozowej. W każdej linijce znajduje się ID pojazdu, numer rejestracyjny pojazdu oraz stawka dla kierowcy za 1 km kursu tym pojazdem. W pliku *kierowcy.txt* zgromadzono informacje o kierowcach: ich ID oraz imiona i nazwiska. W pliku *trasy.txt* zapisane są informacje dotyczące ponad 200 wykonanych kursów. W każdej linijce znajduje się numer kursu, ID pojazdu, data i godzina kursu, długość kursu w kilometrach oraz identyfikator kierowcy.

Twoim zadaniem jest zaimportowanie tych danych z plików tekstowych do tabel bazy danych (wykorzystaj istniejące w tabelach pola jako pola kluczowe tabel) oraz wykonanie bazodanowych operacji, dzięki którym udzielisz odpowiedzi na poniższe pytania.

- Podaj dane o kursach dłuższych niż 100 km. W zestawieniu podaj daty kursów, ich długości i nazwiska kierowców. Posortuj rekordy w porządku malejącym według długości kursów.
- Który pojazd najczęściej wyjeżdżał w trasę, a który przejechał najwięcej kilometrów?
- Podaj dane o kursach rozpoczętych w godzinach nocnych, czyli po 20^{00} i przed 6^{00}. W zestawieniu uwzględnij daty, numery rejestracyjne pojazdów oraz imiona i nazwiska kierujących pojazdami. Posortuj dane według dat (od najwcześniej wykonanego kursu).
- Podaj liczbę kursów wykonanych przez każdego z kierowców. Ile kilometrów przejechał każdy z nich?
- Podaj dane o kursach odbytych w trzecim kwartale 2010 roku. W zestawieniu uwzględnij daty kursów, numery rejestracyjne pojazdów i nazwiska kierowców. Wyświetlenie zestawienia poprzedź podaniem parametru: liczby od 1 do 6 będącej numerem ID pojazdu. Posortuj dane według dat.
- Sporządź zestawienia (dynamiczną tabelę oraz raport), w których będą obliczone sumy zarobków zgrupowane według pojazdów każdego z kierowców. Posortuj zestawienia według nazwisk kierowców, a następnie według dat.
- Firma zarabia na każdym przewozie dwa razy więcej, niż wynosi stawka dla kierowcy. Podaj, jaki był zarobek firmy w 2010 roku.
- Który kierowca zarobił najwięcej, a który najmniej w tej firmie? Podaj ich nazwiska i kwoty zarobionych pieniędzy.

Zadanie 2.2. Odpowiedz na poniższe pytania.

1. Czym jest kartotekowa baza danych? Podaj przykład.
2. Do czego służy baza danych? Podaj kilka przykładów zastosowania.
3. Jaką bazę danych nazywamy relacyjną?
4. Co może wpłynąć na zmniejszenie wielkości pliku bazy danych?
5. W jaki sposób uniknąć zjawiska anomalii? Podaj przykłady.
6. Z ilu tabel może składać się relacyjna baza danych?
7. Jaką funkcję pełni klucz podstawowy?

8. Kiedy klucz jest kluczem podstawowym, a kiedy obcym w tabeli danych?
9. Jaki rodzaj pola wybiera się zwykle dla kluczy podstawowych i dlaczego?
10. Na czym polega relacja *jeden-do-jednego*? Podaj przykład.
11. Na czym polega relacja *jeden-do-wielu*? Podaj przykład.
12. Na czym polega relacja *wiele-do-wielu*? Podaj przykład.
13. Kiedy posłużysz się kwerendą krzyżową?
14. Czy można zabezpieczyć bazę danych przed jej otwarciem przez przypadkowego użytkownika?
15. Dlaczego stosujemy kodowanie bazy danych?

Wybrane zadania maturalne

Na płycie CD dołączonej do podręcznika zamieszczono treści wybranych zadań maturalnych oraz niezbędne dane do ich rozwiązania. Zadania maturalne wymagają od ucznia/uczennicy umiejętności importowania danych oraz ich analizy w celu uzyskania informacji na temat danych zgromadzonych w plikach tekstowych. Rozwiązanie tych zadań zapewni lepsze przygotowanie uczniów do egzaminu maturalnego.

- *matura2.1.pdf* (*Egzamin próbny styczeń 2005 r. Arkusz II, zadanie 5. BAZA DANYCH*).
- *matura2.2.pdf* (*Egzamin maj 2005 r. Arkusz II, zadanie 6. PUCHAR ŚWIATA W SKOKACH*).
- *matura2.3.pdf* (*Egzamin maj 2006 r. Arkusz I, zadanie 3. BAZA DANYCH*).
- *matura2.4.pdf* (*Egzamin maj 2007 r. Arkusz II, zadanie 4.*).
- *matura2.5.pdf* (*Egzamin maj 2008 r. Arkusz II, zadanie 6. WYPADKI*).
- *matura2.6.pdf* (*Informator maturalny od 2009 r. Arkusz II, poziom podstawowy, zadanie 6. PODRÓŻNI*).
- *matura2.7.pdf* (*Informator maturalny od 2009 r. Arkusz II, poziom rozszerzony, zadanie 6. UCZNIOWIE I KLASY*).
- *matura2.8.pdf* (*Egzamin maj 2009 r. Arkusz II, poziom podstawowy, zadanie 6. MIESZKANIA*).
- *matura2.9.pdf* (*Egzamin maj 2009 r. Arkusz II, poziom rozszerzony, zadanie 6. LEKARZE*).
- *matura2.10.pdf* (*Egzamin maj 2010 r. Arkusz II, poziom podstawowy, zadanie 6. NOWORODKI*).
- *matura2.11.pdf* (*Egzamin maj 2010 r. Arkusz II, poziom rozszerzony, zadanie 6. SZKOŁA*).

Rozdział 3.

SYSTEMY OPERACYJNE I SIECI KOMPUTEROWE

W pierwszej części tego rozdziału zapoznasz się z podstawowymi informacjami związanymi z najważniejszymi podzespołami komputera. Omówimy proces uruchamiania komputera, a także budowę i funkcje systemu operacyjnego. A podczas ćwiczeń praktycznych poznasz podstawowe możliwości przykładowych systemów operacyjnych. Do ich przeanalizowania wykorzystasz maszyny wirtualne.

W dalszej części rozdziału omówimy sieci komputerowe — ich strukturę, protokoły sieciowe, model OSI, a także tworzenie strony WWW z wykorzystaniem języka PHP i administrowanie systemem zarządzania treścią.

Przyzwyczailiśmy się już do obecności komputera w domu, w szkole, w sklepie, w banku, w miejscu pracy. Sama jego obecność wokół nas nie jest już niczym nadzwyczajnym. Jednak jeśli weźmiemy pod uwagę technologię i coraz szybszy rozwój sprzętu komputerowego (*hardware*), trzeba przyznać, że komputer jest urządzeniem niezwykłym i bardzo skomplikowanym. Za rozwojem technologii i techniki stara się podążać rozwój programów komputerowych (*software*).

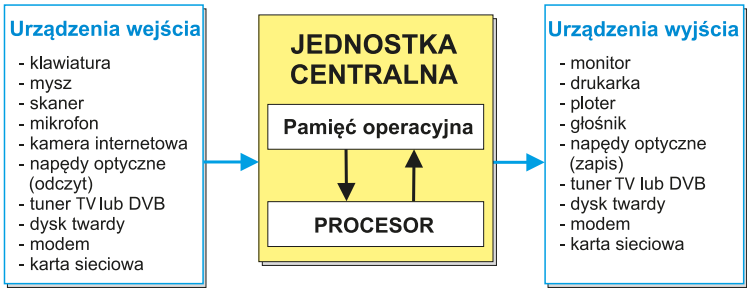
Najwolniej rozwijają się systemy operacyjne, gdyż ich zmiany pociągają za sobą konieczność unowocześniania oprogramowania użytkowego. Co prawda często zachowywana jest *zgodność wsteczna (kompatybilność)*, co oznacza, że na komputerze pracującym pod kontrolą nowszej wersji systemu operacyjnego można uruchomić program ze starszej wersji tego systemu, jednak komfort pracy z takim oprogramowaniem może być niezadowalający. Przy zmianie systemu operacyjnego problemem jest przygotowanie przez wszystkich producentów sprzętu odpowiednich sterowników, które będą efektywnie i prawidłowo obsługiwały dane urządzenia w nowym systemie.

Już ten krótki wstęp pozwala sformułować wniosek, że każdy system operacyjny jest ściśle powiązany ze sprzętem, na którym jest uruchamiany. Trudno więc zrozumieć funkcje oraz rolę systemu operacyjnego bez znajomości budowy komputera.

3.1. Budowa komputera

Najważniejszym elementem komputera jest *jednostka centralna*. W niej znajduje się *procesor* wraz z *pamięcią operacyjną*. Można więc śmiało powiedzieć, że **zestaw komputerowy**

(komputer) to jednostka centralna wraz z urządzeniami zewnętrznymi (wejścia-wyjścia). Na rysunku 3.1 przedstawiono uproszczony schemat zestawu komputerowego.



Rysunek 3.1. Uproszczony schemat zestawu komputerowego

Wskazówka

W języku potocznym często używa się określenia „jednostka centralna" w odniesieniu do obudowy komputera z całą jej zawartością. Jednak pojęcie ściśle informatyczne odnosi się jedynie do procesora wraz z pamięcią operacyjną.

Pamięć operacyjna służy do chwilowego przechowywania *danych wejściowych* i *wyjściowych*. W niej przechowuje się uruchomione programy. Programy przekazują rozkazy do procesora. We współczesnych komputerach funkcję pamięci operacyjnej spełnia pamięć *RAM* (ang. *Random Access Memory*). Jest to pamięć o dostępie swobodnym, co oznacza, że można zapisywać dane do komórek pamięci, odczytywać je oraz kasować. Należy pamiętać, że dane w pamięci RAM zostaną skasowane po wyłączeniu zasilania pamięci.

Procesor jest jednym z najważniejszych elementów komputera.

Definicja

Procesor (*CPU*, ang. *Central Processing Unit* — centralna jednostka wykonawcza) to układ scalony, którego działanie polega na wykonywaniu instrukcji programów. Nadzoruje i synchronizuje pracę wszystkich urządzeń w komputerze.

Współczesne procesory to układy o *bardzo dużej skali integracji VLSI* (ang. *Very Large Scale of Integration*), co oznacza, że procesor składa się z bardzo dużej liczby elementów podstawowych, którymi są *tranzystory*. Dla przykładu procesor Intel Core i7, pokazany na rysunku 3.2, jest zbudowany z około 730 milionów tranzystorów w technologii 32 nm.

Ile to jest 32 nm? Poniższe porównania pozwolą nam wyobrazić sobie rozmiary tego tranzystora:

- 1 nm = 10^{-9} m = 0,000 000 001 m.
- W jednym metrze znajduje się 1 miliard nanometrów (1 m = 1 000 000 000 nm).
- Koszt wytworzenia jednego tranzystora we współczesnym mikroprocesorze to zaledwie jedna milionowa średniej ceny tranzystora z 1968 roku. Gdyby wszystko tak taniało...
- W średnicy ludzkiego włosa można zmieścić prawie 2000 tranzystorów 32-nanometrowych, a na główce szpilki — ponad 30 milionów.



Rysunek 3.2. Procesor czterordzeniowy Intel Core i7-2600K

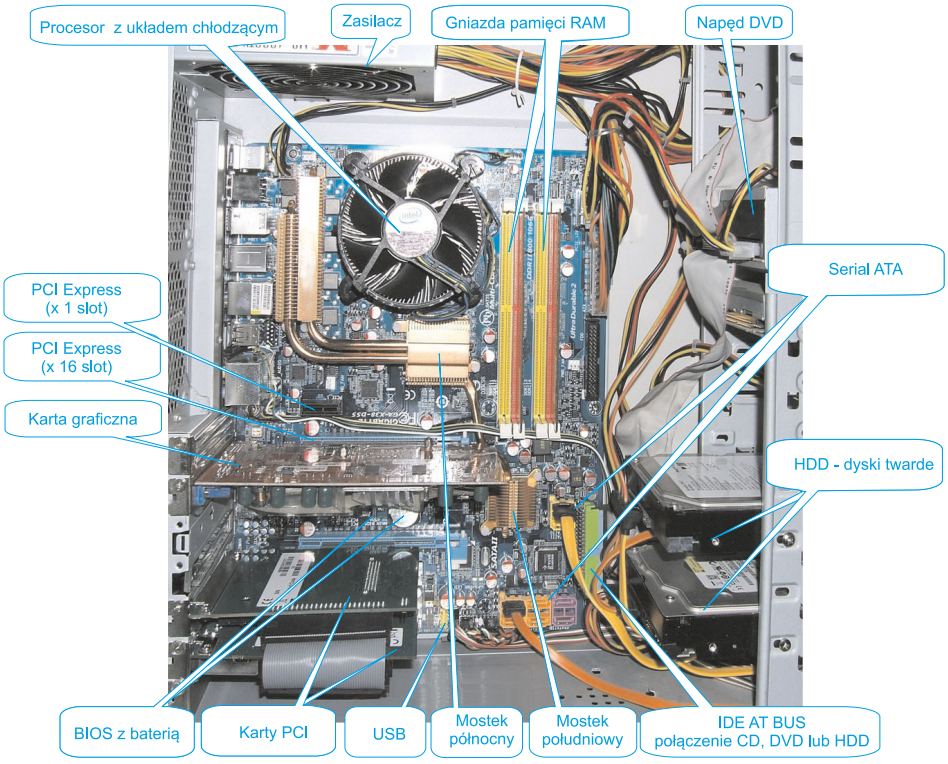
- Tranzystor 32-nanometrowy może włączać się i wyłączać około 300 miliardów razy na sekundę. W czasie, który zajmuje włączenie i wyłączenie tranzystora, wiązka światła przebywa odległość mniej niż ćwierć centymetra.

Pozostałe elementy komputera to tzw. *urządzenia peryferyjne (zewnętrzne)*. Ich funkcją jest dostarczanie informacji (danych, rozkazów) do jednostki centralnej lub odbieranie z niej informacji.

Wyróżniamy następujące grupy urządzeń peryferyjnych (zewnętrznych):
- urządzenia wejściowe, na przykład klawiatura, mysz, skaner,
- urządzenia wyjściowe, na przykład monitor ekranowy, drukarka, ploter, głośnik,
- urządzenia wejścia-wyjścia, na przykład modem, dysk twardy.

Wszystkie elementy komputera są osadzone na *płycie głównej* komputera lub z nią połączone. Część elementów komputera podczas pracy wydziela dużo ciepła. Aby ich praca przebiegała prawidłowo, należy zadbać o prawidłową temperaturę. Do tego celu służą wszelkiego rodzaju radiatory, wentylatory oraz zaawansowane układy chłodzenia cieczą. To dlatego tak łatwo można zlokalizować procesor. Jest on schowany pod największym radiatorem, na którym osadzony jest sporej wielkości wentylator. Zdjęcie na rysunku 3.3 przedstawia podstawowe elementy budowy komputera.

W każdym procesorze można wyróżnić kilka zasadniczych modułów. Są one odpowiedzialne za przepływ danych oraz instrukcji w procesorze. Do najważniejszych modułów zaliczamy:
- *moduł wstępnego pobierania i dekodowania instrukcji*, który jest odpowiedzialny za dostarczenie kolejnych poleceń z pamięci operacyjnej i przekazanie ich do odpowiedniej jednostki wykonawczej;
- *główny moduł wykonawczy*, który jest *jednostką arytmetyczno-logiczną ALU* (ang. *Arithmetic Logic Unit*) odpowiedzialną za przetwarzanie danych stałopozycyjnych (całkowitych). Ma niewielką pamięć, nazywaną zestawem rejestrów. Każdy rejestr to pojedyncza komórka używana do chwilowego przechowywania danych i wyników. Natomiast *FPU* (ang. *Floating Point Unit* — koprocesor numeryczny) wykonuje wszystkie obliczenia zmiennoprzecinkowe. Kiedyś koprocesory były osobnymi układami elektronicz-



Rysunek 3.3. Budowa komputera

nymi. Od momentu pojawienia się procesora 486DX zaczęto je montować w jednej strukturze wraz z procesorem;

- *moduł wyjściowy procesora*, który jest odpowiedzialny za przekierowanie danych będących wynikiem obliczeń do odpowiedniego adresu w pamięci operacyjnej lub urządzenia wyjścia.

Procesory możemy podzielić ze względu na długość słowa, na którym wykonywane są operacje obliczeniowe. Rozróżniamy więc następujące procesory:

Długość słowa	Przykładowe procesory firmy Intel
4-bitowe	4004 (1971 r.)
8-bitowe	8008 oraz 8080 (1972 r.)
16-bitowe	8088 (1979 r.), 80286 (1982 r.), 80386SX (1985 r.)
32-bitowe	80486 (1989 r.), Pentium 60/66MHz (1993 r.), Pentium II, Pentium III
64-bitowe	Intel Xeon (2005 r.)
128-bitowe	Jeszcze nieinstalowany jako CPU w zastosowaniach cywilnych

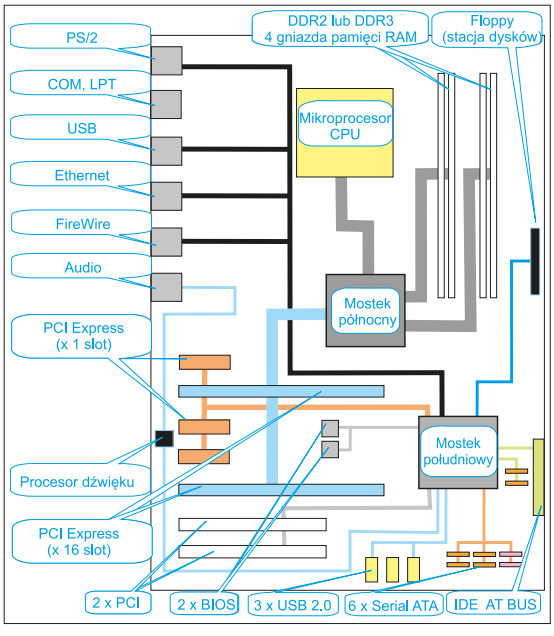
Obecnie w jednostkach centralnych montowane są procesory 32- lub 64-bitowe.

Decyzja o wyborze procesora pociąga za sobą decyzję o wyborze płyty głównej, która pozwala na optymalne wykorzystanie możliwości procesora.

O jakości i o możliwościach płyty głównej decydują jej dwa główne elementy: *mostek północny* (*northbridge*) oraz *mostek południowy* (*southbridge*). Są to kluczowe elementy płyty głównej, które decydują między innymi o rodzaju magistrali. *Magistrala* to określenie, które często pojawia w specyfikacjach wielu komputerowych komponentów, takich jak: procesory, płyty główne, karty graficzne, pamięci flash czy urządzenia sieciowe. Czym zatem jest i do czego służy magistrala?

Magistrala (ang. *bus*), potocznie zwana *szyną systemową*, służy do przekazywania danych, adresów czy instrukcji sterujących w różne miejsca komputera. Przekazywanie informacji może być realizowane na dwa sposoby. Gdy informacje są przesyłane jednocześnie, mamy do czynienia z *magistralą równoległą*, gdy zaś przesyłanie odbywa się po kolei, jedna informacja za drugą — z *magistralą szeregową*.

Na rysunku 3.4 przedstawiony jest ogólny schemat budowy płyty głównej wraz z zaznaczonymi najważniejszymi jej komponentami. Przykładowe rozmieszczenie elementów jest zgodne z rysunkiem 3.3.



Rysunek 3.4. Przykładowe rozmieszczenie elementów na płycie głównej

W komputerach montowane są magistrale różnego rodzaju. Najszybsza w systemie jest *magistrala procesora*, zwana główną. Pośredniczy w przesyłaniu danych między procesorem (CPU) a pamięcią operacyjną RAM. Wszystkie magistrale są ze sobą połączone hierarchicznie, a hierarchia ta wiąże się przede wszystkim z szybkością ich pracy (wolniejsza zawsze podłączana jest do szybszej).

Kolejne w hierarchii są *magistrale lokalne*, łączące system z *kartami rozszerzeń* (karta graficzna, dźwiękowa, sieciowa…). Do tej grupy zaliczają się między innymi magistrale: PCI, AGP i PCI Express. Jeszcze niżej znajdują się *magistrale wejścia-wyjścia*, nazywane również *peryferyjnymi* (ang. *peripheral bus*) lub *magistralami wejścia-wyjścia (I/O)*. Służą one, podobnie jak lokalne, do podłączania urządzeń peryferyjnych wymagających wolniejszej komunikacji, takich jak: napędy dysków, porty USB, porty szeregowe i równoległe oraz PS/2 (port wykorzystywany do podłączania klawiatury i myszy).

Obecnie zarówno procesory, jak i magistrale główne są 64-bitowe. Dostosowano je również do współpracy z procesorami dwu- i czterordzeniowymi, takimi jak Intel Core 2 Duo, Core 2 Quad czy serwerowy Xeon z serii E5000. Niemniej jednak historia magistrali systemowej dobiega powoli końca. W procesorach Nehalem pojawił się nowy interfejs komunikacyjny, oparty na połączeniach punkt-punkt, opisany w rozdziale 5.

Mostek północny ma znaczący wpływ na wydajność systemu. Określa szybkość, z jaką dane mogą płynąć pomiędzy procesorem, pamięcią operacyjną i podsystemem graficznym.

Mostek północny komunikuje się również z wolniejszym *mostkiem południowym*, który jest odpowiedzialny za obsługę wolniejszych magistrali komunikujących się z różnymi urządzeniami poprzez:

- złącza *Serial ATA* (w skrócie: *SATA* — ang. *Serial Advanced Technology Attachment*) — szeregowa magistrala umożliwiająca podłączenie szybkich dysków twardych oraz napędów optycznych. Obecnie w sprzedaży dostępne są dyski z kontrolerem wyposażonym w magistralę SATA3, która umożliwia transfer danych z prędkością 6 gigabitów na sekundę;
- złącza *AT-BUS* (w skrócie: *ATA* — ang. *Advanced Technology Attachment*) — przestarzały, wychodzący obecnie z użycia interfejs systemowy, przeznaczony do komunikacji z dyskami twardymi poprzez magistralę równoległą. Standard ATA nie jest już rozwijany w kierunku zwiększania szybkości transmisji;
- magistralę *PCI* (ang. *Peripheral Component Interconnect*) — magistrala komunikacyjna służąca do podłączenia urządzeń do płyty głównej (karta graficzna, karta dźwiękowa, karta sieciowa, karta telewizyjna…);
- porty *USB* (ang. *Universal Serial Bus*) — uniwersalna magistrala szeregowa, rodzaj portu komunikacyjnego komputerów z pamięcią flash, dyskami twardymi, drukarką, klawiaturą, skanerem i innymi urządzeniami;
- gniazda *PS/2* — port komunikacyjny opracowany przez firmę IBM. Jest on odmianą portu szeregowego przeznaczonego do podłączania klawiatury i myszy;
- gniazda *LPT* — port równoległy wykorzystywany do podłączenia urządzeń peryferyjnych, takich jak drukarki, skanery, plotery. Obecnie porty te są wypierane przez porty USB.

Mostek południowy jest odpowiedzialny również za komunikację z BIOS-em, szczegółowo opisanym w punkcie 3.2, oraz za obsługę karty dźwiękowej zainstalowanej na płycie głównej.

Obecnie wraz z pojawieniem się procesorów Intel Core i7 nastąpiły pewne zmiany w funkcjach, jakie pełnią procesor oraz mostek północny. Po przeniesieniu kontrolera pamięci z mostka północnego do procesora mostek północny odpowiada tylko za komunikację pomiędzy mostkiem południowym a procesorem oraz obsługę magistrali PCI Express dla kart graficznych. Linie PCI Express x 1 dla pozostałych urządzeń obsługuje teraz mostek południowy.

3.2. BIOS i rozruch komputera

Proces uruchamiania komputera i jego przygotowania do pracy przebiega w kilku następujących po sobie etapach. Prześledźmy je i rozszyfrujmy niektóre informacje pojawiające się na ekranie monitora.

Bezpośrednio po włączeniu komputera uaktywniany jest BIOS.

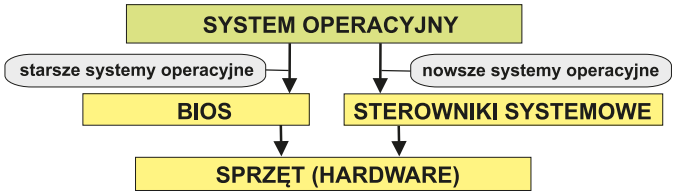
Definicja

BIOS (ang. *Basic Input/Output System*) — podstawowy system wejścia-wyjścia.

BIOS to program zapisany w pamięci stałej na płycie głównej. Zawiera zestaw podstawowych procedur pośredniczących pomiędzy systemem operacyjnym a sprzętem.

Jest on uruchamiany jako pierwszy, przed systemem operacyjnym, i w większości komputerów PC pełni cztery podstawowe funkcje:

1. Procedura POST (ang. *Power On Self Test*) **wykonuje test poprawności** działania podstawowych układów i podzespołów, między innymi karty graficznej, co skutkuje wyświetleniem komunikatu informującego o rodzaju karty graficznej. Następnie testowane są: płyta główna, procesor, pamięć RAM, zainstalowane napędy HDD (dyski twarde), dyski optyczne (na przykład CD-ROM) oraz pozostałe komponenty komputera (tzw. urządzenia wejścia-wyjścia). Jeśli któreś z testowanych urządzeń jest niesprawne, procedura POST zatrzymuje pracę komputera i zwraca (w postaci kodu) informację o rodzaju napotkanego błędu. Rodzaj błędu jest sygnalizowany sekwencją odpowiednich dźwięków generowanych przez głośniczek systemowy lub w postaci odpowiedniego komunikatu na ekranie monitora, co pozwala na szybszą lokalizację ewentualnego uszkodzenia komputera.
2. Program BIOS Setup **umożliwia konfigurację systemu**. Zazwyczaj wyświetlany jest zestaw menu, który pojawi się po wciśnięciu określonego klawisza w trakcie wykonywania procedury POST, (najczęściej *Del* lub *F2*). Program BIOS Setup umożliwia konfigurację płyty głównej oraz takich parametrów jak: data, czas, hasła, parametry dysków twardych, definicje innych podstawowych ustawień systemowych. Możliwa jest też konfiguracja ustawień związanych z funkcją zarządzania energią oraz określenie kolejności urządzeń, z których będzie wgrywany system operacyjny. Zmiany te są zapisywane w pamięci CMOS. Pamięć ta musi być cały czas zasilana (stąd też obecność baterii na płycie głównej). Wyczerpanie się baterii lub skasowanie zawartości pamięci CMOS powoduje, że BIOS powraca do ustawień domyślnych, fabrycznych, które do czasu ponownego wprowadzenia poprawnych ustawień gwarantują pracę komputera, jednak jest ona wolniejsza i sam proces uruchamiania komputera trwa dłużej.
3. Jeśli procedura POST przebiegła bez żadnego problemu, **uruchamiany jest program rozruchowy** (ang. *boot loader*), którego zadaniem jest odszukanie systemu operacyjnego i wczytanie go do pamięci operacyjnej. System operacyjny może być wgrany z dysku twardego, z napędu optycznego, dyskietki, pamięci flash, a nawet poprzez sieć komputerową. Proces poszukiwania systemu operacyjnego przebiega w kolejności określonej w ustawieniach BIOS-u.
4. BIOS **przydziela zasoby systemowe** urządzeniom, które tego wymagają. Zasoby przydzielone przez BIOS są wykorzystywane przez system operacyjny. Jednak obecnie to system operacyjny przejmuje później kontrolę nad przydzielaniem zasobów systemowych odpowiednim urządzeniom. Proces ten przedstawiony jest schematycznie na rysunku 3.5.



Rysunek 3.5. Przydzielanie zasobów poprzez BIOS lub sterowniki systemu operacyjnego

W środowisku Windows często możemy spotkać się z konfliktem sprzętowym, wynikającym z tego, że dwa urządzenia odwołują się do tych samych zasobów sprzętowych (na przykład do tego samego zakresu pamięci RAM), co prowadzi do destabilizacji pracy tych urządzeń.

BIOS ma już swoje lata i aż dziw bierze, że tak stare rozwiązanie nie doczekało się jeszcze następców. Choć nie do końca, Intel opracował bowiem coś, co w najbliższej przyszłości powinno zastąpić BIOS. Mowa tu o projekcie zapoczątkowanym przez firmę IBM — *UEFI* (ang. *Unified Extensible Firmware Interface*), który zredukuje czas rozruchu komputera z kilkudziesięciu do kilku sekund. Umożliwi również obsługę dysków większych niż 2 TB. Niektóre współczesne systemy operacyjne potrafią już współpracować z UEFI. Należą do nich Vista oraz Windows 7 w wersjach 64-bitowych.

3.3. System operacyjny

Definicja

System operacyjny (*według Abrahama Silberschatza*) jest programem, który działa jako pośrednik między użytkownikiem komputera a sprzętem komputerowym. Zadaniem systemu operacyjnego jest tworzenie środowiska, w którym można uruchomić inne programy w sposób zapewniający ich poprawne i efektywne wykorzystanie.

Dzięki systemowi operacyjnemu możliwe jest więc **uruchamianie innych programów**. System operacyjny **nadzoruje wykonywanie tych programów** i **udostępnia im zasoby komputera**. To dzięki systemowi operacyjnemu możliwe są między innymi: **zapisywanie danych na dyskach**, **komunikacja z innymi urządzeniami i komputerami w sieci**.

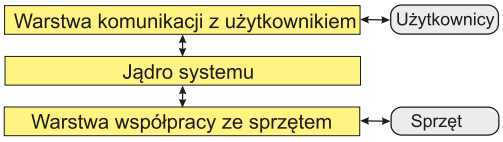
Bez systemu operacyjnego praca z komputerem byłaby niemożliwa. Jest on bowiem elementem pośredniczącym między sprzętem (ang. *hardware*), który często jest nazywany *fizyczną częścią komputera*, a programami (ang. *software*) oraz użytkownikami. Należy pamiętać, że użytkownikiem może być zarówno człowiek, jak i maszyna lub inny komputer.

3.3.1. Budowa systemu operacyjnego

W każdym systemie operacyjnym występują mniej lub bardziej wyodrębnione warstwy składające się na architekturę systemu.

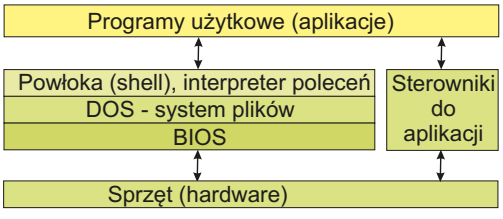
W *ogólnym modelu systemu operacyjnego*, którego schemat zilustrowano na rysunku 3.6, można wyodrębnić następujące warstwy, przypisując im poszczególne zadania:

- *powłoka* (ang. *shell*), stanowiąca interfejs użytkownika (komunikacja z użytkownikiem);
- *jądro systemu* (ang. *kernel*), realizujące jego funkcje (zarządzanie plikami, uruchamianie aplikacji);
- warstwa odpowiedzialna za *współpracę ze sprzętem — HAL* (ang. *Hardware Abstraction Layer* — warstwa abstrakcji sprzętowej), która stanowi ogniwo pośredniczące między sprzętem a jądrem systemu operacyjnego.



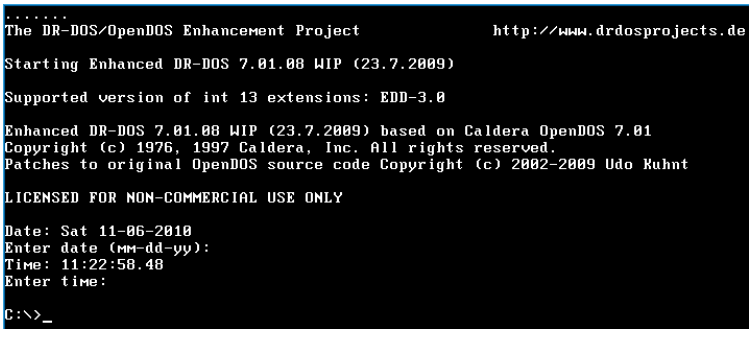
Rysunek 3.6. Struktura warstwowa systemu operacyjnego

Strukturę bardzo podobną do ogólnego modelu ma *system operacyjny DOS* (ang. *Disc Operating System*), którego schemat pokazano na rysunku 3.7.



Rysunek 3.7. Budowa systemu DOS

Na szczególną uwagę zasługuje warstwa powłoki, która zapewnia komunikację między użytkownikiem a systemem operacyjnym. System operacyjny zgłasza swoją gotowość do pracy znakiem zachęty w trybie tekstowym, jak na rysunku 3.8. Komunikacja odbywa się więc w trybie tekstowym, co oznacza, że wprowadzenie polecenia i zatwierdzenie go klawiszem *Enter* wywołuje reakcję systemu operacyjnego. W pierwszej kolejności wprowadzony tekst jest *interpretowany przez system operacyjny* jako jego *polecenie systemowe*, które, jeśli zostanie rozpoznane, jest wykonywane. W przypadku gdy wprowadzony tekst nie jest poleceniem systemowym, system operacyjny sprawdza, czy wprowadzone polecenie to nazwa pliku wykonywalnego, czyli pliku z rozszerzeniem **.exe*, **.com* lub **.bat*.



Rysunek 3.8. System operacyjny DR-DOS gotowy do pracy

Każdy system operacyjny ma niepowtarzalną strukturę warstwową. Im bardziej skomplikowany system, tym więcej warstw, które obsługują coraz to nowsze możliwości systemu operacyjnego.

Nowsze systemy operacyjne starają się przejąć pełną kontrolę nad sprzętem za pomocą odpowiednich sterowników, rezygnując tym samym z konieczności odwoływania się do BIOS-u, który wcześniej pośredniczył między systemem operacyjnym a sprzętem.

3.3.2. Rola systemu operacyjnego

Komunikacja użytkownika z komputerem

Za komunikację między systemem operacyjnym a użytkownikiem odpowiedzialna jest zewnętrzna warstwa systemu, nazywana **powłoką** (ang. *shell*), która umożliwia użytkownikowi uruchomienie aplikacji.

Zarządzanie plikami

Definicja

Plik to jednostka zapisu i przechowywania danych w komputerze.

Plik jest ciągiem bitów danych, opatrzonym nazwą i atrybutami.

Dane w pliku są zapisane w postaci kodu binarnego, który w zapisie heksadecymalnym pokazany jest na rysunku 3.9. Widzimy tam mały fragment pliku graficznego.

```
00 00 00 00 00 00 00 00
10 45 13 00 00 00 00 00
00 00 00 00 00 00 00 01
BD 99 C3 7E 00 00 00 00
FF DB FF C3 E7 FF 7E 00
00 00 44 EF FE FE FE 7C
04 00 00 00 10 38 7C FE
00 04 00 05 00 00 00 18
00 00 00 00 05 00 06 00
18 18 7E 00 00 00 00 06
3C 7E 7E 3C 00 00 00 00
FF FF FF C3 81 81 C3 77
00 00 00 00 3C 7E 66 42
09 00 0A 00 00 FF FF C3
FF 00 00 0A 00 0B 00 00
```

Rysunek 3.9. Fragment pliku binarnego w zapisie heksadecymalnym (szesnastkowym)

Nazwa pliku z reguły składa się z głównej części nazwy i tzw. rozszerzenia (oddzielonego kropką od nazwy). Długość nazwy pliku i zasady jej tworzenia zależą od zainstalowanego systemu operacyjnego oraz *systemu plików*.

Wskazówka

Mimo że niektóre systemy operacyjne pozwalają na stosowanie znaków diakrytycznych (specyficznych dla danego języka narodowego), jeśli to możliwe, unikaj ich w nazwach plików i folderów. W przypadku próby odczytania takiego pliku w systemie operacyjnym innym niż polskojęzyczny prawdopodobnie wystąpią problemy, ponieważ w miejscu znaków diakrytycznych (*ą, ę, ś, ć…*) pojawią się tzw. *krzaczki* w nazwie pliku (na przykład zamiast słowa *Łata* pojawi się *Ł?ata*).

Pliki, oprócz swojej nazwy, dodatkowo mają *atrybuty* — właściwości. Atrybut może oznaczać, że plik jest: *tylko do odczytu*, *ukryty*, *systemowy*, *archiwalny*. Niektóre systemy operacyjne pozwalają na *definicję praw dostępu do pliku* przez użytkownika oraz grupy użytkowników.

System operacyjny pozwala użytkownikowi na zarządzanie plikami, dając mu możliwość ich *tworzenia*, *kopiowania*, *przenoszenia*, *usuwania*, *zmieniania ich nazw i atrybutów*. W celu uporządkowania plików tworzymy strukturę katalogów (folderów), która umożliwia ich pogrupowanie.

Definicja

Katalog (ang. *directory*) to logiczna struktura organizacji danych na nośnikach pamięci. Programy i dokumenty są organizowane w katalogi, które mogą zawierać zarówno pliki, jak i dodatkowe katalogi (podkatalogi). Katalogi pozwalają więc katalogować dane, zamiast składować je bezpośrednio w *katalogu głównym systemu plików*. Katalogi umożliwiają uporządkowanie informacji zapisanych w plikach, ich tematyczne pogrupowanie oraz oddzielenie plików powstałych w różnych aplikacjach.

Definicja

Folder (ang. *folder*) to kontener na programy i pliki w graficznym interfejsie użytkownika (GUI), symbolizowany na ekranie za pomocą specjalnego elementu graficznego (ikony), takiego jak teczka lub szuflada.

Często określenia *folder* i *katalog* są używane zamiennie. Katalog jest fizycznie przechowywany na nośniku pamięci (na przykład dysku) i jest prezentowany jako folder. Jednak folder może być wirtualny i istnieć tylko w obrębie danego systemu operacyjnego. Przykładami takich folderów w systemie Windows są okno *Mój komputer* czy *Panel sterowania*. Foldery te nie istnieją na dysku, a jedynie w systemowym rejestrze.

Definicja

GUI (ang. *Graphical User Interface* — graficzny interfejs użytkownika) — sposób komunikacji użytkownika z komputerem za pomocą elementów graficznych (na przykład: pulpitu z ikonami, kierowanego myszą kursora, okien dialogowych). Obecnie GUI stanowi podstawę większości nowoczesnych systemów operacyjnych i często nazywany jest środowiskiem graficznym. Obsługa GUI jest intuicyjna i interaktywna.

Systemy plików

Bez umiejętności posługiwania się systemem plików trudno mówić o skutecznym użytkowaniu komputera. Dla zwykłej osoby system operacyjny jawi się niemal wyłącznie od strony jego systemu plików. Pliki są tworzone i usuwane, czytane i zapisywane, a niektóre z nich (programy binarne) — ładowane do pamięci i wykonywane. Każdy system plików ma własny interfejs (tekstowy lub graficzny), za pomocą którego zleca się wykonywanie działań na plikach i katalogach. W systemach Windows rolę interfejsu plikowego odgrywa tzw. *zarządca plików* (ang. *file manager*).

Definicja

System plików (ang. *file system*) określa, w jaki sposób informacje są zapisywane i odczytywane. System plików definiuje także wielkość *klastrów*, możliwe do użycia atrybuty plików oraz schemat poprawnych nazw plików i katalogów, określając ich długość i dopuszczalne znaki w nazwach.

Definicja

Klaster (ang. *cluster*, *allocation unit*; w niektórych systemach plików na przykład FAT, NTFS) to określona dla danego nośnika podstawowa, najmniejsza jednostka przechowywania danych, składająca się z jednego lub kilku *sektorów*. Obszar jednego klastra można wypełnić tylko jednym plikiem, nawet jeśli będzie on wypełniał klaster tylko w niewielkiej części.

Definicja

Sektor to najmniejsza fizyczna jednostka zapisu danych na dyskach twardych, dyskietkach i innych nośnikach, na przykład kartach pamięci. Sektor jest zapisywany i czytany zawsze w całości. Wielkość sektora wynosi 2^9, czyli 512 bajtów. Sektory są tworzone podczas procesu formatowania.

Najpopularniejsze systemy plików to:
- FAT16 (stosowany w MS-DOS, Windows 3.1, Windows 9x),
- FAT32 (Windows 95 OSR i Windows 98),
- NTFS (Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7),
- Ext3, ReiserFS oraz Xfs (Linux).

Na płycie CD dołączonej do podręcznika, w pliku *systemy plikow.pdf*, zamieszczona została krótka charakterystyka wybranych systemów plików. Znajdują się tam opisy i porównania systemów plików: FAT, NTFS, Ext2 oraz Ext3.

Zarządzanie zasobami komputera

System operacyjny obsługuje urządzenia wchodzące w skład komputera. Podstawowe zadania w tym zakresie to:
- obsługa wewnętrznych i zewnętrznych elementów komputera, takich jak: procesor, płyta główna, pamięć lub drukarka, skaner itp.;
- badanie stanu urządzeń oraz poprawności komunikacji;
- udostępnianie urządzeń aplikacjom, a tym samym osobie użytkującej (na przykład udostępnienie drukarki w edytorze tekstu, skanera w programie graficznym itp.);
- chwilowe i trwałe podłączanie urządzeń do komputera i odłączanie ich od niego.

Zasoby

Definicja

Zasobem w systemie operacyjnym jest każdy rodzaj sprzętu komputerowego lub danych, niezbędny do prawidłowego wykonywania oraz zakończenia istniejących procesów.

Główne zasoby w systemie operacyjnym to:
- procesor (procesory),
- pamięć (operacyjna, na przykład RAM; zewnętrzna, na przykład HDD),

- urządzenia wejścia-wyjścia (mysz, klawiatura, drukarka),
- dane.

Zarządzanie zasobami systemu komputerowego związane jest z:
- przydziałami zasobów (planowanie obciążenia pamięci, czasu dostępności procesora),
- synchronizacją dostępu do zasobów (kolejność wykonywania procesów, z uwzględnieniem ich ważności),
- ochroną i autoryzacją dostępu do zasobów,
- odzyskiwaniem zasobów,
- rozliczeniem wykorzystywanych zasobów.

Kryteria oceny systemu operacyjnego

Każdy system operacyjny powinien spełniać wymienione niżej kryteria.

Kryterium efektywności:
- szybkość wykonania aplikacji — wykonanie aplikacji najszybciej, jak to jest możliwe;
- maksymalna liczba realizowanych procesów w jednostce czasu;
- równomierne zużycie zasobów — efektywny przydział mocy obliczeniowej CPU (procesora) oraz I/O (urządzeń wejścia-wyjścia) aplikacjom, które potrzebują dużo zasobów CPU lub wykonują wiele operacji I/O.

Kryterium oszczędności:
- pamięci — system operacyjny zużywa jak najmniej pamięci;
- dysków — system operacyjny nie zapełnia dysków.

Kryterium jakości:
- żywotność — system operacyjny nie ulega awarii;
- łatwość naprawy — system operacyjny można łatwo naprawić.

Stopień, w jakim udało się spełnić te kryteria, decyduje o popularności i efektywności danego systemu operacyjnego.

Podział systemów operacyjnych

Systemy operacyjne można podzielić ze względu na wiele charakterystycznych czynników. Z punktu widzenia użytkownika jednym z najważniejszych kryteriów jest liczba jednocześnie wykonywanych zadań. Rozróżniamy:
- *systemy jednozadaniowe* — niedopuszczalne jest rozpoczęcie wykonywania następnego zadania użytkownika przed zakończeniem poprzedniego (na przykład DOS, CP/M);
- *systemy wielozadaniowe* — dopuszczalna jest realizacja jednocześnie wielu zadań (procesów), którym kolejno przydzielany jest procesor. Zwolnienie procesora następuje w wyniku jednej z poniższych operacji: żądania przydziału dodatkowego zasobu, zainicjowania operacji wejścia-wyjścia, przekroczenia ustalonego limitu czasu. Systemy wielozadaniowe nazywa się czasami systemami z podziałem czasu (ang. *time-sharing systems*, na przykład: Linux, Windows NT, OS/2).

Bardzo ważny podział systemów operacyjnych oparty jest na możliwości jednoczesnego korzystania z systemu przez kilku użytkowników. Rozróżniamy więc:

- *systemy dla jednego użytkownika* — zasoby systemu przeznaczone są dla jednego użytkownika (na przykład w przypadku komputerów osobistych), nie ma mechanizmów autoryzacji dostępu, a mechanizmy ochrony informacji są ograniczone;
- *systemy wielodostępne* — z zasobów systemu może korzystać wielu użytkowników, a system gwarantuje ochronę informacji przed niepowołaną ingerencją, na przykład: Unix, Solaris oraz sieciowe systemy operacyjne, takie jak Novell NetWare, Microsoft Windows SBS (Small Business Server) i Linux.

Inne typy systemów operacyjnych to:

- *rozproszone systemy operacyjne* — w zadaniach bierze udział więcej niż jeden procesor, procesory znajdują się w różnych maszynach — są rozproszone, komunikacja odbywa się przez różne łącza komunikacyjne (na przykład Mach, Chorus);
- *systemy równoległe* — w zadaniach bierze udział wiele procesorów, które mogą równolegle wykonywać obliczenia. Procesory te mogą być ze sobą ściśle powiązane, jeżeli współużytkują niektóre pozostałe składowe systemu komputerowego (magistralę, pamięć, urządzenia zewnętrzne itp.). Mogą też mieć własną pamięć i magistralę i tworzyć niezależny podsystem komputerowy, który komunikuje się z pozostałymi poprzez szybkie magistrale. A wszystko to dla uzyskania maksymalnego bezpieczeństwa, zwiększenia szybkości i niezawodności działania systemu (na przykład Tandem, Sun Solaris 2);
- *systemy czasu rzeczywistego RTOS* (ang. *Real Time Operating Systems*) — główną cechą jest przetwarzanie informacji w czasie rzeczywistym lub zbliżonym do rzeczywistego. W bardzo ogólnej klasyfikacji systemów operacyjnych wyróżnia się ich dwie klasy — sterowane zdarzeniami albo czasem. W przypadku pierwszych istnieje ryzyko przeciążenia systemu, które może być spowodowane dużą liczbą napływających danych zewnętrznych i koniecznością ich przetwarzania. Systemy operacyjne sterowane czasem działają w rytmie zegara czasu rzeczywistego i nie są podatne na tego typu przeciążenia, co powoduje, że stosowane są one między innymi w tych gałęziach przemysłu, w których wymagane jest zachowanie wysokiej niezawodności systemu przy dużych ilościach przetwarzanych informacji. Jednym z najpopularniejszych komercyjnych systemów RTOS jest QNX firmy QNX Software Systems.

Część systemów operacyjnych obsługiwana jest w środowisku tekstowym — użytkownik musi wykazać się znajomością szeregu poleceń, dzięki którym może komunikować się z systemem operacyjnym.

Wiele systemów operacyjnych udostępnia środowiska graficzne ułatwiające ich obsługę.

Krótki przegląd systemów operacyjnych

Różnorodność systemów operacyjnych jest bardzo duża. Istniejące systemy są ciągle rozbudowywane i modyfikowane. Powstają ich nowe wersje, które oferują nowe możliwości. Często można spotkać się z jednoznacznym stwierdzeniem, że dany system operacyjny

jest lepszy od drugiego. Być może, jednak takie stwierdzenie pod wieloma względami wydaje się bardzo niesprawiedliwe, a czasami można je potraktować jako po prostu zasłyszane i powtarzane. Nie można bowiem porównywać tak odmiennych produktów. Porównanie to może jedynie dotyczyć konkretnych możliwości i zastosowań.

System operacyjny powinien być przystosowany do konkretnego sprzętu komputerowego, jego procesora i płyty głównej.

Na płycie CD dołączonej do podręcznika, w pliku *systemy operacyjne.pdf*, zamieszczony został krótki przegląd wybranych systemów operacyjnych: MS-DOS, Linux, Windows 3.11, Windows 9x, Windows 2000/NT, Windows XP, Windows Vista i Windows 7.

3.3.3. Maszyna wirtualna

Od kilku lat dynamicznie rozwijana jest koncepcja maszyn wirtualnych.

Ideą maszyny wirtualnej jest utworzenie logicznego komputera udostępniającego w odpowiedni sposób fizyczne zasoby maszyny podstawowej. Tak zadeklarowana maszyna jest w pełni funkcjonującym komputerem wraz z systemem operacyjnym i oprogramowaniem na nim zainstalowanym. Oto zalety i możliwości takiej wirtualizacji:

- poznawanie nowych systemów operacyjnych;
- testowanie oprogramowania pracującego w różnych systemach operacyjnych;
- niezawodne odtwarzanie systemów po awariach;
- niemal natychmiastowy *backup* (tworzenie kopii bezpieczeństwa) maszyn wirtualnych;
- maszyny takie nie wymagają modyfikowania systemu operacyjnego uruchamianego wewnątrz maszyny wirtualnej, instalacja przebiega tak jak na komputerze fizycznym i z tych samych nośników;
- konsolidacja wielu serwerów — redukcja kosztów;
- łatwiejsze zarządzanie serwerami;
- łatwa rozbudowa (na przykład zwiększenie przestrzeni dyskowej);
- serwery są nadal logicznie odseparowane;
- instalacja nowego serwera nie wymusza zakupu sprzętu;
- lepsze wykorzystanie mocy obliczeniowej.

W poniższych ćwiczeniach użyjemy oprogramowania do wirtualizacji programu VMware Player. Pliki niezbędne do ich wykonania zamieszczono na płycie CD dołączonej do podręcznika.

Maszyna wirtualna z systemem operacyjnym DR-DOS

Ćwiczenie 3.1. Przeprowadź proces instalacji oprogramowania maszyny wirtualnej, który polega na:

a) zainstalowaniu programu VMware Player i utworzeniu maszyny wirtualnej, na której zostanie zainstalowany system operacyjny DR-DOS;

b) utworzeniu *aktywnej partycji* na dysku twardym, *sformatowaniu tej partycji (C)* z jednoczesnym *przeniesieniem systemu operacyjnego* na aktywną partycję.

Na płycie CD dołączonej do podręcznika, w pliku *cwiczenie3.1.pdf*, zamieszczony został zilustrowany zrzutami ekranowymi szczegółowy opis wykonania tego ćwiczenia.

Ćwiczenie 3.2. Zainstaluj program FreePascal na maszynie wirtualnej.

Jest to darmowe oprogramowanie, które umożliwia tworzenie i uruchamianie programów napisanych w języku Pascal. Ponieważ program ten wymaga instalacji z kilku dyskietek, wszystkie pliki instalatora FreePascala zostały zgromadzone w obrazie płyty CD (*.iso*). W standardowej wersji DOS-u nie ma możliwości obsługi CD-ROM-u. Aby uzyskać do niego dostęp, należy zainstalować specjalne sterowniki. Obraz dyskietki *DRDOSfree5a.img* zawiera już obsługę CD-ROM-u.

Na płycie CD dołączonej do podręcznika, w pliku *cwiczenie3.2.pdf*, zamieszczony został szczegółowy opis wykonania tego ćwiczenia.

Podczas wykonywania powyższych ćwiczeń poznaliśmy kilka poleceń systemu operacyjnego DOS: `fdisk`, `format`, `cd`, `md`, `dir`. Dwa pierwsze to *polecenia zewnętrzne*, co oznacza, że do ich prawidłowego działania konieczna jest obecność odpowiednich programów. Programy te znajdowały się w obrazie dyskietki *DRDOSfree5a.img*. Kolejne trzy to *polecenia wewnętrzne*, rozpoznawane i wykonywane przez interpreter poleceń, czyli plik *command.com*, który znajduje się w katalogu głównym partycji *C*. Można go zobaczyć (rysunek 3.10) za pomocą polecenia `dir` wydanego z poziomu katalogu głównego. Powróć do niego z katalogu *GO32V2*, wydając polecenie `cd \`.

```
C:\>dir

 Volume in drive C does not have a label
 Directory of  C:\

DRBIO     SYS         35,556   7-23-09   4:40p
DRDOS     SYS         36,597   7-23-09   4:40p
COMMAND   COM         64,495   7-23-09   4:40p
CWSDPMI   SWP              0  11-06-10   9:42a
PP             <DIR>          11-06-10   9:32a
        5 File(s)    509,534,208 bytes free

C:\>_
```

Rysunek 3.10. Wyświetlenie zawartości katalogu głównego dysku C

Wykorzystanie plików przetwarzania wsadowego

Pod koniec procesu startu systemu operacyjnego DOS uruchamiane są pliki przetwarzania wsadowego *config.sys* oraz *autoexec.bat*. Pierwszy z tych plików może konfigurować pamięć RAM komputera, drugi z kolei daje możliwość konfiguracji środowiska pracy (można zmienić sposób zgłaszania systemu, uruchomić sterowniki, na przykład obsługi CD-ROM-u, ustalić ścieżki przeszukiwań).

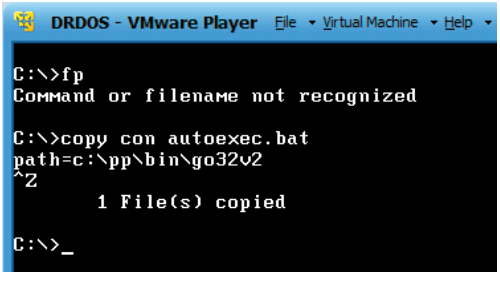
Co to są ścieżki przeszukiwań? Gdy wprowadzimy jakieś polecenie do wykonania, na przykład `fp`, w pierwszej kolejności polecenie to kierowane jest do interpretera poleceń (pliku *command.com*). Jeśli zostanie rozpoznane jako komenda wewnętrzna, nastąpi uruchomienie tego polecenia systemowego. Jeśli nie, polecenie to będzie traktowane jako chęć uruchomienia pliku o nazwie *fp*. Plik ten będzie poszukiwany w aktualnie otwartym katalogu. Dlatego też, gdy otwarty jest katalog główny (*C:*), po wpisaniu polecenia `fp` zostanie wyświetlony następujący komunikat: *Command or filename not recognized* (polecenie lub nazwa pliku nie są rozpoznawane). Co zrobić, aby po wprowadzeniu polecenia `fp` z dowolnej lokalizacji został uruchomiony program FreePascal? Należy zadeklarować ścieżkę przeszukiwań wskazującą systemowi operacyjnemu, gdzie ma szukać

nazw plików, które chcemy uruchomić. Najkorzystniejszym rozwiązaniem będzie utworzenie pliku *autoexec.bat* i zdefiniowanie w nim ścieżki przeszukiwań, czyli wskazanie miejsca na dysku, w którym znajduje się plik *fp.exe*.

Ćwiczenie 3.3. Utwórz w katalogu głównym dysku *C* plik wsadowy *autoexec.bat* i zdefiniuj w nim ścieżkę przeszukiwań do pliku *fp.exe*.

Polecenie `copy con nazwa_pliku` kopiuje z konsoli (klawiatury) tekst, który zostanie zapisany w pliku pod podaną nazwą. W tym ćwiczeniu będzie to plik *autoexec.bat*.

Teraz wpisz ścieżkę przeszukiwań, czyli `path=c:\pp\bin\go32v2`, i przejdź do nowej linii. Aby zakończyć edycję pliku, należy wpisać znak końca pliku `^Z` (*Ctrl+Z* lub *F6*) i zatwierdzić klawiszem *Enter*. (Proces ten pokazano na rysunku 3.11). W tym momencie zostanie utworzony plik *autoexec.bat* z zawartością podaną z klawiatury. Sprawdź, czy to już działa. Wpisz `fp`, aby uruchomić program FreePascala. Niestety, zmiany nie zostały jeszcze uwzględnione. Dopiero po ponownym uruchomieniu komputera wirtualnego (*Send Ctrl+Alt+Del*) zadziałają wprowadzone ustawienia związane z plikiem przetwarzania wsadowego. Wykonaj więc restart i uruchom FreePascal z poziomu katalogu głównego dysku *C*.



Rysunek 3.11. Tworzenie pliku przetwarzania wsadowego autoexec.bat

Pliki i katalogi w DOS-ie

Przez cały czas używamy pojęć: *plik*, *program*, *katalog*. Ich nazwy składają się z głównej części nazwy i rozszerzenia oddzielonego kropką. Może istnieć plik bez rozszerzenia. W DOS-ie nazwa może mieć od 1 do 8 znaków. Pliki można tworzyć, zmieniać ich nazwy, kopiować, usuwać, ustalać i zmieniać ich atrybuty.

`copy con nazwa_pliku` — to znana już metoda tworzenia plików tekstowych bezpośrednio z konsoli.

`rename stara_nazwa_pliku nowa_nazwa_pliku` — zmienia nazwę pliku.

`del nazwa_pliku` — kasuje plik (lub pusty katalog).

`copy co dokąd` — kopiuje plik do wskazanego miejsca.

`type` — wyświetla zawartość pliku. Z reguły używane jest do plików typu ASCII. Pliki binarne, a także pliki tworzone przez edytory tekstów, zawierające znaki sterujące, powodują, że treść pliku nie nadaje się do interpretacji.

Polecenia `del` i `copy` działają również na grupach plików, wykorzystując znaki uogólniające `*` lub `?` (ang. *wild card*).

Polecenia dotyczące katalogów:

- `md nazwa` — tworzy katalog w bieżącej lokalizacji lub podanej w postaci ścieżki poprzedzającej nazwę katalogu;

- `rd nazwa` — usuwa pusty katalog w bieżącej lokalizacji lub podanej w postaci ścieżki poprzedzającej nazwę katalogu;
- `cd nazwa_katalogu` — zmienia katalog bieżący na podany (można podać ścieżkę do katalogu). Szczególne katalogi mają swoje symboliczne nazwy: `\` oznacza znak katalogu głównego, `..` — znak katalogu nadrzędnego, `.` — znak katalogu bieżącego. Polecenie `dir` wyświetla na ekranie nazwy katalogów i plików pisane wielkimi literami, a dla ich odróżnienia przy nazwach katalogów wyświetla napis <*DIR*>;
- `tree` — to polecenie zewnętrzne, pokazujące strukturę drzewiastą katalogów. `tree /f` — oznacza strukturę wraz z plikami.

Do wyświetlenia lub zmiany *atrybutów pliku lub katalogu* służy polecenie zewnętrzne `attrib`. Podstawowe atrybuty plików w systemie operacyjnym DOS to:

r — plik tylko do odczytu,

a — plik archiwalny,

s — plik systemowy,

h — plik ukryty.

Atrybuty pliku w DOS-ie mogą być zmieniane:

+ — dodaje (ustanawia) atrybut,

– — zdejmuje (usuwa) atrybut.

Na przykład `attrib -r -s -h config.sys` usunie atrybuty *tylko do odczytu*, *systemowy* i *ukryty* pliku *config.sys*. Natomiast `attrib +r +s +h config.sys` doda wszystkie trzy atrybuty.

Ćwiczenie 3.4. Utwórz pokazaną na rysunku 3.12 strukturę drzewa katalogowego.

```
C:\>tree
      bytes  files  path
     12,512      1  c:\
          0      0  c:\moje
          0      0  c:\moje\pascal
          0      0  c:\moje\teksty
          0      0  c:\moje\zadania
          0      0  c:\moje\zadania\zadanie1
          0      0  c:\moje\zadania\zadanie2
          0      0  c:\moje\zadania\zadanie3
          0      0  c:\moje\grafika
          0      0  c:\moje\grafika\rastrowa
          0      0  c:\moje\grafika\wektorow
total files 1   total bytes 12,512   disk free space 535,298,048

C:\>_
```

Rysunek 3.12. Drzewo katalogów

Aby móc wydawać polecenia zewnętrzne (na przykład *tree*), uruchom komputer wirtualny ze stacji dyskietek (zamontuj obraz *DRDOSfree5a.img*), a następnie zmień napęd bieżący na *C*. W tym celu wybierz maszynę wirtualną DR-DOS i uruchom *Play virtual machine*. Kliknij prawym przyciskiem myszy ikonę dyskietki i wybierz *Settings…*, a następnie, korzystając z *Use floppy image file*, wskaż plik z obrazem *DRDOSfree5a.img*, który znajduje się na płycie CD. Aby rozruch komputera nastąpił z dyskietki systemowej, należy ponownie uruchomić komputer wirtualny. Z menu *Virtual Machine* wybierz zatem

Send Ctrl+Alt+Del. System operacyjny, uruchomiony z dyskietki, zgłosi się znakiem zachęty `A:\>`. Pozostaje jeszcze zmiana bieżącego dysku na dysk *C*.

Sprawdź zawartość katalogu głównego dysku *C* poleceniami `dir` oraz `tree`. Następnie zacznij tworzyć w katalogu głównym dysku *C* strukturę katalogów, zaczynając od korzenia drzewa, czyli od katalogu *MOJE*, poleceniem `md moje`. Nie ma żadnego komunikatu w stylu „katalog utworzono". Sprawdź ponownie zawartość katalogu poleceniami `dir` i `tree` — zobaczysz katalog *MOJE*. Ponieważ na dysku znajduje się już wcześniej utworzony katalog *PP*, możesz pisać polecenie `tree` z podaniem gałęzi drzewa, które ma być wyświetlone, napisz więc `tree moje`.

Wskazówka

W celu czytelności treści rozdziału nazwy katalogów zapisane zostały wielkimi literami. W tym miejscu należy jednak wyjaśnić, że w systemie operacyjnym DOS litery małe i wielkie nie są rozróżniane. Zatem *MOJE* i *moje* to ta sama nazwa.

Następne katalogi możesz tworzyć, wędrując po istniejącym drzewie katalogów. Zmień więc katalog bieżący na utworzony przed chwilą katalog *MOJE* poleceniem `cd moje`. Wszystkie *podkatalogi*, czyli *PASCAL*, *TEKSTY*, *ZADANIA*, *GRAFIKA*, będą w tej gałęzi, zatem utwórz je kolejno w tym poziomie drzewa. Nie można tego zrobić za jednym razem.

Utwórz pozostałe katalogi. Zmieniaj bieżący katalog, wędrując po drzewie katalogów lub pozostając w katalogu *MOJE*. Można bowiem również wydać polecenie `md zadania\zadanie1`, które utworzy podkatalog w katalogu *ZADANIA*.

Ćwiczenie 3.5. Utwórz pliki, których nazwy i położenie pokazuje rysunek 3.13.

```
C:\>tree /f
--a---       12,512   1-07-99   7:03a  c:\tree.com
--a---            7  11-01-10   5:17p  c:\moje\pascal\program.pas
--a---            6  11-01-10   5:18p  c:\moje\teksty\list.txt
--a---            9  11-01-10   5:18p  c:\moje\teksty\podanie.txt
--a---            6  11-01-10   5:19p  c:\moje\zadania\zadanie1\dane1.txt
--a---            6  11-01-10   5:20p  c:\moje\zadania\zadanie1\rozw1.txt
--a---            6  11-01-10   5:21p  c:\moje\zadania\zadanie2\dane2.txt
--a---            6  11-01-10   5:21p  c:\moje\zadania\zadanie2\rozw2.txt
--a---            7  11-01-10   5:21p  c:\moje\zadania\zadanie3\tresc3.txt
total files 9   total bytes 12,565   disk free space 535,232,512

C:\>_
```

Rysunek 3.13. Struktura katalogów wraz z plikami (tree /f)

program.pas — ten plik ma zawierać kod programu, który po uruchomieniu wypisze na ekranie czerwonymi literkami Twoje dane. Oto sposób jego wykonania:

Zmień katalog bieżący na *PASCAL* (`cd \moje\pascal`). Teraz użyj polecenia kopiowania z konsoli do pliku. Podaj treść pliku, a na końcu znak końca pliku `^Z` (*Ctrl+Z* lub klawisz *F6*).

```
C:\MOJE\PASCAL\copy con program.pas
program wizytowka;
uses crt;
begin
 textcolor (red);
 writeln ('Imie i nazwisko'); //podaj swoje dane
 readln;
end. ^Z
```

Sprawdź poleceniem `dir`, czy plik *program.pas* został utworzony. Za pomocą polecenia `type program.pas` zobacz jego zawartość. Dodatkowo uruchom program FreePascal (poleceniem `C:\pp\bin\go32v2\fp program.pas` z dowolnego miejsca na dysku) i sprawdź jego działanie.

Utwórz następne pliki zgodnie z lokalizacją i nazewnictwem podanymi w treści zadania. Zmieniaj katalogi robocze i wykonuj polecenia tworzenia plików. Napisz list, złóż podanie. W zadaniach napisz krótką treść, na przykład: `ile jest 2+2?` W rozwiązaniach podaj wyniki.

Istnieje też możliwość połączenia dwóch plików w nowy plik, zawierający treść plików łączonych. Zmień katalog roboczy na *ZADANIE1* i połącz plik *dane1.txt* z plikiem *rozw1.txt*: `copy dane1.txt+rozw1.txt razem1.txt`. Zobacz zawartość pliku *razem1.txt*. Zwróć uwagę na kolejność treści.

Możesz sprawdzać położenie plików w całym drzewie poleceniem `tree /f`, które wyświetli drzewo wraz z plikami, jak na rysunku 3.13. Istnieje również możliwość wyświetlenia tego drzewa bez konieczności przejścia do katalogu głównego dysku — za pomocą polecenia `tree \moje /f`, co należy rozumieć następująco: wyświetl strukturę wraz z plikami katalogu *MOJE*, znajdującego się w katalogu głównym bieżącego dysku.

Maszyna wirtualna z systemem operacyjnym Linux

Obecnie większość dystrybucji Linuksa ma przyjazne środowisko graficzne, w którym bardzo łatwo możemy się poruszać, klikając myszką. Jednak każdy Linux udostępnia, oprócz możliwości pracy w trybie graficznym, możliwość użycia trybu tekstowego (trybu konsoli). To właśnie tryb konsoli udostępnia całą siłę Linuksa. Dzięki umiejętności stosowania odpowiednich poleceń pisanych z klawiatury możemy konfigurować nawet bardzo zaawansowane usługi. W obsłudze systemu Linux niezbędna jest zatem znajomość tych poleceń.

Ćwiczenie 3.6. Utwórz drugą maszynę wirtualną z systemem operacyjnym Linux Arudius.

Dokładny opis wykonania ćwiczenia znajduje się na płycie CD dołączonej do podręcznika, w pliku *cwiczenie3.6.pdf*.

Pliki i katalogi w Linuksie

W systemie Linux małe i wielkie litery są rozróżniane, co oznacza, że polecenie napisane małymi literami będzie inaczej interpretowane niż polecenie napisane literami wielkimi.

W celu porównania pracy w obu systemach operacyjnych (DOS i Linux) spróbujemy wykonać podobne operacje na plikach i katalogach jak w DOS-ie, a wszystko to na systemie operacyjnym maszyny wirtualnej. Podobne polecenia występują również w innych systemach operacyjnych.

Zacznijmy od wyświetlenia zawartości (*listowania*) katalogu poleceniem `ls`. Polecenie to wyświetla listę, taką jak na rysunku 3.14. Jak widać, jest tam wyświetlona dokładnie jedna nazwa pliku *arudius_list.txt* oraz jedna nazwa katalogu *tools/*.

```
root@arudius:~# ls
arudius_list.txt  tools/
```

Rysunek 3.14. Lista nazw plików i katalogów po wydaniu polecenia ls

Polecenie `ls` wywołane bez specjalnych parametrów wyświetli tylko katalogi i pliki w obecnym katalogu. Polecenie `ls /usr` spowoduje wypisanie zawartości katalogu */usr*, jak na rysunku 3.15. Można też wyświetlić zawartość katalogu *tools*, wydając polecenie `ls tools`.

```
root@arudius:~# ls /usr
X11@    bin/   doc/                  include/  libexec/  sbin/   src/
X11R6/  com/   etc/                  info/     local/    share/  tmp@
adm@    dict/  i486-slackware-linux/ lib/      man/      spool@  var/
root@arudius:~#
```

Rysunek 3.15. Zawartość katalogu usr

Istnieje możliwość wyświetlenia i odczytania bardziej szczegółowych informacji. Do polecenia `ls` dodamy *przełącznik* `-l` (litera *l*, a nie liczba *1*), co spowoduje wyświetlenie danych w kolumnach, wraz ze szczegółami o zawartości. Polecenie `ls -l` dla katalogu *usr* będzie zapisane następująco: `ls /usr -l`, a efekt jego działania można zobaczyć na rysunku 3.16.

```
root@arudius:~# ls /usr -l
total 0
lrwxrwxrwx  1 root root     5 2006-03-15 23:10 X11 -> X11R6/
drwxr-xr-x  4 root root    42 2005-12-30 02:00 X11R6/
lrwxrwxrwx  1 root root     8 2006-03-15 23:10 adm -> /var/adm/
drwxr-xr-x  4 root bin  11758 2006-03-15 22:30 bin/
drwxr-xr-x  4 root root    13 2005-12-31 07:36 com/
drwxr-xr-x  4 root root     0 1993-11-25 22:40 dict/
drwxr-xr-x  4 root root    59 2006-03-15 22:30 doc/
drwxr-xr-x  4 root root    23 2005-11-19 06:20 etc/
```

Rysunek 3.16. Lista plików i katalogów ze szczegółowymi informacjami

W wykazie tym otrzymujemy następujące informacje:

Prawa dostępu	Liczba powiązań	Właściciel	Grupa przypisana do pliku	Rozmiar	Data i godzina modyfikacji	Nazwa elementu
drwxr-xr-x	4	root	root	42	2005-12-30 02:00	X11R6

Szczegółowo omówimy **prawa dostępu**.

Pierwsza kolumna, zawierająca uprawnienia, zawsze ma długość **dziesięciu** znaków. Pierwszy znak może przyjąć następujące wartości:

- `d` — katalog (ang. ***d**irectory*),
- `-` — zwykły plik,
- `b` — specjalny plik blokowy (ang. ***b**lock*),

`c` — specjalny plik znakowy (ang. ***c**haracter*),

`l` — link symboliczny (ang. *symbolic **l**ink*),

`p` — nazwany potok (ang. *named **p**ipe*),

`s` — gniazdo (ang. ***s**ocket*),

`?` — inny, nieznany rodzaj pliku.

Kolejne dziewięć znaków tworzy trzy równe grupy. Pierwsza grupa to *zbiór praw dla właściciela pliku*, druga to *prawa dla grupy*, a trzecia — dla *pozostałych użytkowników systemu*. Literki w każdej grupie mają przypisane znaczenia, na przykład pierwsza grupa dla katalogu *X11R6* to ***rwx***. Każda literka oznacza jeden z wymienionych wcześniej atrybutów. Pierwsza to prawo odczytu — ***r**ead*, druga to prawo zapisu — ***w**rite*, a trzecia to atrybut wykonywalności — *e**x**ecute*. Każda literka ma przypisane miejsce. Na przykład *r* nie może się znajdować na trzecim miejscu, a *x* na drugim. Jeśli odebraliśmy któreś prawo, to zamiast odpowiadającej mu literki występuje znak -. Przykładowo w katalogu *X11R6* druga oraz trzecia grupa mają postać: *r–x*. Oznacza to, że katalog ma nadany atrybut odczytu i wykonywania, natomiast zapis — dla grupy i dla pozostałych użytkowników systemu — jest niemożliwy.

Oprócz notacji literowej (zwanej zapisem flag) istnieje jeszcze zapis liczbowy. Jego podstawą są cztery liczby, które mają odzwierciedlenie w zapisie literowym. W tabeli 3.1 przedstawiono podstawowe oznaczenia uprawnień do pliku.

Tabela 3.1. Oznaczenia uprawnień do pliku

Atrybut pliku (prawa)	Symbol	Liczba
Wykonywanie	*x*	1
Zapis	*w*	2
Odczyt	*r*	4
Brak uprawnień	-	0

Zapis liczbowy określający uprawnienia do pliku stosuje się częściej, gdyż jest bardzo zwięzły. Liczby określające poszczególne prawa zostały tak dobrane, aby jednoznacznie określały wszystkie uprawnienia użytkownika.

Ćwiczenie 3.7. Podaj w zapisie liczbowym uprawnienia do katalogu *X11R6* zapisane w postaci symbolicznej *rwxr-xr-x*.

Uprawnienia dla właściciela	rwx	4+2+1 = 7
Uprawnienia dla grupy	r-x	4+0+1 = 5
Uprawnienia dla innych	r-x	4+0+1 = 5

Uprawnienia, które w zapisie literowym oznaczone zostały *rwxr-xr-x*, w zapisie liczbowym przyjmą wartość *755*.

Odczytanie zapisu liczbowego jest jednoznaczne:

7 = 4+2+1 *rwx*

6 = 4+2+0 *rw-*

5 = 4+0+1 *r-x*

4 = 4+0+0 *r--*

3 = ...

2 = ...

1 = ...

Oto kilka przykładów nadanych praw do plików i katalogów:

`drw-------`	600	Tylko właściciel ma prawo do odczytu i zapisu.
`-rw-r--r--`	644	Właściciel ma prawo do zapisu i odczytu, a reszta — tylko prawo odczytu.
`-rw-rw-rw-`	666	Wszyscy mają prawo do odczytu i zapisu.
`drwx------`	700	Tylko właściciel ma prawo do odczytu, zapisu, uruchomienia.
`-rwxrwxrwx`	777	Wszyscy mają pełne prawa (niezalecane).
`-rwx--x--x`	711	Wszystkie prawa ma właściciel, reszta — tylko prawo uruchomienia.
`-rwxr-xr-x`	755	Właściciel ma wszystkie prawa do pliku, reszta — tylko prawo do odczytu i uruchomienia.
`drwxr--r--`	744	Właściciel ma pełne prawa do katalogu, reszta — prawo do odczytu.

Na rysunku 3.14 zobaczyliśmy bardzo skromną, dwuelementową listę. Czy to jest faktycznie cała zawartość bieżącego katalogu? Czy być może są tam jakieś ukryte zasoby? Polecenie `ls` z przełącznikiem `-a` pozwala na wyświetlenie zasobów z atrybutem *ukryty*. Wpiszmy więc `ls -a`. Na rysunku 3.17 pokazano zbiór zasobów, których nazwy poprzedza kropka. Oznacza ona nadanie atrybutu *ukryty*.

```
root@arudius:~# ls
arudius_list.txt  tools/
root@arudius:~# ls -a
.Xauthority     .fullcircle/      .gnupg/     .nessusrc.cert  .xinitrc
.airsnortrc     .gconf/           .icons/     .qt/            arudius_list.txt
.bash_history   .gconfd/          .links/     .rnd            tools/
.esd_auth       .gnome/           .mozilla/   .scapy_history
.ethereal/      .gnome2/          .msf/       .siegerc
.fluxbox/       .gnome2_private/  .nessusrc   .ssh/
root@arudius:~#
```

Rysunek 3.17. Lista katalogów i plików wraz z ukrytymi katalogami i plikami

Wróćmy teraz do praw dostępu do plików i katalogów. Czy można je nadawać i jak je zmienić? Oczywiście można posłużyć się graficznym menedżerem plików, ale jak to zrobić w trybie tekstowym? Polecenie `chmod` pozwala dodawać i odbierać atrybuty właścicielowi, grupie i pozostałym użytkownikom. Wystarczy poznać składnię tego polecenia:

`chmod` *`kategoria operatory`*`[+/-/=]` *`uprawnienia nazwa_pliku`*

Opis kategorii użytkowników:

`u` — użytkownik (ang. ***u**ser*),

`g` — grupa (ang. ***g**roup*),

`o` — inni (ang. ***o**thers*),

`a` — wszyscy (ang. ***a**ll*).

Operatory:

`+` — dodajemy uprawnienia,

`-` — odbieramy uprawnienia,

`=` — zmieniamy uprawnienia — przypisujemy takie, jakie podamy.

Przykłady użycia:

`chmod u=rwx,g+rw,o-r plik.txt` — to polecenie wykona następujące operacje na pliku *plik.txt*: właścicielowi nada prawo do odczytu, zapisu oraz wykonania, grupie doda prawo odczytu i zapisu, a innym odbierze prawo odczytu.

`chmod a+w plik.txt` — wszyscy mogą edytować *plik.txt*.

`chmod o-x plik.txt` — odbieramy możliwość wykonania *plik.txt* przez pozostałych użytkowników.

`chmod go+rx plik.txt` — pozostali użytkownicy oraz grupa będą mogli tylko czytać i wykonywać *plik.txt*.

Podczas zmian praw dostępu można posługiwać się zapisem liczbowym.

Ćwiczenie 3.8. Utwórz katalog *NOWE* i ustal następujące prawa dostępu do jego zawartości: *drwxrwxr-x*. Podaj zapis literowy i liczbowy nadawanych praw.

Najpierw utwórz katalog *NOWE* poleceniem `mkdir nowe`, które tworzy katalog o podanej nazwie. Zobacz zawartość bieżącego katalogu poleceniem `ls`. Następnie sprawdź poleceniem `ls -l` (litera *l*, a nie liczba *1*), jakie prawa dostępu zostały przydzielone do tego katalogu. Powinny być takie jak na rysunku 3.18.

```
root@arudius:~# ls -l
total 8
-rw-r--r--  1 root root 7893 2006-03-05 19:08 arudius_list.txt
drwxr-xr-x  2 root root   60 2008-05-11 22:57 nowe/
drwxr-xr-x  4 root root  192 2006-03-05 18:43 tools/
root@arudius:~#
```

Rysunek 3.18. Prawa dostępu do utworzonego katalogu *NOWE*

Katalog *NOWE* ma ustanowione prawa dostępu *drwxr-xr-x*, a zatem prawo do zapisu ma tylko administrator, czyli `root`. Zgodnie z poleceniem w ćwiczeniu należy dodać prawo zapisu dla grupy. Będzie to zatem dodanie prawa *w* dla grupy *g*, czyli: `chmod g+w nowe`.

A jak zapisać to w postaci liczbowej? Rozbij zapis praw *drwxrwxr-x* na grupy:

	u	g	o
d	rwx	rwx	r-x
	421	421	401
	7	7	5

Polecenie zmiany będzie zatem wyglądało tak: `chmod 775 nowe`. Aby zobaczyć zmiany, wydaj polecenie `ls -l`.

Jeśli masz ochotę, wykonaj jeszcze inne zmiany na katalogu *NOWE*, na przykład odbierz prawa wykonywania grupie i wszystkim pozostałym użytkownikom.

W ćwiczeniu 3.8 wykorzystano polecenie tworzenia nowego katalogu. Oto inne polecenia umożliwiające operacje na katalogach:

`mkdir` — tworzy nowy katalog w bieżącej lub w podanej lokalizacji.

`rmdir` — usuwa katalog.

`cd` — zmienia katalog bieżący.

`pwd` — informuje o aktualnym położeniu, o aktualnym katalogu.

`tree` — wyświetla drzewo katalogowe bieżącego lub wskazanego katalogu.

Ćwiczenie 3.9. Usuń katalog *NOWE*, który powstał w poprzednim ćwiczeniu. Następnie utwórz drzewo katalogów pokazane na rysunku 3.19. Zwróć uwagę, że podkatalogi *SCHOWEK* i *KATALOG* mają atrybuty *ukryty*. Nazwę tworzonego katalogu poprzedź kropką. Jest to jednoczesne nadawanie atrybutu i tworzenie katalogu. Do wędrowania po drzewie użyj polecenia `cd` oraz `cd ..` — do przejścia o poziom wyżej (do katalogu nadrzędnego). W razie problemów spójrz na rysunek 3.20, który pokazuje cały tok postępowania.

```
root@arudius:~# tree inform_eu
inform_eu
├── cw1
├── cw2
└── kat1

3 directories, 0 files
root@arudius:~# tree inform_eu -a
inform_eu
├── .schowek
├── cw1
├── cw2
│   └── .katalog
└── kat1

5 directories, 0 files
```

Rysunek 3.19. Drzewo katalogów

```
root@arudius:~# mkdir inform_eu
root@arudius:~# cd inform_eu
root@arudius:~/inform_eu# mkdir .schowek
root@arudius:~/inform_eu# mkdir cw1
root@arudius:~/inform_eu# mkdir cw2
root@arudius:~/inform_eu# mkdir kat1
root@arudius:~/inform_eu# cd cw2
root@arudius:~/inform_eu/cw2# mkdir .katalog
root@arudius:~/inform_eu/cw2# cd ..
root@arudius:~/inform_eu# cd ..
```

Rysunek 3.20. Tok postępowania przy tworzeniu drzewa katalogów

Ćwiczenie 3.10. Korzystając z edytora `vi`, utwórz w powstałej strukturze katalogów pliki tekstowe i nadaj plikom i katalogom prawa dostępu:

~/inform_eu/kat1/podanie.txt — podanie o rezerwację terminu na studniówkę,

~/inform_eu/cw2/list.txt — pozdrowienia z wycieczki,

~/inform_eu/.schowek/paczka.txt — opis zawartości wysłanej wczoraj paczki.

Wykorzystaj do tego celu systemowy edytor `vi` — więcej informacji (w języku polskim) na jego temat można znaleźć na stronie: *http://newbie.linux.pl/wydruk.php?wydruk=208&show=artykul*. Edytor `vi` ma dwa tryby pracy: *tryb poleceń* i *tryb edycji*. W trybie poleceń wydajesz polecenia określające, co chcesz zrobić, w trybie edycji wpisujesz tekst do pliku, który aktualnie edytujesz. **Uwaga!** Po uruchomieniu edytora vi jesteś w trybie poleceń (komend). Przejście do trybu edycji następuje po naciśnięciu klawisza *i* lub *Insert*. Przejście do trybu komend następuje po naciśnięciu klawisza *Esc*.

`vi nazwa_pliku` — to składnia polecenia, które utworzy plik i uruchomi jego edycję.

Oto dwa podstawowe polecenia niezbędne do zakończenia pracy w edytorze `vi` (po naciśnięciu *Esc*):

`:w` — zapis pliku,

`:q` — wyjście z programu.

Zachęcam serdecznie do pracy z tym jakże ciekawym edytorem. Potraktuj to zaproszenie jako wyzwanie. Nie zniechęcaj się.

Po utworzeniu plików sprawdź poleceniem `du`, ile miejsca na dysku zajmuje katalog *INFORM_EU* ze wszystkimi podkatalogami i plikami.

Nadaj plikom w katalogu *INFORM_EU* następujące prawa dostępu:

podanie.txt — każdy z grupy może go czytać, właściciel może czytać i pisać,

list.txt — nikt nie może pisać,

list.txt — brak praw dla pozostałych użytkowników,

cw2 — odbierz prawo do wykonywania dla wszystkich,

kat1 — odbierz prawo do czytania dla wszystkich.

3.3.4. Zadania

Zadanie 3.1. W katalogu *FOTO* zamierzasz udostępnić zdjęcia. Jakie prawa nadasz temu katalogowi, aby wszyscy zainteresowani mogli obejrzeć zdjęcia? Nie pozwól jednak nikomu usuwać ani modyfikować fotografii.

Zadanie 3.2. Odpowiedz na poniższe pytania kontrolne typu prawda – fałsz.

1. Powłoka systemu operacyjnego to interfejs użytkownika.
2. Procedura POST jest wykonywana podczas wysyłania poczty elektronicznej.

3. Terminy *hardware* i *software* to określenia dotyczące twardych i miękkich dysków (dyskietek).
4. Magistrala *PCI Express* jest odpowiedzialna za przesyłanie danych do pamięci RAM.
5. Plik *autoexec.bat* to plik wsadowy.
6. Ustawienia zapisane w pamięci BIOS można modyfikować.
7. Wielodostępność systemu operacyjnego to inaczej wielozadaniowość tego systemu.
8. Graficzny interfejs użytkownika GUI umożliwia pracę z wykorzystaniem urządzenia wskaźnikowego, na przykład myszki.
9. Port LPT umożliwia podłączenie zewnętrznych napędów optycznych.
10. Katalogi i foldery mają strukturę drzewiastą.
11. System plików FAT16 umożliwia tworzenie plików o nazwach składających się maksymalnie z 256 znaków.
12. Plik może mieć nadany jeden atrybut, na przykład: archiwalny, ukryty, systemowy lub do odczytu.
13. Atrybuty plików we wszystkich systemach operacyjnych są takie same.
14. Komputer może wczytać system operacyjny wyłącznie z dysku twardego.
15. Port USB zastąpił przestarzałe interfejsy COM oraz LPT.
16. Pamięć RAM przechowuje zapisane w niej informacje nawet po wyłączeniu zasilania.
17. System 64-bitowy może być zainstalowany na dowolnym komputerze.

3.4. Sieci komputerowe

Sieci komputerowe, podobnie jak systemy operacyjne, rozwijają się bardzo dynamicznie. Wciąż słyszymy o nowych technologiach, nowych sposobach przekazywania danych, o nowych protokołach i narzędziach. Większość z nas na co dzień spotyka się z sieciami komputerowymi. Internet jest dziś tak powszechny jak kiedyś radio czy telewizja. Nie będzie też wielkim odkryciem stwierdzenie, że internet nie istniałby, gdyby nie sieć komputerowa. Coraz popularniejsze stają się też nowe technologie, jak sieci bezprzewodowe, Bluetooth czy sieci komórkowe. Znajomość zagadnień leżących u podstaw projektowania i wykorzystywania sieci komputerowych jest przydatna każdemu, kto chce być na bieżąco z nowoczesnymi technologiami. W tym podrozdziale znajdziesz podstawowe informacje o sieciach komputerowych.

3.4.1. Podział sieci komputerowych

Definicja

Sieć komputerowa jest zbiorem sprzętu i oprogramowania współpracujących ze sobą w celu wymiany danych lub współużytkowania zasobów.

Wymianę danych lub współużytkowanie zasobów przeprowadzają *protokoły komunikacyjne*, będące *zbiorem reguł* pozwalających na komunikowanie się. Zbiór tych reguł jest akceptowany przez obie strony wymiany danych.

Podział ze względu na zasięg

Sieci komputerowe można podzielić ze względu na rozległość. Są to sieci:

- *LAN* (ang. *Local Area Networks*) — lokalna, wewnętrzna sieć obejmująca najczęściej obszar pracowni, szkoły, jednego budynku lub z kilku budynków. Jest to najmniej rozległa postać sieci komputerowej.
- *MAN* (ang. *Metropolitan Area Networks*), zwana siecią miejską, to duża sieć komputerowa, której zasięg obejmuje aglomerację lub miasto. Tego typu sieci używają najczęściej połączeń światłowodowych do komunikacji pomiędzy wchodzącymi w jej skład rozrzuconymi sieciami LAN. Przykładem sieci miejskich są te budowane przez ośrodki akademickie, łączące nie tylko budynki uniwersyteckie, lecz także ośrodki poza głównymi zabudowaniami. Takie sieci mają też połączenia WAN z innymi uniwersytetami oraz często z internetem.
- *WAN* (ang. *Wide Area Networks*) — sieć rozległa, łącząca ze sobą oddzielne sieci LAN i MAN. W ten sposób sieci WAN łączą geograficznie oddalonych użytkowników. Są to największe sieci komputerowe, których przykładem jest internet.

Podział ze względu na topologię

Kolejny ważny podział sieci komputerowych związany jest z ich topologią, czyli sposobem rozmieszczenia komputerów w sieci i ich wzajemnego połączenia.

Topologie sieci LAN mogą być opisane zarówno na płaszczyźnie fizycznej, jak i logicznej. *Topologia fizyczna* określa, jak rozmieszczone są poszczególne elementy wchodzące w skład takiej sieci. *Topologia logiczna* opisuje wszelkie możliwe połączenia między parami mogących się komunikować punktów sieci.

Elementy fizyczne sieci to:

- komputery (hosty),
- podłączone do sieci urządzenia sieciowe (na przykład: router, hub, switch, modem, drukarka sieciowa),
- media umożliwiające połączenie elementów sieciowych (na przykład okablowanie).

Oto najczęściej spotykane topologie sieciowe:

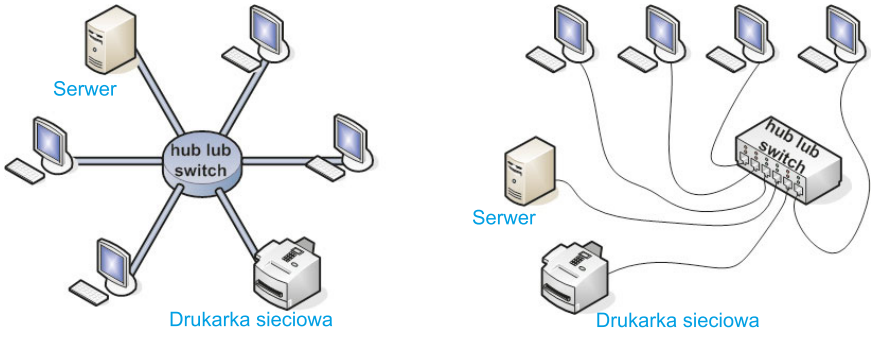
- z magistralą liniową,
- gwiazdy,
- pierścienia,
- podwójnego pierścienia,

- mieszane:
  - pierścień-gwiazda,
  - gwiazda-pierścień,
- drzewa,
- komórkowa.

Topologia logiczna opisuje reguły komunikacji, z których powinna korzystać każda stacja robocza (host) podczas komunikowania się w sieci. Poza połączeniem fizycznym hostów i ustaleniem standardu komunikacji topologia fizyczna zapewnia bezbłędną transmisję danych.

Topologia gwiazdy (ang. *star*)

Z połączeniem takim mamy do czynienia wtedy, gdy wszystkie elementy sieci podłączone są do jednego węzła centralnego, jak na rysunku 3.21. Funkcję węzła centralnego mogą pełnić urządzenia sieciowe, takie jak **koncentrator** (ang. *hub*) lub **przełącznik** (ang. *switch*). Uszkodzenie węzła centralnego skutkuje paraliżem całej sieci. Z kolei uszkodzenie jednego z połączeń powoduje jedynie zerwanie połączenia z danym komputerem, a pozostała część sieci działa poprawnie.

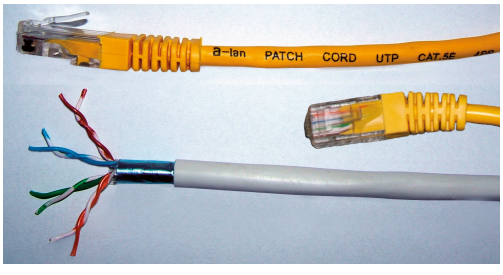


Rysunek 3.21. Topologia gwiazdy

Połączenia między komputerami a węzłem centralnym realizowane są za pomocą popularnej *skrętki* (rysunek 3.22). Skrętka wyróżnia się dużą niezawodnością i niskimi kosztami budowy sieci. Rozróżniamy dwa rodzaje tego typu kabla:

- kabel ekranowany **STP** (ang. *Shielded Twisted Pair*),
- kabel nieekranowany **UTP** (ang. *Unshielded Twisted Pair*).

Kable ekranowane zmniejszają straty transmisji i są bardziej odporne na zakłócenia. Mimo to powszechnie stosuje się skrętkę nieekranowaną (zapewne z powodów ekonomicznych). Jednak wszędzie tam, gdzie występują pola magnetyczne i gdzie konieczna jest niezawodność działania sieci (w szpitalach, wojsku, bankach), stosuje się skrętkę ekranowaną.



Rysunek 3.22. Kabel UTP zakończony wtyczką RJ-45 i skrętka STP (ekranowana) z czterema parami przewodów

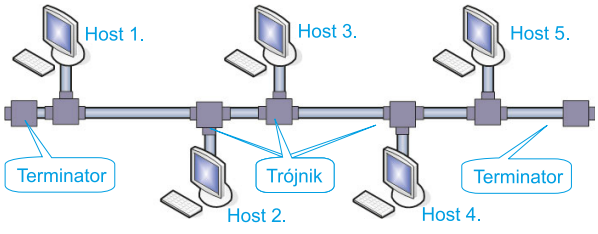
Odległości pomiędzy komputerami a hubem lub switchem nie powinny przekraczać 100 metrów. Jest to dominujące rozwiązanie we współczesnych sieciach LAN.

Zalety topologii gwiazdy:
- łatwa konserwacja i lokalizacja uszkodzeń,
- prosta rekonfiguracja sieci,
- proste (łatwe) podłączenie kolejnego urządzenia sieciowego,
- stosunkowo szybka komunikacja w sieci,
- niezawodność (uszkodzenie jednego z połączeń nie paraliżuje całej sieci).

Wady:
- duża liczba kabli (tyle połączeń, ile urządzeń),
- ograniczona możliwość rozbudowy sieci,
- ograniczenie odległości komputera od koncentratora lub przełącznika — w zależności od kategorii wynosi maksymalnie od 90 m do około 110 m,
- w przypadku awarii koncentratora lub przełącznika przestaje działać cała sieć.

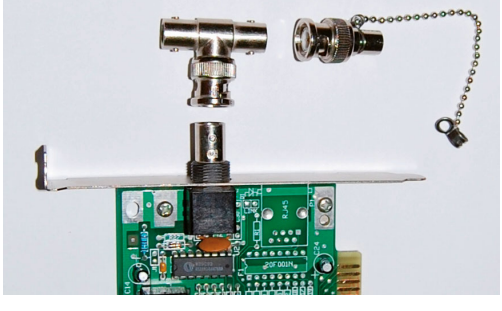
Topologia magistrali (ang. *bus*)

Topologia magistrali, zwana czasami szyną, pokazana została na rysunku 3.23. W topologii tej nie przewidziano żadnych urządzeń pośredniczących, zatem wszystkie hosty przyłączone do sieci słuchają transmisji przesyłanych magistralą i odbierają pakiety do nich zaadresowane. Dane są przesyłane kablem do wszystkich elementów sieci. Brak dodatkowych urządzeń sprawia, że sieci lokalne oparte na topologii magistrali są proste w konstrukcji i niedrogie. Są one przeznaczone przede wszystkim do użytku w domach i małych biurach.



Rysunek 3.23. Topologia magistrali

Typowa magistrala składa się z pojedynczego kabla łączącego wszystkie komputery (hosty) w sposób charakterystyczny dla sieci równorzędnej. Długość sieci w tej topologii nie powinna przekroczyć 185 m (licząc od jednego terminatora do drugiego). Kabel koncentryczny biegnie od komputera do komputera i jest podłączony do kart sieciowych. Każda z kart sieciowych zaopatrzona jest w złącze cylindryczne **BNC** (ang. *Bayonet Neill-Concelman*), do którego podłączany jest trójnik, jak na rysunku 3.24.



Rysunek 3.24. Karta sieciowa ze złączem BNC, trójnik i terminator

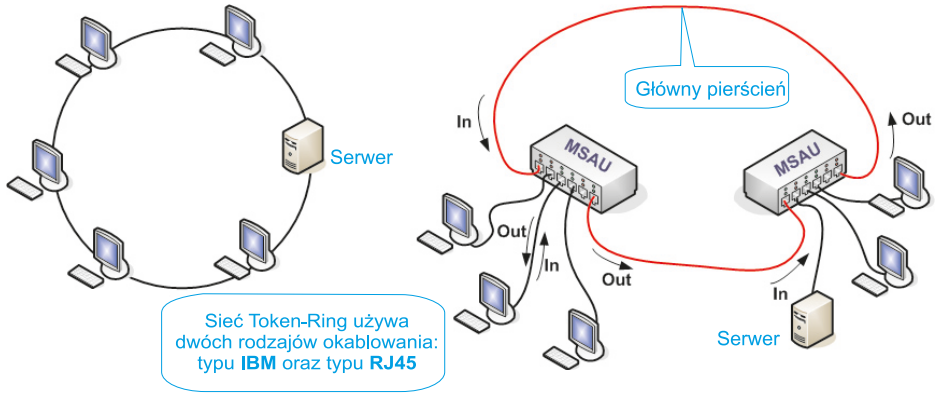
W przypadku uszkodzenia kabla przestaje działać cała sieć.

Zalety topologii magistrali:
- małe zużycie kabla, niski koszt samej sieci,
- prosta instalacja,
- bardzo prosta rozbudowa sieci,
- łatwe łączenie segmentów sieci w jeden system (bez zmian oprogramowania komunikacyjnego),
- każdy komputer jest podłączony tylko do jednego kabla.

Wady topologii magistrali:
- problem z dostępem do sieci — wszystkie komputery współużytkują jeden kabel;
- utrudniona diagnostyka błędów z powodu braku centralnego systemu zarządzającego siecią; w dużej sieci problem z lokalizacją uszkodzenia;
- zwykle dla uniknięcia zakłóceń sygnałów należy zachować pewną odległość między punktami przyłączenia poszczególnych stacji;
- kabel koncentryczny jest dość wrażliwy. Nie znosi ostrych skrętów ani nawet łagodnie przykładanej siły gniotącej. Jego struktura łatwo ulega uszkodzeniu, co powoduje bezpośrednie pogorszenie transmisji sygnału;
- przerwanie kabla paraliżuje całą sieć;
- sieć wolna, niski transfer danych — 10 Mb/s (kabel 10 Base-2).

Topologia pierścienia (ang. *ring*)

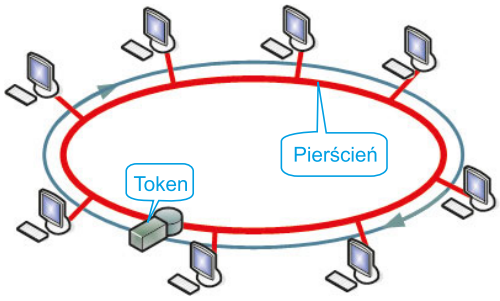
W topologii pierścienia połączenie kablowe elementów sieci tworzy zamknięty pierścień, tak jak to pokazano na rysunku 3.25. Przykładem sieci o topologii pierścienia jest *Token-Ring*, gdzie stacje sieciowe podłącza się bezpośrednio do urządzeń MSAU (ang. *Multi-Station Access Unit* — jednostka dostępowa dla wielu stacji), które z kolei łączy się ze sobą tak, by tworzyły jeden duży pierścień. Urządzenia MSAU mają wbudowane elementy obejściowe umożliwiające pracę pierścienia nawet po odłączeniu dowolnej stacji z MSAU. To oznacza, że odłączenie lub awaria któregoś z komputerów nie paraliżuje działania całej sieci.



Rysunek 3.25. Topologia pierścienia (schemat ogólny, przykładowe rozwiązanie połączenia)

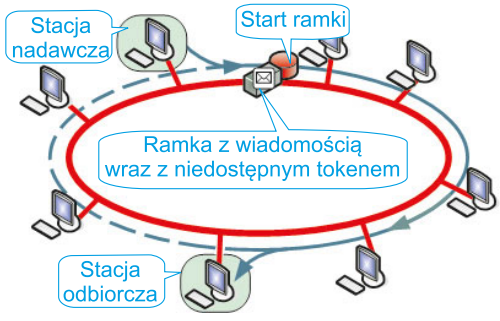
Dane przesyłane są wokół pierścienia, w jednym kierunku. Każda stacja robocza pobiera i odpowiada na pakiety do niej zaadresowane, a pozostałe pakiety przesyła dalej, do następnej stacji roboczej, wchodzącej w skład sieci. Jak to się dzieje?

W pierścieniu sieci Token-Ring krąży mała ramka, zwana *tokenem* (żetonem). Stacja sieciowa uzyskuje prawo do transmisji informacji tylko wtedy, gdy ma token. Jeśli więc dowolna stacja sieciowa przejmuje token, ale w tym momencie nie zamierza transmitować danych, przesyła żeton do następnej w kolejności stacji sieciowej, co ilustruje rysunek 3.26. Każda stacja może przetrzymywać token tylko przez określony czas (na przykład 10 ms).

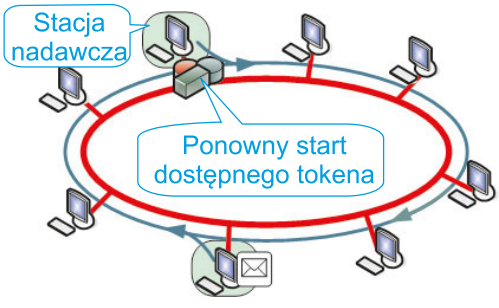


Rysunek 3.26. Cisza w sieci — krąży token

Stacja nadawcza, przy której znajdzie się token, mająca informację do przesłania, zmienia jeden bit w tokenie, co powoduje start ramki z niedostępnym tokenem, następnie dodaje informację, którą chce transmitować, po czym całość wysyła do następnej stacji w pierścieniu. Podczas gdy ramka przesuwa się w pierścieniu, nie ma w nim dostępnego tokena, co oznacza, że inne stacje, chcące w tym czasie rozpocząć transmisję, muszą czekać (rysunek 3.27). Oznacza to także, że w sieciach Token-Ring nie występują kolizje.



Rysunek 3.27. Start ramki — token niedostępny



Rysunek 3.28. Zakończenie transmisji — ponowny start tokena

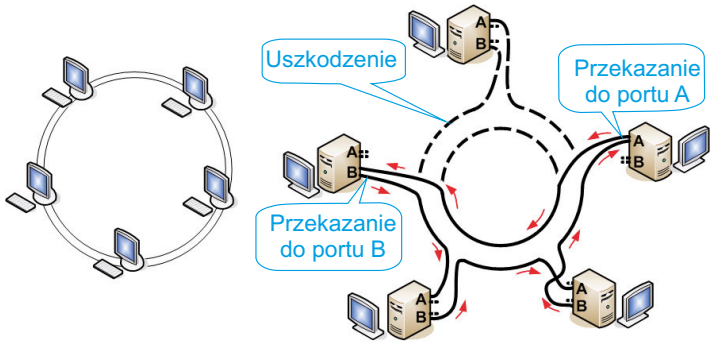
Ramka informacyjna, krążąc w pierścieniu, osiąga wreszcie stację odbiorczą, która kopiuje ją do dalszego przetwarzania. Ramka kontynuuje wędrówkę w pierścieniu aż do momentu osiągnięcia stacji nadawczej. Tutaj zostaje usunięta z pierścienia. Stacja nadawcza może sprawdzić, czy ramka dotarła do stacji odbiorczej i czy tam została skopiowana. Stacja nie ma prawa wysłania nowej wiadomości. Po zakończeniu transmisji generuje zatem nowy token (zmienia bit) i wysyła go do następnej stacji (rysunek 3.28). Dopiero gdy ponownie otrzyma token, może wysłać nową wiadomość.

Zaletą takiej topologii jest możliwość w miarę dokładnego określenia czasu odpowiedzi sieci. Czas ten się wydłuża, gdy sieć jest rozbudowana. Z kolei wadą takiej topologii jest jej unieruchomienie po przerwaniu połączenia kablowego.

Wady topologii pierścienia:

- awaria jednego węzła lub łącza może być powodem awarii całej sieci;
- utrudniona diagnoza uszkodzeń;
- dołączenie nowego węzła wymaga wyłączenia całej sieci;
- dane poruszają się w jednym kierunku;
- czas przesyłu danych jest wprost proporcjonalny do liczby komputerów.

Topologia podwójnego pierścienia składa się z dwóch pierścieni o wspólnym środku (dwa pierścienie nie są połączone ze sobą) — zob. rysunek 3.29. Jest to topologia podobna do poprzedniej, z tą różnicą, że drugi zapasowy pierścień łączy te same urządzenia. W celu zapewnienia niezawodności i elastyczności sieci każde urządzenie sieciowe jest częścią dwóch niezależnych topologii pierścienia.



Rysunek 3.29. Reakcja sieci w topologii podwójnego pierścienia na uszkodzenie pierścienia głównego

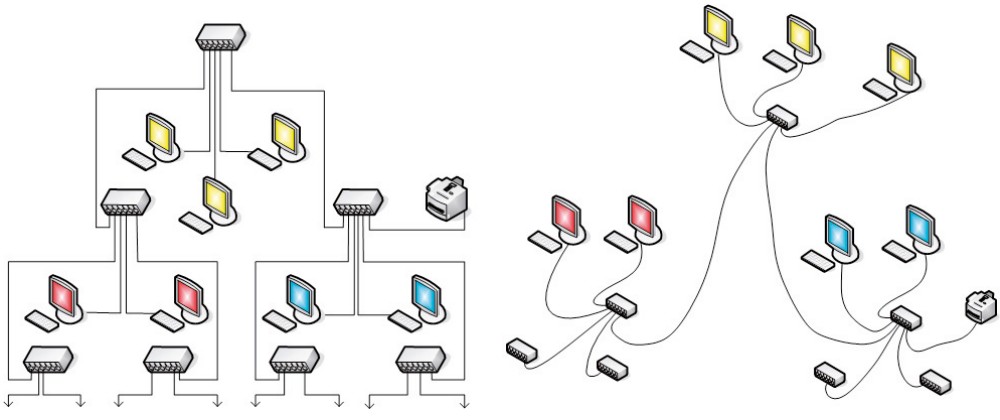
Takie rozwiązanie (dwa niezależne pierścienie) pozwala na zwiększenie niezawodności sieci. W poprzedniej wersji topologii sieci Token-Ring z pojedynczym pierścieniem uszkodzenie sieci w dowolnym miejscu paraliżowało całą sieć. W przypadku topologii sieci z podwójnym pierścieniem problem ten został wyeliminowany.

Jeden z pierścieni jest nazywany głównym (*primary*), drugi — pomocniczym (*secondary*). W zwykłych warunkach wszystkie dane krążą po pierścieniu głównym, a pierścień pomocniczy pozostaje niewykorzystany. Pierścień pomocniczy zostaje użyty dopiero wtedy, gdy pierścień główny ulega przerwaniu. Następuje wówczas automatyczna rekonfiguracja polegająca na przełączeniu portów i przekierowaniu sygnału do pierścienia pomocniczego w taki sposób, aby sygnał mógł nadal podążać w zamkniętym pierścieniu. Dlatego też często spotykamy się z określeniem, że w obu pierścieniach podobne strumienie danych krążą w przeciwnych kierunkach. Ma to oczywiście miejsce wtedy, gdy pierścień pomocniczy musi być wykorzystywany podczas awarii pierścienia głównego.

Topologia drzewa (ang. *tree*)

Topologia drzewa nazywana jest także topologią hierarchiczną. Przypomina topologię gwiazdy, z tą różnicą, że na jej zakończeniach zamiast stacji roboczych mogą być przy-

łączone kolejne koncentratory, do których przyłączamy kolejne stacje lub kolejne koncentratory. Dzięki temu możemy budować naprawdę duże i wydajne sieci. Istnieją dwa rodzaje tej topologii: *drzewo binarne*, gdzie każdy węzeł ma dwa połączenia, jak na rysunku 3.30, oraz *drzewo szkieletowe*, w którym węzły rozchodzą się od pnia szkieletu. Pień to przewód składający się z kilku warstw rozgałęzień. Przepływ informacji jest hierarchiczny, pierwszeństwo mają „ci na górze".



Rysunek 3.30. Drzewo binarne — topologia drzewa

Topologia ta jest bardzo elastyczna i w niektórych systemach transportu sieciowego umożliwia praktycznie dowolne konfiguracje. Wadą tego typu rozwiązania jest utrudnienie wyszukiwania błędów sieci w przypadku awarii.

Zalety:

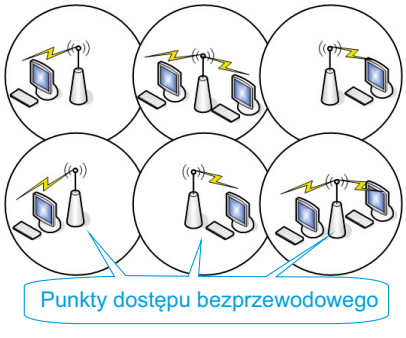
- pozwala na stosowanie krótszych przewodów;
- ogranicza liczbę urządzeń, które muszą być podłączone do centralnego węzła.

Wady:

- awaria na łączu pomiędzy koncentratorami lub przełącznikami odcina cały segment sieci.

Topologia komórkowa (ang. *cellular*)

Topologia komórkowa składa się z kulistych lub sześciennych obszarów, z których każdy ma jeden węzeł będący centrum. W topologii komórkowej nie ma fizycznego połączenia, dane transmitowane są za pomocą fal elektromagnetycznych, co symbolicznie ilustruje rysunek 3.31. Czasami przemieszczają się węzły odbiorcze, a czasami węzły nadawcze (na przykład komunikacyjne łącze satelitarne). Oczywistą zaletą topologii komórkowej (bezprzewodowej) jest to, że medium przenoszącym sygnał jest atmosfera lub próżnia kosmiczna.

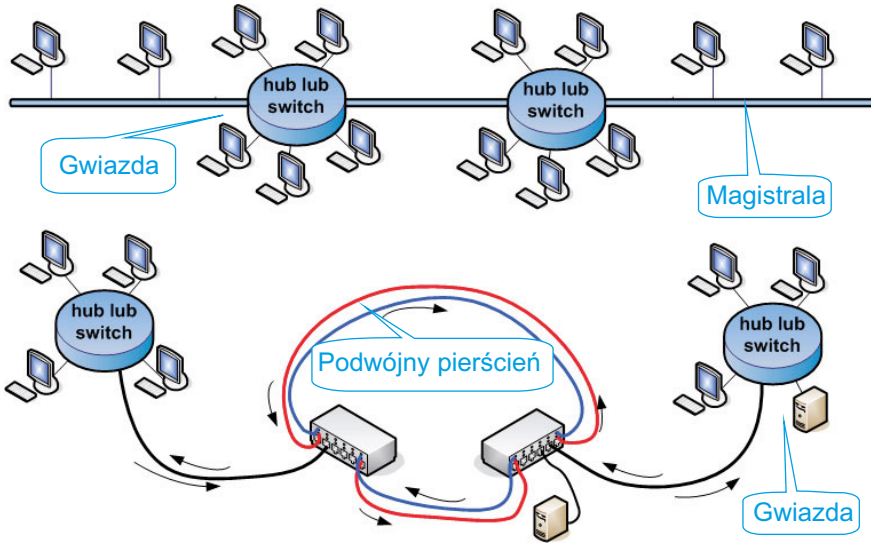


Rysunek 3.31. Topologia komórkowa

W przypadku sieci bezprzewodowych trudno ustalić ścisłe terytorium rozchodzenia się fal radiowych, gdyż mogą być one zależne na przykład od pogody. Materiały takie jak woda, metal czy beton również znacznie obniżają jakość sygnału. Fale mogą być zakłócane przez inne urządzenia nadające na tych samych częstotliwościach.

Topologia mieszana

Sieci rzadko są projektowane w postaci pojedynczej topologii. W topologii mieszanej dwie (lub więcej) topologie połączone są w jedną sieć. Na przykład można zaprojektować sieć topologii gwiazdy i magistrali w celu wykorzystania zalet każdej z nich. Dwa rodzaje topologii mieszanych są często stosowane: **magistrala-gwiazda** oraz **podwójny pierścień-gwiazda** (rysunek 3.32).



Rysunek 3.32. Topologie mieszane (topologia magistrala-gwiazda oraz topologia podwójny pierścień-gwiazda)

W topologii magistrala-gwiazda kilka sieci o topologii gwiazdy jest połączonych w układzie magistrali. Awaria jednego komputera nie wpływa na działanie reszty sieci. Jednakże jeśli awarii ulegnie koncentrator łączący wszystkie komputery gwiazdy, to komputery podłączone do tego urządzenia nie będą mogły komunikować się w sieci.

W topologii podwójny pierścień-gwiazda komputery są podłączone do centralnego urządzenia jak w topologii gwiazdy, a następnie urządzenia te są połączone ze sobą w topologii podwójnego pierścienia.

Podobnie jak w przypadku topologii magistrala-gwiazda, awaria jednego komputera nie wpływa na działanie reszty sieci. Dzięki metodzie przekazywania tokena możliwy jest większy ruch między segmentami sieci niż w przypadku sieci o topologii magistrala-gwiazda. Zastosowanie podwójnego pierścienia gwarantuje większą niezawodność całej sieci.

Definicja

Przepustowość (pojemność kanału) to maksymalna ilość informacji, jaka może być przesłana przez kanał telekomunikacyjny lub łącze w jednostce czasu, mierzona w b/s (bps, ang. *bits per second* — bity na sekundę).

Definicja

Przepływność to szybkość transmisji danych (ang. *bit rate*), mierzona w takich samych jednostkach jak przepustowość.

Ze względu na tę samą jednostkę przepustowość jest mylnie utożsamiana z przepływnością. Przepływność to miara natężenia strumienia informacji (danych), podczas gdy przepustowość jest cechą toru lub kanału telekomunikacyjnego.

Okablowanie strukturalne

W przypadku sieci komputerowych rozróżniamy kilka podstawowych typów okablowania. Są one ściśle powiązane z zastosowaną topologią sieciową. Na przykład specyfikacja *Ethernet* umożliwia wykorzystanie topologii fizycznej gwiaździstej lub magistrali, ale nie pozwala zbudować sieci o topologii pierścienia.

W zależności od specyfikacji sieci komputerowej stosujemy różne okablowanie. Ze względu na ciągły rozwój sieci i technologii obserwujemy tworzenie nowych standardów, które umożliwiają wydajniejszą i bezpieczniejszą komunikację w sieci. Mówiąc o mediach transmisyjnych, powinniśmy pamiętać, że nie mamy tu do czynienia jedynie z sieciami opartymi na połączeniach kablowych opisanych powyżej (BNC i skrętka). Obecnie stosujemy jeszcze sieci oparte na *światłowodach* oraz na *technologii radiowej*.

W **światłowodach** do transmisji informacji wykorzystuje się wiązkę światła, która jest odpowiednikiem prądu w innych kablach. Wiązka ta jest modulowana zgodnie z treścią przekazywanych informacji. To rozwiązanie otworzyło nowe możliwości w dziedzinie tworzenia szybkich i niezawodnych sieci komputerowych. Właściwie dobrany kabel może zostać ułożony w każdym środowisku. Szybkość transmisji może wynosić nawet 3 Tb/s. Więcej informacji na temat światłowodów zamieszczono na płycie CD, w pliku *swiatlowody.pdf*.

Najczęściej spotykane specyfikacje sieci komputerowej

Ethernet to zbiór reguł opisujących sposób tworzenia i działania głównie sieci lokalnych. Zdefiniowano między innymi sposoby przesyłania informacji oraz specyfikację kabli sieciowych. Specyfikacja ta została podana w standardzie *802.3 IEEE* (ang. *Institute of Electrical and Electronic Engineers* — Instytut Inżynierów Elektryków i Elektroników). Szczegółowy podział i opis specyfikacji znajduje się na płycie CD, w dokumencie *specyfikacje sieci komputerowej.pdf*.

HSPA to mobilny szerokopasmowy dostęp do internetu. Muzyka i wideo na żądanie oraz mobilna telewizja to tylko kilka przykładowych usług, które zyskują na atrakcyjności

dzięki wprowadzeniu HSPA (ang. *High Speed Packet Access*), czyli szybkiej transmisji pakietowej. Sieci HSPA są aktualnie komercyjnie dostępne w wielu krajach na świecie, w tym również w Polsce. Pozwalają na osiągnięcie szybkości do 14,4 Mb/s w kierunku do abonenta oraz do 1,92 Mb/s w kierunku od abonenta do sieci. Uruchomienie komercyjnej usługi HSPA w większości sieci zaowocowało bardzo dużym, ciągłym wzrostem ruchu pakietowego abonentów. Niezaprzeczalny sukces tej technologii pozwala operatorom już dziś na silną rywalizację z sieciami kablowymi w oferowaniu dostępu do internetu.

Dziś mamy do czynienia z powszechną dostępnością przenośnych urządzeń dostępowych HSPA, takich jak karty PC, telefony oraz komputery przenośne z wbudowanymi modemami. Atrakcyjne taryfy, w połączeniu z imponującymi osiągami oferowanymi przez HSPA, są czynnikiem, który powoduje zwiększone zainteresowanie rynku usługami bezprzewodowej transmisji danych.

LTE (ang. *Long Term Evolution* — ewolucja długofalowa) jest nazwą sieci mobilnej. Standard umożliwia zwiększenie możliwości transferu danych w systemach telefonii komórkowej. Zaproponowany przez Sony Ericsson i T-Mobile, został przyjęty w styczniu 2008 roku. Oferuje transfery na poziomie ponad 100 Mb/s (teoretycznie nawet do 300 Mb/s). Sieć LTE może być uruchomiona w różnej gamie częstotliwości, w przedziale od 1,25 do 20 MHz. Zastosowanie będzie zależało od dostępności spektrum w danym kraju. Technologia LTE ma szansę na szybkie upowszechnienie, gdyż wprowadzenie jej nie wiąże się z koniecznością budowy zupełnie nowej infrastruktury. Niezbędne jest tylko przebudowanie istniejących stacji bazowych, by można było oferować klientom mobilny dostęp do internetu czwartej generacji (4G), w którym przeglądanie stron internetowych i przesyłanie plików będzie możliwe natychmiast po kliknięciu.

Urządzenia sieciowe — rozbudowa sieci

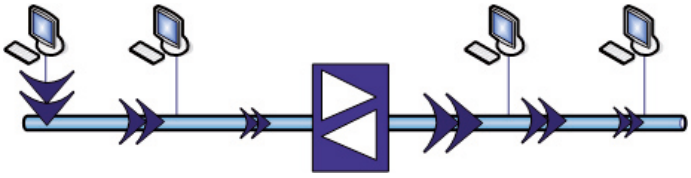
Rozwijające się firmy często stają przed problemem rozbudowy sieci komputerowej lub zwiększenia jej wydajności. Nie można powiększać sieci jedynie przez dodawanie nowych komputerów i okablowania. Każda topologia sieciowa lub architektura sieciowa ma swoje ograniczenia. Można jednak w istniejącej sieci zainstalować dodatkowe urządzenia w celu jej rozbudowy.

Urządzenia sieciowe zapewniają prawidłowe działanie sieci. Pełnią określone funkcje, takie jak: regeneracja sygnału, filtrowanie pakietów, trasowanie (ang. *routing*) i konwersja sygnału.

Do urządzeń umożliwiających rozbudowę sieci należą: wzmacniak, koncentrator, most, przełącznik, router.

Wzmacniak (ang. *repeater*)

Wzmacniaki odbierają sygnały i retransmitują je z oryginalną mocą i charakterystyką, co ilustruje rysunek 3.33. Jeśli kabel jest bardzo długi, sygnał słabnie i może stać się nieczytelny. Zainstalowanie wzmacniaka między segmentami kabla umożliwia przesyłanie sygnału na większe odległości.



Rysunek 3.33. Wzmacniak (repeater) wzmacnia sygnał, umożliwia przedłużenie sieci

Wzmacniaki nie tłumaczą i nie filtrują sygnałów. Aby wzmacniak działał prawidłowo, oba segmenty podłączone do wzmacniaka muszą używać tej samej metody dostępu. Na przykład wzmacniak nie może tłumaczyć pakietów sieci Ethernet na pakiety sieci Token-Ring.

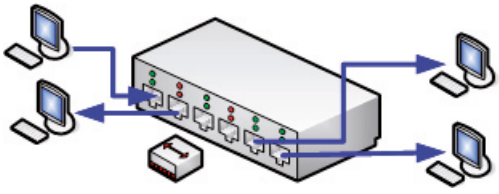
Wzmacniaki nie mogą pełnić funkcji filtrów zapobiegających problemom w ruchu sieciowym. Wzmacniaki przesyłają każdy bit danych z jednego segmentu do drugiego, nawet jeśli dane zawierają pakiety nieprawidłowe lub takie, które nie są przeznaczone dla żadnego komputera w segmencie.

Za pomocą wzmacniaka można:

- połączyć dwa segmenty podobnego lub różnego okablowania;
- regenerować sygnał w celu zwiększenia jego zasięgu;
- przekazywać cały ruch w sieci w obu kierunkach;
- połączyć dwa segmenty najtańszym kosztem.

Koncentrator (ang. *hub*)

Koncentrator to urządzenie do łączenia komputerów pracujących w topologii gwiazdy. Koncentrator ma wiele portów do podłączania urządzeń sieciowych. Pojedynczy pakiet danych jest wysyłany przez koncentrator do wszystkich podłączonych komputerów. Tę zasadę działania ilustruje rysunek 3.34.



Rysunek 3.34. Koncentrator (hub) transmituje dane do wszystkich komputerów w topologii gwiazdy

Jeśli używany jest koncentrator, przerwanie kabla sieciowego nie wpływa na całą sieć, a jedynie na odcięty segment.

Istnieją dwa rodzaje koncentratorów:

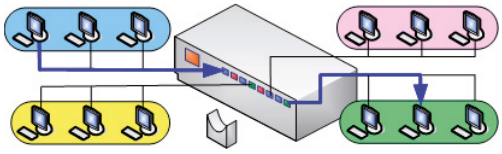
- pasywne, które wysyłają przychodzący sygnał bez jego przetwarzania, bezpośrednio do portów;
- aktywne, czasami nazywane wzmacniakami wieloportowymi, które odbierają sygnał, przetwarzają go i retransmitują w jego oryginalnej postaci do wszystkich podłączonych komputerów i urządzeń.

Za pomocą koncentratora można:

- w łatwy sposób zmienić i rozbudować system okablowania;
- połączyć różne rodzaje okablowania poprzez różne porty koncentratora.

Most (ang. *bridge*)

Most jest urządzeniem przesyłającym pakiety danych między segmentami sieci LAN używającymi tego samego protokołu komunikacyjnego. Most przesyła jednorazowo jeden sygnał. Jeśli pakiet jest adresowany do komputera w tym samym segmencie co komputer wysyłający, most zatrzymuje pakiet wewnątrz tego segmentu. Jeśli pakiet jest adresowany do innego segmentu, most przesyła go do tego segmentu, co pokazano na rysunku 3.35.



Rysunek 3.35. Transmisja sygnału poprzez most między dwoma segmentami sieci

Jeśli pakiet został przesłany przez most, informacje o *adresie fizycznym MAC* komputera wysyłającego są zapisywane w pamięci mostu. Most używa tych informacji do zbudowania tabeli adresów podłączonych do niego urządzeń. W trakcie wysyłania danych most rozbudowuje tabelę adresów określających każdy komputer oraz jego lokalizację w segmentach sieci.

Definicja

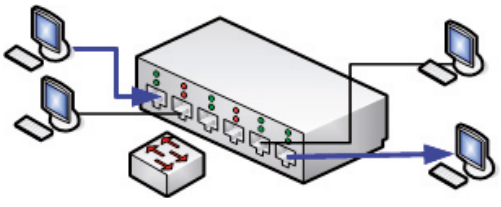
Adres fizyczny MAC (ang. *Media Access Control*) to 48-bitowy niepowtarzalny adres urządzenia sieciowego jednoznacznie identyfikujący urządzenia w sieci. Sam adres składa się z dwóch głównych części: pierwsza identyfikuje producenta urządzenia, a druga — rodzaj urządzenia i jego unikatowy numer.

Za pomocą mostu można:

- zwiększyć liczbę segmentów;
- uwzględnić zwiększenie liczby komputerów w sieci;
- zmniejszyć wpływ nadmiaru liczby komputerów na przepustowość sieci;
- podzielić przeciążoną sieć na dwie odseparowane sieci, aby zmniejszyć ruch w każdym segmencie i zwiększyć wydajność każdej sieci;
- połączyć różne elementy okablowania, na przykład skrętkę z kablem koncentrycznym.

Przełącznik (ang. *switch*)

Przełączniki działają podobnie jak mosty, lecz oferują bardziej bezpośrednie połączenie między komputerami źródłowym i docelowym. Kiedy przełącznik odbierze pakiet danych, tworzy oddzielne wewnętrzne połączenie między dwoma portami i, opierając się na informacji zawartej w nagłówku każdego pakietu, przesyła pakiet dalej jedynie do portu komputera przeznaczenia, co ilustruje rysunek 3.36. Dzięki temu połączenie jest odizolowane od innych portów, a komputery źródłowy i przeznaczenia mogą w pełni wykorzystać przepustowość sieci.

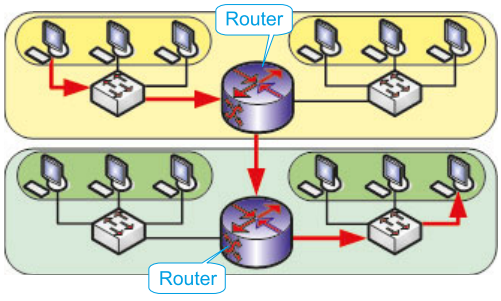


Rysunek 3.36. Przełącznik przesyłający pakiety danych w jednej podsieci

Za pomocą przełącznika można:
- dzięki zapamiętanym w tablicy adresom MAC wysyłać pakiety bezpośrednio z komputera źródłowego do komputera przeznaczenia;
- umożliwić większe prędkości transmisji danych.

Router

Router jest urządzeniem działającym w warstwie trzeciej (więcej o warstwach modelu OSI na str. 157). Przesyłając dane między różnymi sieciami, routery sprawdzają nagłówek pakietu, aby określić najlepszą drogę jego przesłania. Drogę pakietu przesyłanego z jednej sieci do drugiej poprzez router ilustruje rysunek 3.37. Dzięki informacjom przechowywanym w *tablicy routingu* router zna ścieżki do wszystkich sieci. Zapisuje w tablicy adresy IP kolejnych sieci, do których może posyłać pakiety danych. Routery umożliwiają współużytkowanie przez wszystkich użytkowników pojedynczego łącza do sieci internet.



Rysunek 3.37. Przesłanie pakietu poprzez router pomiędzy różnymi podsieciami

Za pomocą routera można wysyłać pakiety bezpośrednio do komputera przeznaczenia pracującego w innej sieci.

Routery używają bardziej kompletnego adresu pakietu niż mosty w celu określenia, który router lub klient ma jako następny odebrać pakiet. Routery zapewniają, że pakiety wędrują do miejsca przeznaczenia najbardziej efektywną trasą. Jeśli połączenie między dwoma routerami ulegnie awarii, router wysyłający może określić alternatywny router w celu zapewnienia dalszego przesyłania pakietów.

Routery czytają wyłącznie zaadresowane pakiety sieciowe i przesyłają informacje tylko wtedy, gdy adres sieci jest znany. Routery nie przesyłają uszkodzonych danych ani pakietów zaadresowanych adresem broadcastowym. Możliwość kontroli przesyłanych danych przez router zmniejsza ruch między sieciami i pozwala routerom na bardziej efektywne wykorzystanie połączeń. Obciążenie sieci jest więc mniejsze.

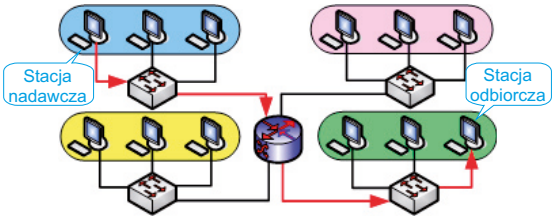
Sposoby transmisji i adresowania w LAN

Wyróżnia się trzy sposoby transmisji i adresowania w sieciach lokalnych LAN — są to:
- transmisja pojedyncza *Unicast*,
- transmisja grupowa *Multicast*,
- transmisja rozgłoszeniowa *Broadcast*.

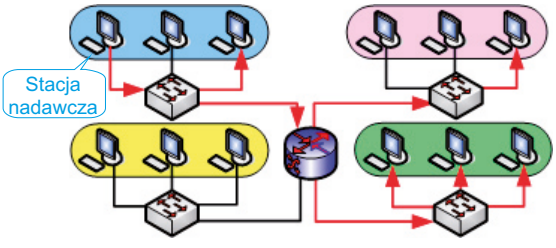
W transmisji **Unicast** pojedynczy pakiet jest wysyłany przez stację nadawczą **do jednej stacji odbiorczej**. Przedtem jednak stacja nadawcza adresuje pakiet, używając adresu stacji odbiorczej. Po zaadresowaniu pakiet jest wysyłany do sieci. Taką transmisję do pojedynczej stacji odbiorczej ilustruje rysunek 3.38.

W transmisji **Multicast** pojedynczy pakiet danych jest kopiowany i **wysyłany do grupy stacji sieciowych** (określonej przez adres multicast). Przedtem jednak stacja nadawcza adresuje pakiet, używając adresu multicast. Po zaadresowaniu pakiet jest wysyłany do sieci, gdzie zostaje skopiowany. Kopia pakietu jest wysyłana jednocześnie do wszystkich stacji należących do grupy adresów multicast. Schematyczny przekaz pakietu ze stacji nadawczej do grupy stacji odbiorczych zilustrowano na rysunku 3.39.

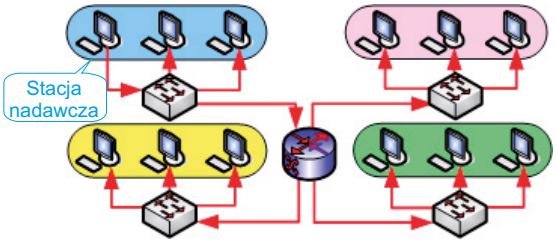
W transmisji **Broadcast** pojedynczy pakiet jest kopiowany i wysyłany **do wszystkich stacji sieciowych**. W tym typie transmisji stacja nadawcza adresuje pakiet, używając adresu broadcast. Następnie pakiet jest wysyłany do sieci, gdzie zostaje skopiowany. Kopie są wysyłane do wszystkich stacji sieciowych. Schematyczny przekaz pakietu ze stacji nadawczej do wszystkich stacji sieciowych zilustrowano na rysunku 3.40.



Rysunek 3.38. Transmisja do pojedynczej stacji odbiorczej — Unicast



Rysunek 3.39. Transmisja grupowa — Multicast



Rysunek 3.40. Transmisja rozgłoszeniowa — Broadcast

3.4.2. Model OSI

We wczesnych latach istnienia sieci transmitowanie poprzez nią danych sprawiało wiele problemów, ponieważ duże firmy opracowywały własne, różne standardy łączenia komputerów w sieć. Aplikacje działające na różnych rodzajach komputerów, napisane przez różnych producentów, nie były w stanie komunikować się ze sobą. Niezbędne było stworzenie standardu, który pozwoliłby na wymianę informacji przez komputery różnych typów, z różnymi systemami operacyjnymi i różnym oprogramowaniem.

Aby umożliwić współpracę urządzeń pochodzących od różnych dostawców, konieczne stało się opracowanie zasad opisujących sposoby ich komunikowania się. Standardy takie są tworzone przez organizacje międzynarodowe. Do najbardziej znanych należą **ISO** (ang. *International Standard Organization* — Międzynarodowa Organizacja Normalizacyjna) i **IEEE** (ang. *Institute of Electrical and Electronic Engineers*). Chociaż ich postanowienia nie mają mocy prawnej, wiele rządów czyni z nich obowiązujące standardy.

W 1978 roku Międzynarodowa Organizacja Normalizacyjna (ISO) zaproponowała *model architektury sieci*, zwany *modelem OSI* (ang. *Open System Interconnection*), integrujący wszystkie lokalne podsieci. Był to pierwszy krok w kierunku stworzenia jednolitego standardu komunikacyjnego.

Podstawowym założeniem modelu jest podział systemów sieciowych na siedem warstw, pokazanych na rysunku 3.41, współpracujących ze sobą w ściśle określony sposób. Model ten nie określa fizycznej budowy poszczególnych warstw, lecz koncentruje się na sposobach ich współpracy.



Rysunek 3.41. Warstwy w modelu OSI

Funkcje poszczególnych warstw

1. Warstwa fizyczna

Fundament, na którym zbudowany jest model referencyjny OSI, stanowi warstwa fizyczna. Określa ona wszystkie składniki sieci niezbędne do obsługi elektrycznego, optycznego, radiowego wysyłania i odbierania sygnałów. Jest odpowiedzialna za przesyłanie strumieni bitów. Odbiera ramki danych z warstwy 2. (łącza danych) i przesyła szeregowo, bit po bicie, całą ich strukturę oraz zawartość. Jest ona również odpowiedzialna za odbiór kolejnych bitów przychodzących strumieni danych. Strumienie te są następnie przesyłane do warstwy łącza danych w celu ich ponownego ukształtowania. Warstwa ta ustala również szybkość wysyłania informacji, zależną od jakości połączenia. Warstwa 1. nie jest wyposażona w żadne mechanizmy pozwalające jej na rozróżnianie znaczenia i wagi kolejnych bitów. Operuje jedynie wartościami 1 i 0, czyli jest uzależniona od fizycznych własności przesyłania sygnałów (elektrycznych lub optycznych). Parametry przesyłu zależą od rodzaju użytego nośnika. Warstwa fizyczna obejmuje procesy i mechanizmy dotyczące przenoszenia sygnałów. Nie obejmuje natomiast medium transmisyjnego, czyli nośnika (kabli koncentrycznych, skrętki, światłowodów). Media transmisyjne czasami określane są mianem warstwy zerowej.

2. Warstwa łącza danych

Jest odpowiedzialna za końcową zgodność przesyłania danych. Zajmuje się pakowaniem danych w tzw. **ramki** i wysyłaniem ich do warstwy fizycznej.

Definicja

Ramka jest formą organizacji danych warstwy łącza danych, która zawiera ilość informacji wystarczającą do pomyślnego przesyłania danych przez sieć lokalną do ich miejsca docelowego.

Czasami wysyłane ramki nie osiągają miejsca docelowego lub ulegają uszkodzeniu podczas transmisji. Warstwa łącza danych jest odpowiedzialna za rozpoznawanie i naprawę każdego takiego błędu. Warstwa łącza danych jest również odpowiedzialna za ponowne składanie otrzymanych z warstwy fizycznej strumieni binarnych i umieszczanie ich w ramkach.

3. Warstwa sieciowa

Warstwa sieciowa jest odpowiedzialna za określenie trasy transmisji między komputerem nadawcą a komputerem odbiorcą. Warstwa ta nie ma żadnych wbudowanych mechanizmów korekcji błędów i w związku z tym musi polegać na wiarygodnej transmisji końcowej warstwy łącza danych. Warstwa sieciowa używana jest do komunikowania się z komputerami znajdującymi się poza lokalnym segmentem sieci LAN. Umożliwia im to własna architektura trasowania, niezależna od adresowania fizycznego warstwy 2. Korzystanie z warstwy sieciowej nie jest obowiązkowe. Wymagane jest jedynie wtedy, gdy komputery komunikujące się znajdują się w różnych segmentach sieci przedzielonych routerem.

4. Warstwa transportowa

Warstwa transportowa dba o poprawność przesyłania danych. Aby informacje mogły zostać przesłane w dół, często muszą zostać podzielone na mniejsze fragmenty. Warstwa transportowa jest odpowiedzialna za przesyłanie danych i integralność transmisji. Ustala odpowiednią kolejność pakietów przed wysłaniem ich zawartości do warstwy sesji. Odpowiada za nawiązywanie połączenia, wymianę danych oraz zamykanie połączenia. Potrafi wykrywać pakiety, które zostały odrzucone przez routery, i automatycznie generować żądanie ich ponownej transmisji.

5. Warstwa sesji

Warstwa sesji otrzymuje od różnych aplikacji dane, które muszą zostać odpowiednio zsynchronizowane. Potrafi rozpoznać połączenia między aplikacjami, dzięki czemu może zapewnić właściwy kierunek przepływu informacji. W nowoczesnych systemach sieciowych może jednocześnie pracować wiele aplikacji. Warstwa sesji odpowiada za nawiązanie sesji, zapewnienie uporządkowanej wymiany danych między aplikacjami i zamknięcie sesji, korzysta przy tym z usług warstwy transportowej (w niektórych sieciach obie warstwy są ze sobą połączone).

6. Warstwa prezentacji

Warstwa prezentacji jest odpowiedzialna za zarządzanie sposobem kodowania wszelkich danych. Nie każdy komputer korzysta z tych samych schematów kodowania

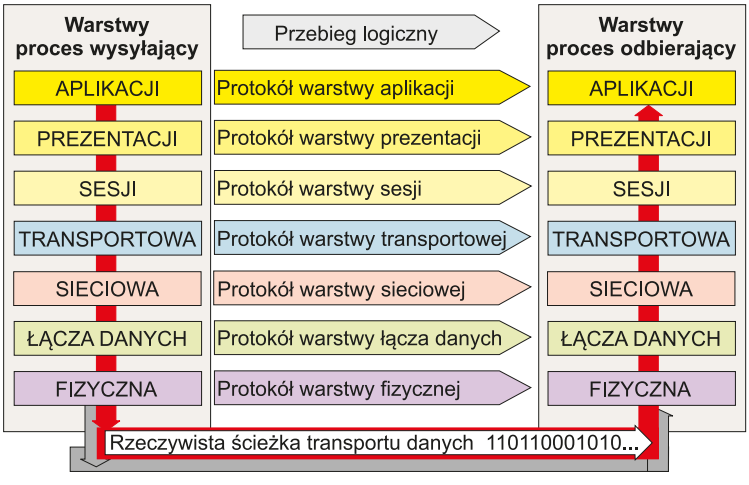
danych, więc warstwa prezentacji odpowiedzialna jest za translację między niezgodnymi schematami kodowania. Warstwa ta może być również wykorzystywana do szyfrowania i rozszyfrowywania wiadomości.

7. Warstwa aplikacji

Kiedy użytkownik, korzystając z oprogramowania, chce przesłać dane poprzez urządzenia sieciowe, to trafiają one właśnie do warstwy aplikacji. Pełni ona funkcję interfejsu pomiędzy aplikacjami użytkownika a usługami sieci. Warstwę tę można uważać za inicjującą sesje komunikacyjne.

Przepływ informacji pomiędzy warstwami

Aby dane mogły pokonać stos złożony z poszczególnych warstw, warstwy te muszą się między sobą komunikować. Co więcej — muszą być określone konkretne protokoły sieciowe wykorzystywane do porozumiewania się. Mamy wtedy do czynienia z przebiegiem logicznym informacji pomiędzy warstwami, pokazanym na rysunku 3.42. W rzeczywistości jednak sprawa ma się nieco inaczej: prawdziwa komunikacja odbywa się poprzez warstwę fizyczną.



Rysunek 3.42. Komunikacja między warstwami modelu OSI

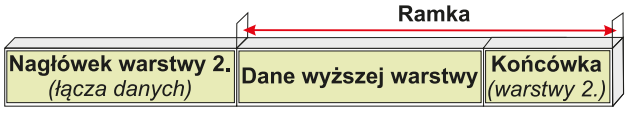
Definicja

Protokoły sieciowe to zbiór ścisłych reguł i kroków postępowania, które są automatycznie wykonywane przez urządzenia komunikacyjne w celu nawiązania łączności i wymiany danych.

Formaty informacji przesyłanych w sieci

W sieci przesyłane są dane oraz informacje sterujące. Jedne i drugie mogą przyjmować różne formaty. Jednak aby dokładnie zrozumieć, jakie formaty informacji są przesyłane w sieci, należy poznać nazwy tych formatów. Problem z nazwami jest bardzo poważny, gdyż nie zawsze są one używane konsekwentnie, co może powodować spore nieporozumienia.

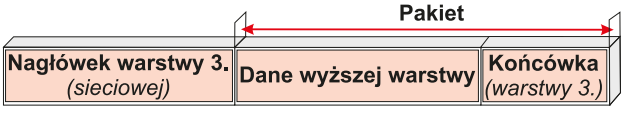
Ramka (ang. *frame*) to jednostka informacji, której źródłem i przeznaczeniem jest warstwa *łącza danych* (warstwa 2.) modelu OSI. Ramka składa się z elementów pokazanych na rysunku 3.43.



Rysunek 3.43. Struktura ramki

Nagłówek i końcówka zawierają informację sterującą przeznaczoną dla warstwy *łącza danych* w stacji odbiorczej. Można powiedzieć, że dane z wyższej warstwy są opakowane (kapsułkowane) przez nagłówek i końcówkę warstwy *łącza danych*.

Pakiet (ang. *packet*) to jednostka informacji, której źródłem i przeznaczeniem jest warstwa *sieciowa* (warstwa 3.) modelu OSI. Pakiet składa się z elementów pokazanych na rysunku 3.44.

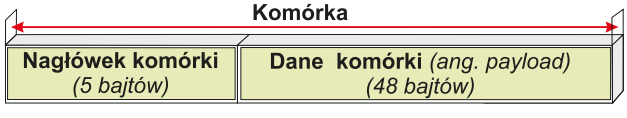


Rysunek 3.44. Struktura pakietu

Nagłówek i końcówka zawierają informację sterującą przeznaczoną dla warstwy 3. w stacji odbiorczej. Można powiedzieć, że dane z wyższej warstwy są otoczone przez nagłówek i końcówkę warstwy 3.

Komórka (ang. *cell*) to jednostka informacji złożona z dwóch elementów:
- nagłówka komórki o długości 5 bajtów,
- danych komórki o długości 48 bajtów.

Komórka, którą schematycznie pokazano na rysunku 3.45, ma zawsze stałą długość 53 bajtów i odnosi się do warstwy *łącza danych* (warstwy 2.) modelu OSI.



Rysunek 3.45. Struktura komórki

Datagram to jednostka informacji, której źródłem i przeznaczeniem jest warstwa *sieciowa* (warstwa 3.) modelu OSI, używająca bezpołączeniowej obsługi sieci.

Segment to jednostka informacji, której źródłem i przeznaczeniem jest warstwa *transportowa* (warstwa 4.) modelu OSI.

Komunikat to jednostka informacji, której źródłem i przeznaczeniem jest zwykle warstwa *aplikacji* (warstwa 7.) modelu OSI.

Jak w praktyce przebiega przekazywanie strumienia danych?

Strumień danych wypływa z warstw 7., 6. i 5. Dopiero w warstwie *transportowej* (4.) zostaje zamieniony na *segmenty*. Każdy z segmentów ma nagłówek warstwy 4., który zostaje nadany przez tę właśnie warstwę. Jest to między innymi numer sekwencyjny,

potrzebny do ustalenia kolejności podawania danych. Kolejna warstwa, czyli *sieci* (3.), jest odpowiedzialna za podzielenie danych na równe porcje, zwane *pakietami*; podobnie jak w warstwie *transportowej* dodawany jest nagłówek, tylko teraz nagłówek ten zawiera adresy nadawcy i odbiorcy. Taki pakiet wędruje do warstwy *łącza danych* (2.) i, podobnie jak w poprzednich warstwach, dopisywany jest kolejny nagłówek warstwy 2., a dane dzielone są na *ramki*. Ramki zawierają między innymi informacje o adresie fizycznym MAC. Kolejna warstwa zamienia dane na ciąg znaków binarnych.

3.4.3. Model protokołu TCP/IP

Definicja

TCP/IP (ang. *Transmission Control Protocol/Internet Protocol* — protokół kontroli transmisji) jest pakietem najbardziej rozpowszechnionych protokołów komunikacyjnych współczesnych sieci komputerowych. TCP/IP to najczęściej wykorzystywany standard sieciowy, stanowiący podstawę współczesnego internetu. Nazwa pochodzi od dwóch najważniejszych jego protokołów: **TCP** oraz **IP**.

Protokół internetu (IP) został opracowany dla Departamentu Obrony USA, który szukał sposobu połączenia różnych rodzajów komputerów i sieci je obsługujących w jedną, wspólną sieć. Osiągnięto to za pomocą warstwowego protokołu, który odizolował aplikacje od sprzętu sieciowego. Protokół ten, znany jako model TCP/IP, używa modelu nieco różniącego się od modelu OSI.

Stos protokołów TCP/IP zawiera cztery warstwy funkcjonalne, nawiązujące do siedmiu warstw wzorcowego modelu OSI; są to:

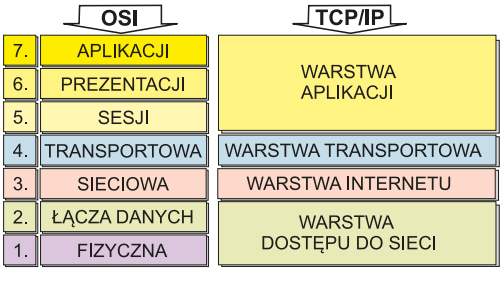
- warstwa aplikacji,
- warstwa transportowa (TCP),
- warstwa internetu (IP),
- warstwa dostępu do sieci.

Rysunek 3.46 przedstawia porównanie modeli OSI i TCP/IP.

Podobnie jak w modelu OSI, kolejne warstwy dołączają lub usuwają (w zależności od tego, w którą stronę przesuwają się dane) własne nagłówki.

Dane przekazywane z TCP do IP nazywane są **segmentem TCP**.

Dane, które IP przesyła do interfejsu sieciowego, nazywane są **datagramem IP**.

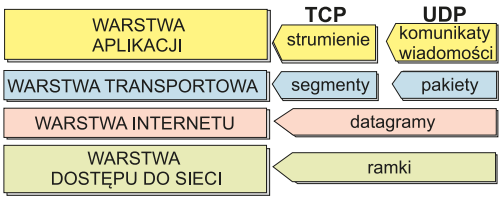


Rysunek 3.46. Porównanie modeli OSI i TCP/IP

Ciąg bitów przepływający w sieci Ethernet nazywany jest **ramką**.

Rysunek 3.47 przedstawia nazwy jednostek danych w kolejnych warstwach modelu TCP/IP.

Funkcje spełniane przez poszczególne warstwy modelu OSI nie pokrywają się z funkcjami warstw modelu TCP/IP. Dlatego też poziomy warstw TCP/IP nie odpowiadają poziomom warstw modelu OSI. Na rysunku obok została przedstawiona jedynie zbieżność nazewnictwa warstw w obu modelach.



Rysunek 3.47. Nazwy jednostek danych w kolejnych warstwach modelu TCP/IP

Zadania warstw w TCP/IP

Warstwa aplikacji (ang. *process layer*), nazywana również warstwą procesową lub warstwą programów użytkowych, to najwyższy poziom, w którym pracują niezbędne dla użytkownika aplikacje, takie jak serwer WWW czy przeglądarka internetowa. Obejmuje ona zestaw gotowych protokołów, które aplikacje wykorzystują do przesyłania różnego typu informacji w sieci, czyli do wysyłania lub odbierania danych w postaci pojedynczych komunikatów lub strumienia bajtów.

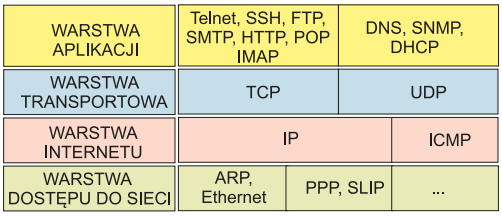
Warstwa transportowa (ang. *host-to-host layer*) — jej podstawowym zadaniem jest zapewnienie przesyłania danych oraz kierowanie właściwych informacji do odpowiednich aplikacji. Opiera się to na wykorzystaniu portów określonych dla każdego połączenia. W jednym komputerze może istnieć wiele aplikacji wymieniających dane z tym samym odbiorcą w sieci i nie nastąpi wymieszanie się przesyłanych przez nie danych. To właśnie ta warstwa nawiązuje i kończy połączenia między komputerami oraz gwarantuje pewność transmisji. W tym celu organizuje wysyłanie przez odbiorcę potwierdzenia otrzymania danych oraz ponowne wysyłanie przez nadawcę utraconych pakietów.

Warstwa internetu (ang. *internet layer*) lub warstwa protokołu internetowego (ang. *internet protocol layer*) odpowiada za obsługę komunikacji w sieci. W tej warstwie przetwarzane są datagramy mające adresy IP. Ustalana jest odpowiednia droga do docelowego komputera w sieci. Przyjmuje ona pakiety z warstwy transportowej razem z informacjami identyfikującymi odbiorcę, opakowuje (kapsułkuje) pakiet w datagramie IP, wypełnia jego nagłówek i przekazuje datagram do interfejsu sieciowego. W niektórych urządzeniach sieciowych jest to warstwa najwyższa. Są to routery, które zajmują się kierowaniem ruchu w sieci. Proces szukania przez routery właściwej drogi określa się jako **trasowanie**.

Warstwa dostępu do sieci lub **warstwa fizyczna** (ang. *network access layer*) jest najniższą warstwą i to ona zajmuje się przekazywaniem danych przez fizyczne połączenia między urządzeniami sieciowymi. Najczęściej są to karty sieciowe lub modemy. Warstwa dostępu realizuje następujące funkcje:

- sterowanie przepływem,
- ustalanie priorytetów,
- zabezpieczanie przed błędami,
- utajnianie przesyłanych danych.

Na rysunku 3.48 znajduje się zbiór usług i grupa przykładowych protokołów, które są powiązane z odpowiednimi warstwami protokołem TCP/IP.



Rysunek 3.48. Zbiór wybranych protokołów i usług poszczególnych warstw TCP/IP

W warstwie transportowej działa protokół TCP, protokół IP zaś znajduje się w warstwie internetu. Warstwa transportowa odpowiedzialna jest za znalezienie drogi łączącej dwa hosty oraz określenie logicznej adresacji.

Definicja

Host to każdy komputer podłączony do internetu lub innej sieci używającej protokołu TCP/IP i mający unikatowy *adres IP*.

3.4.4. Adres IP (ang. *Internet Protocol address*)

Definicja

Adres IP to unikatowy numer przyporządkowany do każdego urządzenia sieci komputerowych, na przykład komputera, drukarki, routera.

IPv4

Adres IPv4 składa się z 32 bitów, a więc z 4 oktetów (oktet = 8 bitów). Aby przedstawić adres w postaci czytelnej dla wszystkich użytkowników i łatwiejszej do zapamiętania, zazwyczaj wartość każdego oktetu zapisuje się osobną liczbą dziesiętną, poszczególne oktety oddzielając kropkami, na przykład 83.3.250.66 (każda liczba dziesiętna odpowiada 8 bitom adresu IP).

Ten ogólnie przyjęty sposób zapisu adresu IP, czytelny dla użytkownika, jest znany jako *format bajtowo-dziesiętny*, a taki zapis nosi nazwę **notacji dziesiętnej z kropkami** (ang. *dotted decimal notation*).

Na przykład 32-bitowy adres 10000010 00001010 00001100 00011111 jest zapisany jako 130.10.12.31.

Fenomen tego systemu adresowania polega na tym, że umożliwia on wyznaczanie tras pakietów. Jest to możliwe dzięki temu, że adres IP zawiera informację o tym, do jakiej sieci jest włączony dany komputer, a także jednoznaczny adres komputera w tej sieci. Adres IP jest używany przy wszystkich operacjach związanych z wymianą informacji z danym komputerem.

W obrębie adresu wyróżnia się dwa składniki: *identyfikator sieciowy* (ang. *network id*) oraz *identyfikator komputera* (ang. *host id*). Istnieją różne klasy adresowe, o różnej długości tych dwóch składników:

- *Klasa A* to adresy o 8-bitowym identyfikatorze sieciowym i 24-bitowym identyfikatorze hosta, czyli 8/24 (*network id/host id*).
- *Klasa B* to adresy o 16-bitowym identyfikatorze sieciowym i 16-bitowym identyfikatorze hosta, czyli 16/16 (*network id/host id*).
- *Klasa C* to adresy o 24-bitowym identyfikatorze sieciowym i 8-bitowym identyfikatorze hosta, czyli 24/8 (*network id/host id*).

Bity w adresie IP są zatem interpretowane jako: *[adres sieci, adres hosta]*.

Wszystkie hosty w danej sieci mają ten sam adres sieci i unikatowy adres hosta. Dwa komputery w różnych sieciach muszą mieć inne adresy sieciowe, ale mogą mieć ten sam adres hosta.

O przynależności adresu IP do danej klasy adresowej świadczą pierwsze bity adresu IP (pierwszy oktet). Na rysunku 3.49 pokazano poszczególne klasy adresowe wraz z pierwszymi bitami adresu oraz z zakresami adresów należących do każdej z klas (adres najniższy i adres najwyższy sieci).

Klasa	Struktura bitowa adresu IP	Zakres adresów	Zakres adresów IP dostępnych dla użytkowników
A	0 Sieć (7 bitów) Host (24 bity)	0.0.0.0 127.255.255.255	1.0.0.0 126.0.0.0
B	1 0 Sieć (14 bitów) Host (16 bitów)	128.0.0.0 191.255.255.255	128.0.0.0 191.254.0.0
C	1 1 0 Sieć (21 bitów) Host (8 bitów)	192.0.0.0 223.255.255.255	192.0.0.0 223.255.254.0
D	1 1 1 0 adres grupowy (28 bitów)	224.0.0.0 239.255.255.255	żaden
E	1 1 1 1 zarezerwowane na przyszłość	240.0.0.0 255.255.255.255	żaden

Rysunek 3.49. Klasy adresów IPv4

Obserwując pierwszy oktet adresu, możemy stwierdzić, do jakiej klasy należy dany adres, a w efekcie określić, ile bitów będzie adresowało sieć, ile — sam komputer. Zwróćmy uwagę, że aby określić przynależność do jednej z trzech zasadniczych klas (A, B, C), wystarczą dwa pierwsze bity.

Adresów *klasy A*, wykorzystywanych przez duże sieci, jest niewiele. W tej klasie może istnieć zaledwie 126 sieci, z których każda ma ponad 16 milionów hostów (na adresy sieci przeznaczonych jest 7 bitów, jednak skrajne wartości, czyli `0` oraz `127`, mają specjalne znaczenie, stąd $2^7-2 = 126$ adresów sieci).

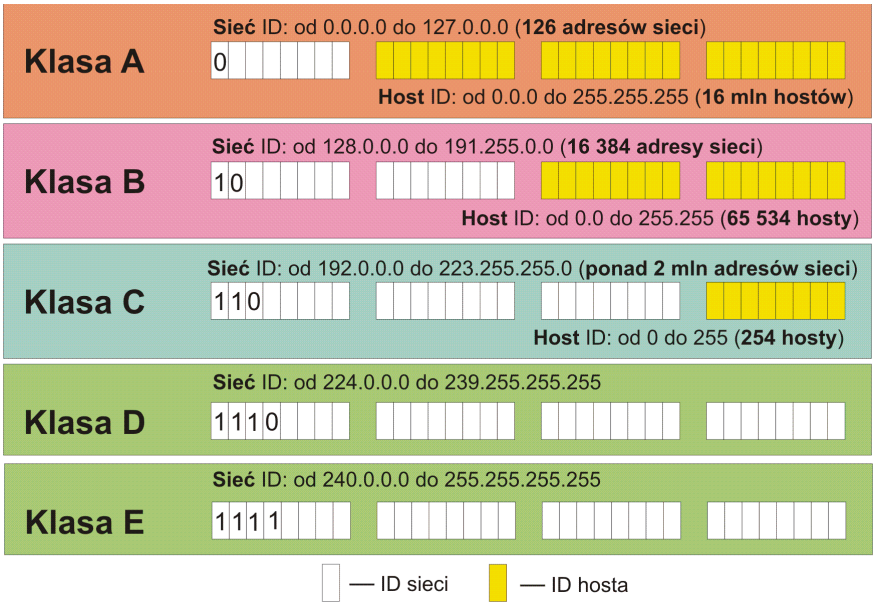
Klasa B przeznaczona jest dla sieci średniej wielkości. 14 bitów określa sieć, 16 bitów zaś można wykorzystać do adresacji hostów. W efekcie otrzymujemy 16 384 (2^{14}) sieci, które mogą mieć do 65 534 ($2^{14}-2 = 65\,536-2 = 65\,534$) hostów każda.

W *klasie C* sieć adresowana jest za pomocą 21 bitów — daje to 2^{21} sieci (2 097 152), ale w każdej z nich może być co najwyżej 2^8, czyli 256 adresów IP, jednak tylko 254 z nich można przeznaczyć na hosty ($2^8-2 = 254$).

Adres *klasy D* (ang. *multicast address*) ma specjalne znaczenie — jest używany w sytuacji, gdy zachodzi jednoczesna transmisja do większej liczby urządzeń.

Adresy *klasy E* mają specjalne znaczenie — są używane w celach eksperymentalnych. Obecnie adresy rozpoczynające się od `1111` są zarezerwowane do wykorzystania w przyszłości. Prawdopodobnie, jeśli dla kolejnej klasy adresów zostanie przydzielony jakiś sposób ich wykorzystania, to definicja klas zostanie zmodyfikowana tak, że klasa E będzie się zaczynała od `11110`, a nowa zdefiniowana *klasa F* (jako rezerwa na przyszłość) wyróżniana będzie początkowymi bitami w postaci `11111`.

Szczegółowy rozkład ID sieci i ID hosta ilustruje rysunek 3.50.



Rysunek 3.50. Tablica adresów klasowych IPv4

Jeszcze nie tak dawno sądzono, że pula około 4 miliardów adresów IPv4 na długo zapewni niezawodne działanie internetu. Szacuje się jednak, że każdego dnia przybywa kilkanaście tysięcy nowych hostów podłączonych do sieci. Problem z możliwością wyczerpania puli adresów IP jest bardzo poważny.

Jednak internet ciągle działa, a nowe komputery wciąż podłączane są do sieci. Jak to możliwe? I dlaczego mimo ograniczonej liczby adresów IPv4 nadal można podłączać komputery do internetu?

Problem ten rozwiązano na kilka sposobów:

- Pierwszy sposób polegał na opracowaniu nowego protokołu internetowego *IPv6*. Jest to rozwiązanie, które radykalnie zwiększy przestrzeń adresową z 32 bitów w IPv4 aż do 128 bitów w IPv6. Jednakże wprowadzenie IPv6 zajmie kilka lat, a migracja z IPv4 do IPv6 nie zostanie przeprowadzona w krótkim czasie.

  - IPv4 to 4*8 bitów = 32 bity,
  - IPv6 to 8*16 bitów = 128 bitów.
- Krótkoterminowym rozwiązaniem jest koncepcja bezklasowego routingu międzydomenowego *CIDR* (ang. *Classless Inter-Domain Routing*), która umożliwia bardziej wydajny sposób przydzielania przestrzeni adresowej IPv4 przez rezygnację z tradycyjnej koncepcji klas A, B i C.
- Innym sposobem miało być zastosowanie puli adresów IP prywatnych w sieci lokalnej i podłączenie jej do internetu poprzez jeden **adres publiczny**. Proces tłumaczenia adresu z puli prywatnej na publiczną i odwrotnie to translacja adresów, czyli NAT (ang. *Network Address Translation*). W praktyce oznacza to, że korzystając z jednego adresu publicznego IP, możemy podłączyć do internetu wiele komputerów, tak jak to ma miejsce na przykład w szkole.

Adres rozgłoszeniowy

Adres, w którym wszystkie bity w części adresu przeznaczonego dla hostów przyjmą wartość `1`, nazywamy *adresem rozgłoszeniowym* (ang. *broadcast*). Taki adres jest zarezerwowany w każdej sieci i nie może być użyty do adresacji hostów. Wysłanie wiadomości na adres rozgłoszeniowy jest wysłaniem wiadomości do wszystkich, czyli *komunikatem sieciowym* (*rozgłoszeniowym*). Na przykład adres `152.23.255.255` oznacza adres rozgłoszeniowy w sieci `152.23.0.0` (klasa B). Każdy host pracujący w sieci IP odbiera komunikaty przychodzące na jego własny adres IP oraz na adres typu *broadcast*.

Adres sieci

Adres, w którym wszystkie bity w części adresu przeznaczonego dla hostów przyjmą wartość `0`, nazywamy *adresem sieci* (ang. *network address*). Taki adres jest adresem zarezerwowanym w każdej sieci i tak jak adres *broadcast* nie może być użyty do adresacji hostów.

Dlatego zawsze liczba hostów jest mniejsza o 2 od całej puli dostępnych adresów IP. Przykładowo w każdej sieci klasy C jest do dyspozycji 256 adresów IP, jednak dwa z nich są adresami specjalnymi (*sieci* i *rozgłoszeniowym*), więc dla hostów pozostają 254 adresy. Ostatni oktet przeznaczony dla hostów w klasie C, czyli 8 bitów, pozwala na podłączenie $2^8-2 = 256-2 = 254$ hostów. Dla klasy B to odpowiednio: $2^{16}-2$ hosty, a dla klasy A: $2^{24}-2$ hosty.

Adresy specjalne

Adres pętli zwrotnej

Adres `127.0.0.1` należący do sieci `127.0.0.0`, nazywany z języka angielskiego *adresem loopback*, nie łączy się z żadną siecią. Jest używany głównie w celu testowania oprogramowania TCP/IP na maszynie, która nie musi być przyłączona do sieci. Adres ten może być używany na przykład do adresowania odwołań oprogramowania klienta z serwerem uruchomionym na tej samej maszynie, bez konieczności używania zewnętrznego adresu IP hosta.

Specyfikacja protokołu IP wymaga, aby ten adres (jak i cała sieć `127.0.0.0`) nigdy nie był przypisywany do adresacji hosta w sieci. Jeśli tak się zdarzy, adresy te będą odrzucane przez każdy host lub router.

Adres domyślny

Adres domyślny (ang. *default*) `0.0.0.0` to adres specjalny, który oznacza lokalną sieć. Adres `0.0.0.88` oznacza host z numerem `88` w tej sieci.

Adresy niepubliczne (prywatne)

Adresów *niepublicznych*, inaczej zwanych *nierutowalnymi*, nie można używać w internecie. Są one przeznaczone do budowy sieci lokalnych. Jeśli sieć publiczna korzysta z adresów niepublicznych (prywatnych), a hosty mają mieć dostęp do internetu, musi zostać zastosowane maskowanie adresów niepublicznych, inaczej zwane *NAT-owaniem*.

Dokument *RFC 1918* określa, jakie adresy IP mogą być użyte wewnątrz prywatnej sieci. Zarezerwowane są dla nich trzy grupy adresów IP klas A, B, C. Wydzielone odpowiednie pule adresowe przeznaczone na adresy niepubliczne zestawiono na rysunku 3.51.

Klasa	Zakresy adresów prywatnych w danych klasach	
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

Rysunek 3.51. Zakresy adresów niepublicznych (prywatnych)

Maski sieciowe

Maska sieci składa się, podobnie jak adres IP, z 4 oktetów. Używana jest do wydzielenia z adresu IP części odpowiadającej za identyfikację sieci i części odpowiadającej za identyfikację komputera.

Klasa adresów sieciowych wyznacza maskę sieciową, co pokazano na rysunku 3.52.

Klasa	Bity — maska podsieci	Notacja dziesiętna
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

Rysunek 3.52. Domyślne maski dla klas adresowych A, B, C

Należy pamiętać, że maska (jak również adres IPv4) zapisana jest w postaci binarnej, a zapis kropkowo-dziesiętny stosuje się dla wygody użytkowników.

Adresowanie bezklasowe IPv4

Podział adresów na klasy adresowe spowodował, że dużo adresów IP się marnowało. Rozważmy dla przykładu średnich rozmiarów przedsiębiorstwo, które potrzebuje 300 adresów IP. Adres klasy C, dający 254 adresy hostów, jest niewystarczający. Wykorzystanie dwóch adresów klasy C dostarczy więcej adresów, niż potrzeba, ale w wyniku tego w ramach przedsiębiorstwa powstaną dwie odrębne sieci. Z kolei zastosowanie adresu

klasy B zapewni potrzebne adresy w ramach jednej sieci, ale zmarnują się w ten sposób 65 234 (65 534–300) adresy.

Od 1997 roku podział na klasy sieci jest już nieaktualny. Obecnie adresy IPv4 są przydzielane bez specjalnego zwracania uwagi na klasy sieci — według założeń bezklasowego routingu międzydomenowego *CIDR* (ang. *Classless Inter-Domain Routing*).

Zaczęto więc nadawać maski, które nie są maskami według klas adresów IP (czyli takich, w których liczba jedynek jest wielokrotnością oktetów, zob. rysunek 3.52). Zwiększenie liczby jedynek przy takiej samej liczbie bitów (32) umożliwiło uzyskanie maski, na przykład `11111111 11111111 11111111 11110000` (`255.255.255.240`), która pozwala na objęcie podsiecią 14 hostów (16 numerów IP). Rysunek 3.53 przedstawia wszystkie możliwe podsieci dla zakresu od 2 do 254 hostów. Wprowadzenie pozaklasowej maski podsieci pozwoliło na ekonomiczne wykorzystanie numerów IPv4.

Maska *(dziesiętnie)*	Maska *(binarnie)*	Liczba podsieci	Liczba hostów w podsieci
255.255.255.0	11111111 11111111 11111111 00000000	1	254
255.255.255.128	11111111 11111111 11111111 10000000	2	126
255.255.255.192	11111111 11111111 11111111 11000000	4	62
255.255.255.224	11111111 11111111 11111111 11100000	8	30
255.255.255.240	11111111 11111111 11111111 11110000	16	14
255.255.255.248	11111111 11111111 11111111 11111000	32	6
255.255.255.252	11111111 11111111 11111111 11111100	64	2
255.255.255.254	11111111 11111111 11111111 11111110	128	0

Rysunek 3.53. Wszystkie możliwe podsieci dla zakresu od 2 do 254 hostów

Host jest zatem określany przez adres IP i maskę podsieci.

Adres sieci w adresowaniu bezklasowym

Znając adres IP hosta oraz jego maskę, można określić *adres sieci* (ang. *network address*). W tym celu należy wykonać funkcję `AND` pomiędzy adresem IP a jego maską sieci. Zasadę koniunkcji (iloczynu logicznego) przypomina rysunek 3.54.

Kombinacja bitów	Wartość
1 AND **1**	**1**
1 AND **0**	**0**
0 AND **1**	**0**
0 AND **0**	**0**

Rysunek 3.54. Koniunkcja bitów jest prawdziwa wtedy i tylko wtedy, gdy oba zdania p i q są prawdziwe

Adres sieciowy jest **bitowym iloczynem** maski sieciowej i adresu IP hosta.

Czasami można spotkać skrótowo zapisany adres IP w postaci: `197.3.182.0/24`, gdzie część stojąca przed znakiem `/` jest adresem IP, natomiast liczba `24` jest skrótowo zapisaną maską sieciową. Jest to liczba ciągu bitów w masce sieciowej mających wartość `1`. Przykład nasz można zatem zapisać na trzy równoznaczne sposoby: binarnie, kropkowo-dziesiętnie lub w zapisie skróconym w tzw. notacji CIDR:

IP: `11000101 00000011 01110101 00000000`
z maską `11111111 11111111 11111111 00000000`,

IP: `197.3.182.0` z maską `255.255.255.0`,

IP/maska: `197.3.182.0/24`.

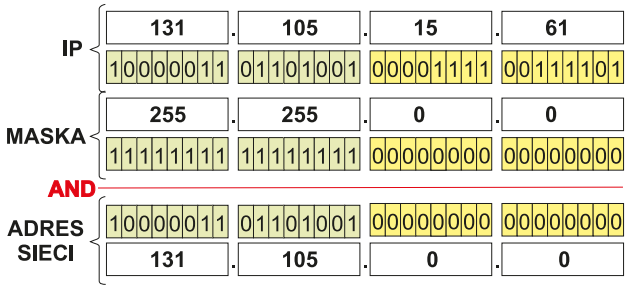
Przykład 3.1.

Dla adresu IP: `131.105.15.61/16` określ jego klasę, adres sieci oraz adres hosta.

Określenie klasy polega na sprawdzeniu pierwszych bitów pierwszego oktetu badanego adresu IP. W tym przypadku należy sprawdzić wartość `131`, zamieniając ją na zapis binarny: `10000011`. Dwa pierwsze bity wystarczą do określenia klasy. Pozycja zera na drugim miejscu świadczy o tym, że adres IP `131.105.15.61` należy do klasy adresowej B.

Analizując maskę sieci, można określić adres sieci oraz adres komputera (hosta). Maska definiuje bowiem podział adresu IP na część złożoną z adresu sieci i z adresu komputera (hosta) w danej sieci komputerowej. Zapis IP w postaci `131.105.15.61/16` oznacza, że jest to maska domyślna `255.255.0.0`.

Wykonując binarne `AND` adresu IP i maski, otrzymasz adres sieci. Obliczenia zawarto na rysunku 3.55.



Rysunek 3.55. Rozróżnianie adresu sieci i hosta na podstawie maski

Odpowiedź: Ten adres należy do sieci klasy B, adres sieci to 131.105.0.0, a adres hosta w tej sieci to 0.0.15.61.

Ćwiczenie 3.11. Znając adresy IP oraz maski dwóch komputerów, sprawdź, czy oba komputery należą do tej samej sieci.

A) Pierwszy komputer ma adres IP `172.21.10.10/16`,
drugi komputer — adres IP `172.21.11.102/16`.

B) Pierwszy komputer ma adres IP `192.168.13.25/24`,
drugi komputer — adres IP `192.168.15.23/24`.

C) Pierwszy komputer ma adres IP `192.168.1.25/24`,
drugi komputer — adres IP `192.168.1.126/24`.

Adres rozgłoszeniowy w adresowaniu bezklasowym

Przypomnijmy, że adres rozgłoszeniowy jest zarezerwowanym adresem IP. Wszystkie komputery w sieci nasłuchują pakietów kierowanych na ten adres. Jeżeli chcemy wysłać pakiet adresowany do wszystkich komputerów w danej sieci, korzystamy właśnie z adresu rozgłoszeniowego (ang. *broadcast*).

Aby określić adres rozgłoszeniowy (broadcast) na podstawie adresu IP hosta i jego maski, wykonujemy działanie logiczne negujące `IP XOR maska`, czyli `NOT(IP XOR maska)`. Zasadę działania alternatywy wykluczającej ilustruje rysunek 3.56.

Kombinacja bitów	Wartość
1 XOR **1**	**0**
1 XOR **0**	**1**
0 XOR **1**	**1**
0 XOR **0**	**0**

Rysunek 3.56. Alternatywa wykluczająca xor jest prawdziwa wtedy i tylko wtedy, gdy dokładnie jedno ze zdań p lub q jest prawdziwe

Przykład 3.2.

Oblicz adres rozgłoszeniowy dla IP `195.116.241.160/27`.

IP	`11000011`	`01110100`	`11110001`	`10100000`	
maska	`11111111`	`11111111`	`11111111`	`11100000`	
xor	`00111100`	`10001011`	`00001110`	`01000000`	*— wynik działania logicznego* `xor`
not	`11000011`	`01110100`	`11110001`	`10111111`	
	`195`	`116`	`241`	`191`	*— negacja wyniku, będąca adresem rozgłoszeniowym*
broadcast		`195.116.241.191`			*— adres rozgłoszeniowy w zapisie kropkowo-dziesiętnym*

Ćwiczenie 3.12. Dla przykładowego adresu IP i maski `218.131.18.56/22` oblicz adres sieci i adres rozgłoszeniowy broadcast.

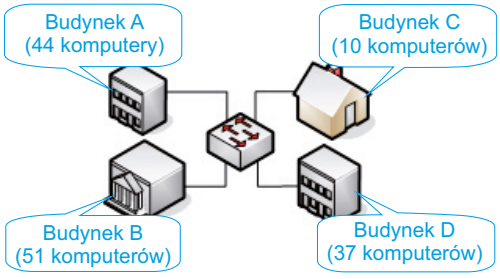
Dzielenie sieci — podsieci

Pierwotnie adresowanie IP wymagało, by każda sieć miała unikatową część adresującą sieć, w której można przyłączyć określoną liczbę hostów. Dla przykładu adres hosta: `198.200.55.X` jest adresem klasy C, gdzie `X` jest liczbą z zakresu od `1` do `254`. Nawet jeśli w sieci jest tylko 40 komputerów, to bez stosowania podsieci wszystkie adresy od `198.200.55.1` do `198.200.55.254` byłyby zarezerwowane dla tej niewielkiej liczby komputerów. W ten sposób aż 214 adresów (254–40) pozostaje niewykorzystanych. Dzięki zastosowaniu podsieci sieć ta może wykorzystać tylko ¼ adresów z dostępnej puli 254 adresów (zgodnie z podziałem na 4 podsieci, co wyliczono na rysunku 3.53). Będą to adresy z zakresu pierwszej podsieci, czyli od `198.200.55.1` do `198.200.55.62`. Rezerwujemy w tym przypadku 64 adresy IP dla 62 hostów. Po przydzieleniu 40 adresów pozostaną nam w rezerwie, jako niewykorzystane, 22 adresy IP dla kolejnych hostów. Pozostałe adresy trzech podsieci (192 adresy IP) mogą być wykorzystane przez innych użytkowników. Jako wyjaśnienie niech posłuży przykład 3.3.

Przykład 3.3.

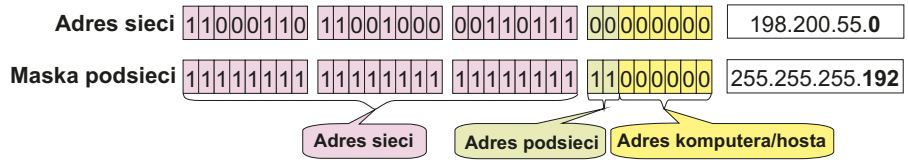
Należy podzielić sieć klasy C o adresie `198.200.55.0/24` na podsieci obsługujące poszczególne budynki pokazane na rysunku 3.57.

Możemy utworzyć podsieci, w których liczba hostów przyjmie następujące wartości: 2, 6, 14, 30, 62, 126, 254. Z analizy rysunku 3.57 wynika, że należy utworzyć cztery podsieci o 62 dostępnych adresach hostów w każdej z nich.



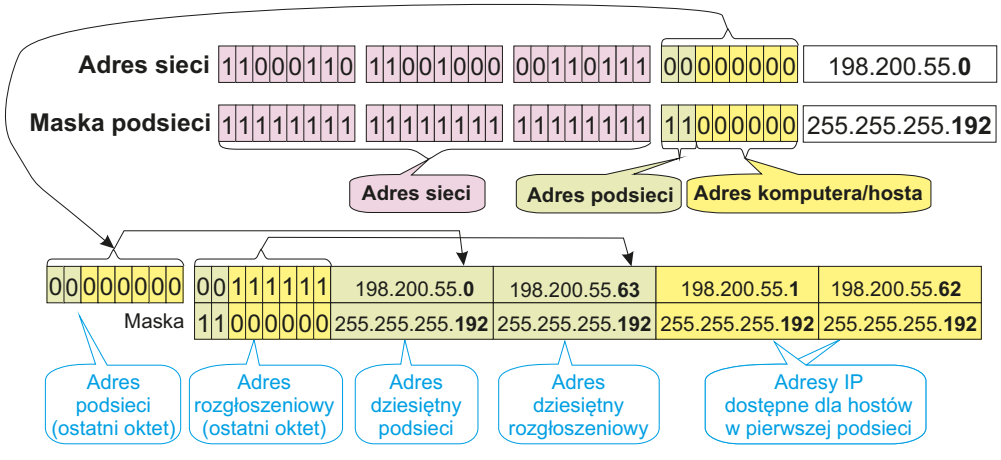
Rysunek 3.57. Rozmieszczenie komputerów w czterech budynkach wydziałów uczelni

Ostatni oktet adresu komputera w sieci klasy C jest przeznaczony do adresowania hostów. Ze względu na fakt, że chcemy utworzyć cztery podsieci, dla adresu podsieci należy wykorzystać pierwsze dwa bity z czwartego oktetu adresu C. Podział ten spowoduje wprowadzenie maski, w której pierwsze dwa bity przyjmą wartość `1`, jak na rysunku 3.58.



Rysunek 3.58. Ustalenie maski w celu utworzenia czterech podsieci

Znając ostatni oktet maski dla podziału sieci na cztery podsieci, możemy ustalić maskę na `255.255.255.192` (11000000 = 192). Po ustaleniu maski można określić adresy podsieci. Adresy czterech podsieci są związane z dwoma pierwszymi bitami czwartego oktetu adresu IP, który może przyjąć cztery różne wartości (`00`, `01`, `10`, `11`). Szczegółowe obliczenia związane z pierwszą podsiecią przedstawione są na rysunku 3.59. Ponieważ wszystkie obliczenia są przeprowadzane w tym przypadku na ostatnim oktecie, przeznaczonym dla hostów (klasa C), pokazano jedynie bity tego czwartego oktetu.

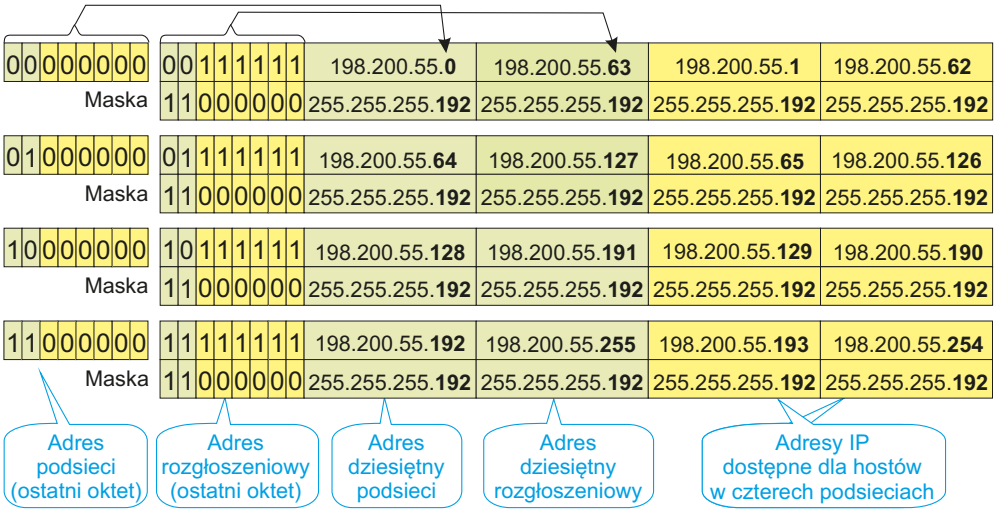


Rysunek 3.59. Ustalenie adresów w pierwszej podsieci

Mimo że w pierwszej podsieci przeznaczyliśmy na adresację hostów 6 bitów, do sieci możemy podłączyć maksymalnie 62 komputery (adresy skrajne są zarezerwowane: `198.200.55.1` to adres sieci, natomiast `198.200.55.63` to adres rozgłoszeniowy). Dla potrzeb zadania jest to wystarczająca liczba hostów.

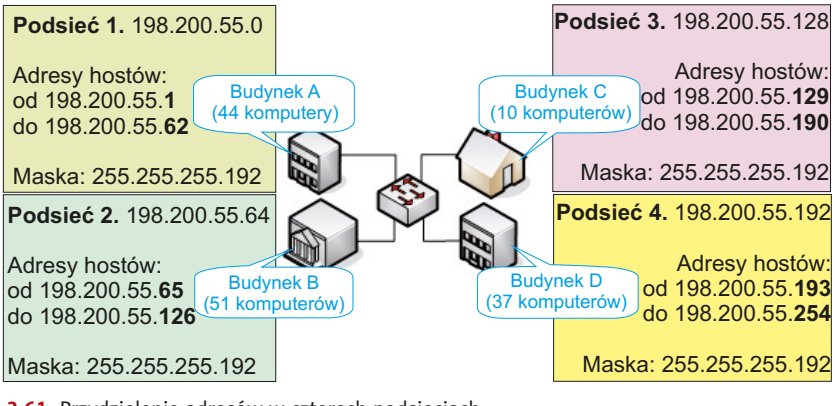
Na 6 bitach przeznaczonych dla hosta można przydzielić 2^6-2 adresów (na 5 bitach — 2^5-2 itd.).

Podobnie przeprowadzone obliczenia pozwolą zdefiniować trzy pozostałe podsieci, co ilustruje rysunek 3.60.



Rysunek 3.60. Ustalenie adresów dla wszystkich czterech podsieci

Na rysunku 3.61 przedstawione jest rozmieszczenie poszczególnych podsieci. Podsieć 3. w tym przypadku nieefektywnie gospodaruje naszymi adresami. Z puli dostępnych adresów tracimy aż 62–10 = 52 adresy. Zastanów się, czy można odzyskać część utraconych adresów IP.



Rysunek 3.61. Przydzielenie adresów w czterech podsieciach

Etapy tworzenia podsieci

1. Ustal, jakiej klasy adres trzeba podzielić (w przykładzie był to adres klasy C).
2. Ustal, ile podsieci trzeba utworzyć (w przykładzie cztery).
3. Ile bitów potrzeba na ustalenie adresu podsieci?

   $$liczba\_podsieci = 2^{liczba\_bitów\_przeznaczona\_na\_podsieć}$$

   Pamiętaj, że bity podsieci można „odbierać" wyłącznie z części hosta, nigdy z części sieci. W klasie C mamy więc do dyspozycji 8 bitów. Liczba możliwych podsieci może wynosić: 2, 4, 8, 16, 32, 64 lub 128. Pamiętając o klasie adresowej (w tym przypadku C), ustal, ile bitów ma zostać przydzielonych do tworzenia podsieci. Dla 4 podsieci potrzebujesz 2 bitów. Na tej podstawie ustalamy maskę podsieci. Do maski sieci dodajemy dwa bity: `11111111 11111111 11111111 11000000`, co odpowiada zapisowi dziesiętnemu `255.255.255.192`. Ciąg 26 jedynek w masce można zapisać w notacji CIDR jako `/26`.
4. Ustal adresy podsieci. Na podstawie ustawień bitów w części podsieci obliczamy adresy podsieci. W naszym przykładzie ostatni oktet adresów czterech tworzonych podsieci może przyjąć następujące wartości: **00**000000, **01**000000, **10**000000, **11**000000 — co w zapisie dziesiętnym wynosi, odpowiednio: **0**, **64**, **128**, **192**. Teraz można już zapisać adresy wszystkich podsieci. W przykładzie będą to: `198.200.55.0/26`, `198.200.55.64/26`, `198.200.55.128/26` i `198.200.55.192/26`.
5. Ustal adresy hostów w utworzonych podsieciach. W każdej podsieci najniższy adres stanowi adres sieci (ustalony w punkcie 4.). Najwyższy adres każdej podsieci jest adresem rozgłoszeniowym, pozostałe adresy mogą być przydzielone hostom. W przykładzie będą to adresy:

   od `198.200.55.1/26` do `198.200.55.62/26`,

   od `198.200.55.65/26` do `198.200.55.126/26`,

   od `198.200.55.129/26` do `198.200.55.190/26`,

   od `198.200.55.193/26` do `198.200.55.254/26`.

Korzyści ze stosowania podsieci

Adresowanie z podziałem na podsieci zapewnia wiele korzyści. Do najważniejszych należy zaliczyć:

- efektywniejsze wykorzystanie puli adresów IP;
- możliwość odseparowania podsieci;
- możliwość komunikacji pomiędzy urządzeniami w różnych podsieciach poprzez routery;
- wyodrębnienie podsieci powoduje zmniejszenie ruchu w sieci.

Ćwiczenie 3.13. Podziel adres sieciowy IP: `208.182.34.0/24` na 16 podsieci.

Ćwiczenie 3.14. Na podstawie adresu IP hosta i maski podsieci ustal jego klasę oraz oblicz adres rozgłoszeniowy, adres sieci i adres podsieci. Zapisz ten adres w notacji CIDR.

IP: `83.3.249.66`.

Maska podsieci: `255.255.255.248`.

Przykład 3.4.

Czy adres IP hosta `207.13.47.48` z maską `255.255.255.248` jest prawidłowy? Odpowiedź uzasadnij.

Sprawdzimy adres sieci dla tego IP oraz dla tej maski.

IP	`207.13.47.48`	`11001111.00001101.00101111.00110000`
Maska	`255.255.255.248`	`11111111.11111111.11111111.11111000`
Adres sieci	`207.13.47.48`	`11001111.00001101.00101111.00110000`

Przy masce `255.255.255.248` (29 jedynek) podany w ćwiczeniu adres IP można zapisać w notacji CIDR następująco: `207.13.47.48/29`. W tym przypadku na adresy hostów przeznaczone są 3 bity (8 kolejnych adresów). Dwa z tych ośmiu adresów są zarezerwowane, pierwszy na adres sieci (jak w tym przypadku), ostatni na broadcast (adres rozgłoszeniowy). Adresy tej sieci są zatem następujące:

207.13.47.48/29 — adres sieci,

207.13.47.49/29,
207.13.47.50/29,
207.13.47.51/29,
207.13.47.52/29,
207.13.47.53/29,
207.13.47.54/29,
} adresy hostów,

207.13.47.55/29 — adres rozgłoszeniowy.

W tym przypadku nieprawidłowe jako adresy hostów będą te skrajne (adres sieci `207.13.47.48/29` i adres rozgłoszeniowy `207.13.47.55/29`).

Adresowanie IPv6

Protokół IPv6 (ang. *Internet Protocol version 6*) jest nową wersją protokołu IP, który ma zastąpić IPv4. Pierwsze dokumenty opisujące IPv6 pochodzą z 1995 roku. Protokół IPv6 wprowadzono ze względu na pewne ograniczenia protokołu IPv4 oraz na wyczerpującą się pulę wolnych adresów IP.

Poniżej podajemy główne cechy nowego protokołu:

- *Nowy format nagłówka.* Nagłówek IPv6 ma nowy format, który zaprojektowano w taki sposób, aby zminimalizować obciążenie związane z przetwarzaniem nagłówka (na routerach pośrednich). Nagłówki IPv4 i IPv6 nie współdziałają ze sobą i protokół IPv6

nie jest zgodny z protokołem IPv4. Aby host lub router rozpoznawał i przetwarzał oba formaty nagłówków, musi korzystać z implementacji zarówno protokołu IPv4, jak i IPv6.

- *Olbrzymia przestrzeń adresowa.* Źródłowe i docelowe adresy IPv6 mają 128 bitów (dla porównania: adresy IPv4 składają się tylko z 32 bitów).
- *Ułatwiona konfiguracja adresów.* Dla uproszczenia konfiguracji hostów protokół IPv6 obsługuje konfigurację adresów zarówno przy obecności serwera DHCP, jak i bez serwera DHCP. W tym drugim przypadku hosty podłączone do łącza automatycznie konfigurują swoje adresy IPv6 dla tego łącza.

Budowa IPv6

Dla IPv6 128-bitowy adres dzieli się na 16-bitowe człony. Każdy 16-bitowy blok konwertowany jest na postać szesnastkową.

Oto 128-bitowy adres IPv6 (podzielony na 16-bitowe fragmenty, człony adresu):

```
1101100101011100 0000110011001100 1000000100001010 1111110001000010
1111010110100001 0111001000100001 0001000010011010 0000000000010011
```

Każdy 16-bitowy człon adresu jest, dla skrócenia zapisu, zapisywany w postaci szesnastkowej i oddzielony dwukropkiem. Oto rezultat:

```
D95C:0CCC:810A:FC42:F5A1:7221:109A:0013
```

Reprezentacja IPv6 może zostać uproszczona przez usunięcie poprzedzających zer z każdego bloku 16-bitowego, przy czym każdy blok musi mieć przynajmniej jeden znak. Po usunięciu poprzedzających zer reprezentacja adresu wygląda następująco:

```
D95C:CCC:810A:FC42:F5A1:7221:109A:13
```

Niektóre typy adresów zawierają dłuższe sekwencje zer. Aby jeszcze bardziej uprościć adres IPv6, sąsiadujące sekwencje 16-bitowych bloków złożonych z zer w formacie szesnastkowym można zapisać jako `::`. Przykładowo adres typu link-local `FE80:0:0:0:2AC:FF:FE9A:4CA2` może zostać skrócony do postaci `FE80::2AC:FF:FE9A:4CA2`. Kompresji zer można użyć tylko raz w danym adresie. Prefiks w IPv6 pełni podobną funkcję jak maska w IPv4. Prefiks to liczba w systemie dziesiętnym, która informuje o liczbie bitów przeznaczonych na „adres sieci", na przykład `26DA:D4:0:1F3B::/64`.

Dozwolone jest zastąpienie jednego bloku zer podwójnym dwukropkiem, początkowe zera w grupach również mogą być opuszczane, w związku z czym poniższe sposoby zapisu są prawidłowe i równoznaczne:

```
2001:0CA7:0000:0000:0000::2528:57F8
2001:0CA7:0:0:0:0:2528:57F8
2001:0CA7:0:0::2528:57F8
2001:0CA7::2528:57F8
2001:CA7::2528:57F8
```

Sekwencja ostatnich 4 bajtów adresu może być również zapisana w postaci adresu IPv4, z wykorzystaniem kropek jako separatorów — adres `::ffff:80.55.69.250` jest równoznaczny adresowi `::ffff:5037:45FA`.

Gdy jest to wymagane, do adresu może być dołączona maska sieci w notacji CIDR, na przykład `2004:0da6:1212::/48`. Jeżeli natomiast zachodzi potrzeba podania portu docelowego (przykładowo w adresie URL), adres IPv6 otaczany jest nawiasami kwadratowymi, na przykład:

```
http://[2004:0da6:1212:08d5:1321:8c12:0381:6322]:443/
```

3.4.5. Podstawowe rodziny protokołów

Protokół nierutowalny (na przykład *NetBEUI* opracowany w 1985 roku przez firmę IBM) wykorzystuje jedynie adresowanie zapewniane przez warstwy 1. i 2. (na przykład adresy fizyczne MAC dla sieci Ethernet), przez co nie może służyć do komunikacji pomiędzy sieciami o różnej architekturze fizycznej (inny sposób adresowania, inna budowa ramek itd.). Protokoły tego typu sprawdzają się jedynie w niewielkich, odizolowanych sieciach LAN typu *peer-to-peer*. Protokoły nierutowalne nie są „przepuszczane" przez routery.

Protokoły rutowalne (na przykład *TCP/IP*, najczęściej stosowany zestaw protokołów sieciowych, który łączy komputery pracujące na różnych platformach sprzętowo-systemowych, lub *IPX/SPX*, opracowany w latach 70. przez firmę Novell) pozwalają na przesyłanie danych pomiędzy sieciami (router — urządzenie łączące sieci — odczytuje informacje i podejmuje decyzje, dokąd i jaką drogą przesłać pakiet informacji).

Protokoły internetowe

Każdy komputer podłączony do sieci i korzystający z zasobów sieciowych umożliwia uruchomienie wielu usług, które korzystają z sieci. Często słuchasz muzyki z radia internetowego, wysyłasz pocztę, przeglądasz strony WWW, rozmawiasz ze znajomymi i wysyłasz pliki na serwer FTP. Wszystkie te czynności (i wiele innych) możesz wykonywać jednocześnie. Jak to możliwe? Jednoczesne działanie tylu różnych usług umożliwiają różne protokoły sieciowe. Protokół HTTP pozwala przeglądać strony internetowe, protokół FTP umożliwia przesyłanie plików na serwer FTP, protokoły POP3 i SMTP są związane z pocztą internetową. Jest bardzo duża różnorodność usług sieciowych i związanych z nimi protokołów. Każdy protokół jest ściśle powiązany z warstwą modelu OSI. Poniżej zamieszczono wybrane protokoły sieciowe z podziałem na poszczególne warstwy. Protokoły umieszczono w kolejności alfabetycznej. Cyfry w nawiasie oznaczają domyślne numery portu.

W komputerze, jak w dużym budynku firmy czy hotelu, jest wiele drzwi (portów) wejściowych. Innymi drzwiami wchodzą klienci, innymi obsługa, innymi kierownictwo. Podobnie jest w komputerze. Każda usługa sieciowa wykorzystuje inny **port** (inne drzwi). Dzięki wykorzystaniu różnych portów możliwe jest jednoczesne działanie wielu usług opartych na różnych protokołach.

Warstwa łącza danych

ARP, RARP, DCAP.

Warstwa sieciowa

DHCP (67, 68), DVMRP, ICMP/ICMPv6, IGMP, IP, IPv6, MARS, PIM, RIP2, RIPng dla IPv6, RSVP, VRRP.

Warstwa transportowa

ISTP, Mobile IP, RUDP, Tali, TCP, UDP, XOT.

Warstwa sesji

BGMP, DIS, DNS (53), ISAKMP/IKE, iSCSI, LDAP, MZAP, NetBIOS (137, 138, 139).

Warstwa aplikacji

COPS, FANP, Finger, FTP (20, 21), HTTP (80), IMAP (143), IMPPpre/IMPPmes, IPDC, IRC (194, 529), ISAKMP, ISP, NTP (123), POP3 (110), Radius, RLOGIN, RTSP, SCTP, S-HTTP, SLP, SMTP (25), SNMP (161, 162), SOCKS, SSH (22), TACACS+, TELNET (23), TFTP, WCCP, X-Window.

Szerzej omówimy tylko wybrane protokoły.

Protokół HTTP

HTTP (ang. *Hypertext Transfer Protocol* — protokół przesyłania dokumentów hipertekstowych) to protokół sieci WWW (ang. *World Wide Web* — ogólnoświatowa sieć dokumentów hipertekstowych). Za jego pomocą przesyła się żądania udostępnienia dokumentów WWW. Zadaniem stron WWW jest publikowanie informacji, natomiast protokół HTTP umożliwia ich udostępnianie.

Protokół HTTP jest bardzo użyteczny, ponieważ pozwala na komunikowanie się komputerów w znormalizowany sposób. Określa on formę żądań **klienta** dotyczących danych oraz formę odpowiedzi **serwera** na te żądania. Jest zaliczany do protokołów bezstanowych (ang. *stateless*), ponieważ nie zachowuje żadnych informacji o poprzednich transakcjach z klientem (po zakończeniu transakcji wszystko przepada). Pozwala to znacznie zmniejszyć obciążenie serwera, jednak jest kłopotliwe w sytuacji, gdy trzeba zapamiętać konkretny stan dla użytkownika, który wcześniej łączył się już z serwerem. Najczęstszym rozwiązaniem tego problemu jest wprowadzenie mechanizmu *cookies* (plików tekstowych wysyłanych przez serwer HTTP, zapisywanych na dysku klienta).

HTTP standardowo korzysta z portu nr 80.

Protokół HTTPS

HTTPS (ang. *HyperText Transfer Protocol Secure* — protokół przesyłania zaszyfrowanych dokumentów hipertekstowych) różni się od zwykłego protokołu HTTP po prostu tym, że tekst w komunikacji jest szyfrowany za pomocą protokołu SSL (ang. *Secure Socket Layer* — protokół bezpiecznych połączeń). Zamiast standardowo pisać w przeglądarce *http://*,

dodajemy na końcu *s* (https://), informując serwer, że chcemy nawiązać bezpieczne połączenie. SSL daje poczucie bezpieczeństwa, które dzisiaj jest tak ważne.

HTTPS korzysta z portu 443 (na którym nasłuchuje SSL).

Ciekawostka

SSL jest protokołem otwartym. Oznacza to, że firma Netscape, która jest właścicielem tego protokołu, zezwoliła na korzystanie z niego w aplikacjach internetowych. SSL jest w rzeczywistości kolejną warstwą, która znajduje się ponad warstwą aplikacji, zawierającą między innymi protokół HTTP. Zanim zostanie zastosowane szyfrowanie za pomocą protokołu SSL, przeglądarka klienta oraz serwer muszą uzgodnić połączenie. Służy do tego proces uzgadniania połączenia.

Protokół FTP i SFTP

FTP (ang. *File Transfer Protocol* — protokół transferu plików) to protokół typu **klient-serwer**, który umożliwia przesyłanie plików z serwera i na serwer poprzez sieć TCP/IP.

Do komunikacji wykorzystywane są dwa połączenia TCP. Jedno z nich jest połączeniem kontrolnym, za pomocą którego przesyłane są polecenia do serwera, drugie służy do transmisji danych. FTP domyślnie wykorzystuje do transmisji danych port 20., natomiast do kontroli transmisji — port 21. Podstawową zaletą FTP jest możliwość kontynuowania przerwanego transferu danych aż do uzupełnienia brakujących części pliku oraz duża prędkość transmisji. Główną wadą protokołu jest jednak brak możliwości szyfrowania transmisji, mimo to nadal jest on popularnym protokołem przesyłania danych.

Lepszym rozwiązaniem jest protokół SFTP (ang. *Secure File Transfer Protocol* — protokół bezpiecznego przesyłania plików). Przesyła on pliki z danymi z serwera i na serwer, szyfrując nie tylko dane, ale i hasło dostępu do serwera. Protokół ten zapewnia szyfrowanie połączenia, przesyłu danych oraz rozpoznawanie użytkownika.

Protokół SMTP

SMTP (ang. *Simple Mail Transfer Protocol*) to protokół komunikacyjny opisujący sposób wysyłania poczty elektronicznej w internecie. SMTP działa najczęściej na porcie 25., jednak ze względu na bezpieczeństwo transmisji usługodawcy zalecają zmianę tego portu na wyższy, wskazując jego konkretną wartość dla konfiguracji klienta pocztowego (na przykład 587.).

SMTP nie pozwala na pobieranie wiadomości ze zdalnego serwera. Do tego celu służą POP3 lub IMAP.

POP3 i IMAP

POP3 (ang. *Post Office Protocol version 3*) to protokół internetowy z warstwy aplikacji pozwalający na odbiór poczty elektronicznej ze zdalnego serwera do lokalnego komputera poprzez połączenie TCP/IP.

Protokół POP3 jest przeznaczony do pracy offline. Po połączeniu z serwerem cała nieprzeczytana korespondencja wraz z załącznikami przesyłana jest ze skrzynki pocztowej na serwerze do komputera lokalnego. Odebrane listy zostają oznaczone jako odczytane i pozostają na serwerze lub (domyślnie) są z niego kasowane. Czytanie listów i przygotowywanie odpowiedzi może odbywać się bez konieczności utrzymywania połączenia z internetem. Ponowne połączenie wymagane jest tylko w celu wysłania wiadomości lub sprawdzenia obecności nowych listów.

Istnieje nowszy i bardziej zaawansowany protokół — **IMAP** (ang. *Internet Message Access Protocol*). Wiadomości przechowywane są na serwerze w skrzynce pocztowej i w dodatkowych folderach pocztowych. Po połączeniu z serwerem na komputer lokalny przesyłane są jedynie nagłówki wiadomości. Transmisja treści oraz załączników następuje dopiero po otwarciu danego listu. Możliwe jest kasowanie i filtrowanie korespondencji bez jej pobierania z serwera. Ten sposób obsługi poczty doskonale sprawdza się w przypadku pracy na wielu komputerach, na każdym z nich widać bowiem tę samą zawartość skrzynki.

DNS

DNS (ang. *Domain Name Service*) to protokół obsługujący system nazywania domen. Umożliwia on nadawanie nazw domenowych — które są zrozumiałe i łatwe do zapamiętania dla człowieka — tłumacząc je na adresy IP. Dzięki temu możliwe jest wpisanie w pasku adresu przeglądarki internetowej adresu **URL**, na przykład *www.wp.pl*, zamiast adresu IP `212.77.100.101`. Adres ten zostaje odszukany w domyślnym serwerze DNS i zamieniony na adres IP. Następnie jest lokalizowany serwer o tym adresie, który odpowiada, przesyłając zawartość strony do naszej przeglądarki internetowej.

DNS oferuje ponadto hierarchiczną usługę rozróżniania nazw hostów, które zorganizowane są w hierarchiczną formę drzewa. Początek drzewa nazywany jest korzeniem. Nazwy najwyższego poziomu składają się z dwuliterowych domen narodowych opartych na zaleceniach ISO 3166. Nadrzędna domena narodowa w Polsce oznaczona jest skrótem *pl*. Jeżeli chodzi o domeny trzyliterowe, ich znaczenie jest następujące:

com — organizacje komercyjne,

gov — agencje rządowe,

edu — instytucje edukacyjne,

mil — organizacje wojskowe,

org — pozostałe organizacje,

net — organizacje, których działalność dotyczy sieci komputerowych.

DHCP

DHCP (ang. *Dynamic Host Configuration Protocol* — protokół dynamicznego konfigurowania węzłów) to standardowy protokół przydzielający adresy IP poszczególnym kom-

puterom. Przydział adresów IP jest realizowany poprzez serwer DHCP, który obsługuje daną podsieć. Hosty, zgłaszając się do sieci, otrzymują kolejne adresy IP z puli przydzielonej serwerowi DHCP. Oprócz automatycznej konfiguracji adresów IP następuje również przydzielenie maski, adresu bramy domyślnej (umożliwiającej dostęp do internetu) oraz adresu domyślnego serwera DNS (obsługa adresów URL).

Telnet i SSH

Telnet jest usługą (programem) pozwalającą na zdalne połączenie się komputera (terminala) z oddalonym od niego komputerem (serwerem) przy użyciu sieci. Wykorzystuje do tego celu protokół TCP-IP oraz standardowo przypisany port 23. Telnet umożliwia zatem użytkownikowi ustanowienie zdalnej sesji na serwerze.

Wszystkie polecenia muszą być wprowadzane w trybie znakowym w wierszu poleceń. Polecenia wydawane za pomocą naszego komputera zostają przesłane poprzez sieć do serwera, na którym zainstalowane jest oprogramowanie serwera telnetu. Do korzystania z tej usługi niezbędne jest posiadanie na serwerze konta typu *shell*.

Usługa ta uruchamiana jest po wpisaniu polecenia: `telnet adres part`, gdzie `adres` jest adresem IP komputera, z którym chcemy się połączyć, lub jego nazwą domenową, gdyż telnet dopuszcza obie te formy podawania adresu. Po nawiązaniu połączenia w celu zalogowania się do serwera należy podać nazwę użytkownika oraz hasło (*login* i *password*).

SSH (ang. *secure shell* — bezpieczna powłoka) jest następcą protokołu telnet. SSH, podobnie jak telnet, służy do łączenia się ze zdalnym komputerem (terminalem). Jednakże SSH zapewnia szyfrowanie oraz umożliwia rozpoznawanie użytkownika. W szerszym znaczeniu SSH jest wspólną nazwą całej rodziny protokołów. Obejmuje ona nie tylko podstawowe protokoły służące do zdalnego połączenia, ale również protokoły do przesyłania plików, zdalnej kontroli zasobów komputera i wiele innych zastosowań. Wspólną cechą tych wszystkich protokołów jest technika szyfrowania połączenia, przesyłu danych oraz metoda rozpoznawania użytkownika.

Funkcja *Pulpit zdalny* (ang. *Remote desktop*) daje możliwość zdalnego sterowania komputerem — tym samym umożliwia korzystanie w trybie graficznym z danych, aplikacji oraz zasobów sieciowych dostępnych na odległym komputerze.

Protokół IP

Najważniejszym protokołem komunikacyjnym warstwy sieciowej jest protokół IP.

Oto jego cechy:

- jest to protokół bezpołączeniowy (nie wysyła i nie przyjmuje żadnych informacji kontrolnych, które przed wysłaniem danych ustanawiałyby połączenie między odbiorcą a nadawcą);
- nie ma systemu korekcji błędów;
- nie sprawdza, czy dane zostały prawidłowo odebrane;
- zajmuje się adresowaniem datagramu.

3.4.6. Obsługa i konfiguracja sieci w Windows

Jak skonfigurować komputer pracujący pod kontrolą systemu operacyjnego Windows, aby uzyskać dostęp do internetu?

Dokładne instrukcje zamieszczone zostały na płycie CD, w pliku *obsluga i konfiguracja sieci w Windows.pdf*.

3.4.7. Ogólne zasady administrowania siecią komputerową w architekturze klient-serwer

Architektura klient-serwer (ang. *client-server*) umożliwia podział zadań (ról) pomiędzy serwerem a klientem. Klient może komunikować się z serwerem, wysyłając żądanie obsługi (ang. *request*). Serwer cały czas nasłuchuje, czekając na klientów, i w momencie otrzymania żądania przetwarza je i wysyła odpowiedź. Serwer odpowiada więc na żądanie i zapewnia usługi dla klientów. Z usług jednego serwera może korzystać wielu klientów. Zdarza się, że jeden klient może korzystać jednocześnie z usług wielu serwerów.

Połączenie pomiędzy klientem a serwerem opisane jest przy użyciu protokołów sieciowych. W modelu OSI komunikacja pomiędzy stronami przebiega w warstwie aplikacji.

Serwery działające na podstawie architektury klient-serwer to: serwer poczty elektronicznej, serwer WWW, serwer plików, serwer aplikacji oraz większość nowoczesnych systemów zarządzania bazą danych.

Prawdopodobnie w szkolnej pracowni komputerowej znajduje się serwer, na którym uruchomiono wiele usług typu klient-serwer. Komputer ten jest nie tylko serwerem plików, ale i serwerem aplikacji, serwerem pocztowym, serwerem WWW, serwerem bazodanowym, serwerem DHCP. W szkolnych pracowniach komputerowych najczęściej są używane dwa typy oprogramowania sieciowego: Novell NetWare lub Windows Server 2000 (lub 2003).

Zarządzaniem i administracją sieci zajmuje się administrator. Jest to specjalny użytkownik, który posiada pełne prawa do wszystkich zasobów sieciowych. Większość swojego czasu administrator systemu poświęca trzem głównym zadaniom, są to: instalacja, konfiguracja i utrzymanie bezawaryjnej pracy sieci.

Instalacja obejmuje instalację serwera oraz dostępnych aktualizacji. W przypadku oprogramowania serwera Novell NetWare niezbędna jest również instalacja programu klienta na stacjach roboczych (komputerach użytkowników).

Konfiguracja obejmuje między innymi tworzenie kont użytkowników oraz grup, do których przypisani są użytkownicy. Zarządzanie grupami umożliwia nadanie tych samych praw wielu użytkownikom, co znacznie ułatwia pracę administratora. Na przykład wszyscy użytkownicy w danej grupie otrzymują prawo dostępu do zasobów i aplikacji oraz drukowania na drukarce sieciowej. Jest to zatem zarządzanie środowiskiem użytkownika za pomocą zasad grupy.

Utrzymanie bezawaryjnej pracy sieci to monitorowanie pracy serwera, ochrona systemu przed awariami, usuwanie awarii, zarządzanie aktualizacjami oprogramowania oraz systematyczne tworzenie kopii bezpieczeństwa. Wszystkie informacje przechowywane są na serwerze, wobec tego możliwe jest lepsze zabezpieczenie danych, jednak w czasie, gdy serwer nie działa, dostęp do danych jest niemożliwy.

3.4.8. Bezpieczeństwo sieci

Żyjemy w świecie, który nieustannie się rozwija. Rozwój ten nie omija sieci komputerowych. Przesyłanie różnorodnych danych poprzez sieć jest bardzo wygodne, ale i bardzo niebezpieczne. Dane narażone są na kradzież. Sieciowi włamywacze oraz różnego rodzaju wandale prześcigają się w łamaniu coraz to nowszych zabezpieczeń. Nawet najbardziej doświadczeni administratorzy sieci oraz osoby odpowiedzialne za bezpieczeństwo danych cyfrowych muszą nieustannie poszerzać swoje umiejętności. Dziś podanie **loginu** oraz **hasła** nie jest wystarczającym zabezpieczeniem danych. Istnieje bardzo wiele niebezpieczeństw czyhających na nasze dane znajdujące się w sieci komputerowej.

W jaki sposób dane mogą zostać ukradzione?

Najprostszy sposób to uzyskanie bezpośredniego dostępu do komputera, w którym znajdują się dane. Dostęp ten można uzyskać po dokonaniu lokalnego lub zdalnego włamania się do tego komputera. **Hakerzy**, jak nazywa się potocznie ludzi przeprowadzających elektroniczne włamania, używają wielu metod. Najczęściej wykorzystywane są luki, które znajdują się w protokole sieciowym TCP/IP oraz protokołach mu pokrewnych. Sam protokół TCP/IP nie ma żadnych mechanizmów szyfrowania, co oznacza, że dane wysyłane są w sposób jawny.

Często wykorzystywane są błędy w zabezpieczeniach, które popełnił administrator sieci.

Innym sposobem włamania jest zdobycie przez hakera danych umożliwiających zalogowanie się na zdalnej maszynie. Najbardziej niebezpieczne jest włamanie na konto administratora komputera lub sieci. Usuwanie skutków takiego włamania jest bardzo trudne i nigdy nie daje stuprocentowej pewności, czy haker nie będzie miał możliwości kolejnego przejęcia kontroli nad naszym komputerem.

Mając na uwadze bezpieczeństwo danych, należy przestrzegać pewnych standardów bezpieczeństwa. Ich zbiór został opublikowany w 1983 roku w dokumencie zwanym *Orange Book* (a dokładnie w standardzie *Trusted Computer System Evaluation Criteria*) przez Departament Obrony USA. Zdefiniowano tam siedem poziomów bezpieczeństwa komputerowego systemu operacyjnego: D, C1, C2, B1, B2, B3 oraz A1 (zostały opisane w umieszczonym na płycie CD pliku *orange book.pdf*).

Hasła

Zastosowanie hasła jest chyba najbardziej rozpowszechnioną metodą uwierzytelniania użytkownika. Oczywiście metoda ta nie daje stuprocentowej pewności, że osoba

logująca się jako użytkownik jest tą osobą, która ma prawo się logować. W zasadzie, gdy mamy do czynienia z tego typu logowaniem w środowisku sieciowym, złamanie hasła jest stosunkowo proste.

Metoda nasłuchiwania sieci może potencjalnemu hakerowi dostarczyć wszystkich informacji, by mógł odczytać kombinację użytych znaków, które stanowią hasło. Programy takie jak IPtrace, Snoop czy też LanWatch są często stosowanymi narzędziami do nasłuchu i identyfikacji hasła. Szczególnie niebezpieczne jest połączenie ze zdalną maszyną za pomocą protokołu *telnet*. Dane, takie jak hasło i login, przesyłane są otwartym tekstem (bez szyfrowania), co powoduje możliwość przejęcia danych i uprawnień.

Bardzo niebezpieczne są wszelkiego typu programy zainstalowane nieświadomie przez użytkownika na jego komputerze, takie jak **konie trojańskie**, które często mają na celu pozyskanie danych logowania.

Typowym sposobem łamania haseł jest tzw. **atak poprzez słownik** (ang. *dictionary attack*). Polega on na próbkowaniu programu autoryzującego całym słownikiem danych. Jeżeli weźmiemy pod uwagę, że w przypadku klasycznego systemu Unix maksymalna długość hasła wynosi 8 znaków, a funkcja szyfrująca używa tylko siedmiu bitów znaczących, otrzymujemy klucz o długości 56 bitów. Ponieważ większość użytkowników stosuje proste hasła, pole poszukiwań może być zawężone tylko do na przykład małych liter — odpowiada to sytuacji używania klucza o długości 19 bitów. Przykładem programu dokonującego ataku poprzez słownik jest COPS, który używa tej metody do diagnozowania skuteczności zabezpieczeń hasłem. Dlatego należy używać hasła jak najdłuższego i dopilnować, aby zawierało ono zarówno małe, jak i wielkie litery oraz liczby. Hasło nie jest na całe życie, należy je często zmieniać.

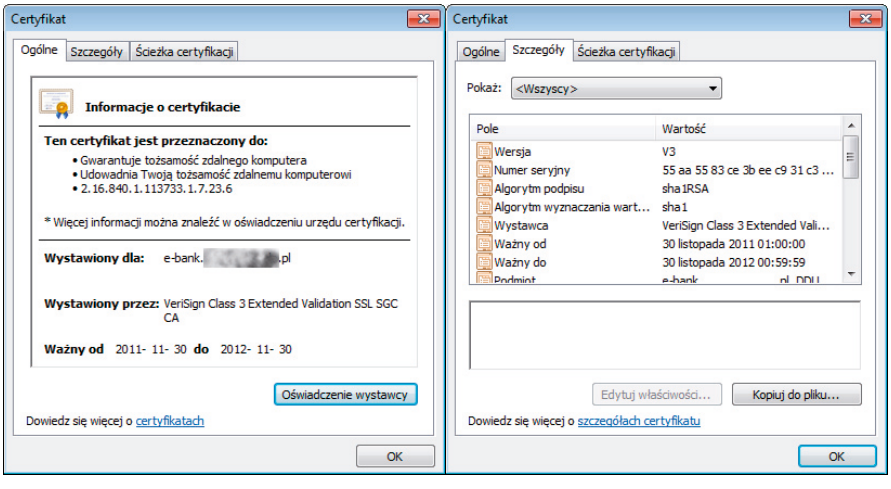
Jednak najbardziej wrażliwym elementem systemu bezpieczeństwa opartego na hasłach jest sam użytkownik. Często bowiem hasła pożyczane są innym osobom, a te najczęściej, nie chcąc ich zapomnieć, zapisują je na żółtych karteczkach i przyklejają na monitorach lub pod klawiaturą. W takim przypadku cała polityka haseł oraz zabezpieczeń nie ma żadnego sensu. Haker musi jedynie umieć czytać, aby uzyskać dostęp do danych.

Jak należy postępować, aby zabezpieczyć się przed takimi zagrożeniami?

Jeśli chodzi o hasła, w pierwszej kolejności należy wprowadzić jasną politykę bezpieczeństwa obowiązującą wszystkich użytkowników, którzy mają prawo logować się do komputerów i sieci. Administrator sieci powinien tak skonfigurować konta użytkowników, aby system sam przypominał o wygaśnięciu ważności hasła, narzucał jego minimalną długość i wymagał użycia małych i wielkich liter oraz cyfr. W przypadku udostępnienia zdalnego logowania do sieci należy tak ją skonfigurować, aby każda nowa sesja logowania wymagała podania nowego hasła. Można również zwiększyć poziom uwierzytelnienia użytkownika przez wprowadzenie dodatkowych zabezpieczeń w postaci kart logowania czy też technologii sprawdzania linii papilarnych. Oczywiście te technologie nie wyczerpują gamy możliwości zwiększenia poziomu bezpieczeństwa podczas logowania. Innym sposobem jest tworzenie zaszyfrowanego połączenia, dzięki któremu cała komunikacja między komputerami jest szyfrowana w obie strony.

Szyfrowanie a prywatność

Szyfrowanie jest skomplikowaną technologią, jednak w praktyce często z tej technologii korzystamy, nawet nie wiedząc o tym. Przykładowo przy próbie zalogowania się na konto internetowe banku użytkownik po wprowadzeniu adresu *www.*bank*.pl* zostanie przekierowany na stronę *https://www.*bank*.pl/*. Użycie protokołu HTTP wraz z protokołem szyfrowania SSL (o czym świadczy początek adresu *https://*) gwarantuje szyfrowanie informacji między komputerami połączonymi w sieci. Choć wszystko na razie wygląda prawidłowo, należy się jeszcze upewnić, że znajdujemy się tam, gdzie trzeba. Aby mieć pewność, że nie trafiliśmy na stronę, która tylko udaje bank internetowy, powinniśmy bezwzględnie sprawdzić informacje o certyfikacie cyfrowym, a właściwie ważność certyfikatu, oraz to, na kogo certyfikat jest wystawiony. W tym celu na pasku adresowym klikamy symbol kłódki, która powinna być zamknięta, a następnie wybieramy *Wyświetl certyfikat*, co spowoduje wyświetlenie okna z danymi dotyczącymi certyfikatu, tak jak to pokazano na rysunku 3.62.



Rysunek 3.62. Informacja o szyfrowaniu połączenia i szczegółach certyfikatu wraz z pełnymi danymi o certyfikacie

Z informacji widniejących na zakładce *Ogólne* można wywnioskować, że faktycznie strona, na której się znajdujemy, jest miejscem, gdzie możemy rozpocząć proces logowania do bankowego konta internetowego.

Jak działa protokół SSL?

SSL (ang. *Secure Socket Layer* — protokół bezpiecznych połączeń), protokół bezpiecznej komunikacji między klientem a serwerem, współpracuje z istniejącymi protokołami, takimi jak: HTTP, FTP, SMTP, NNTP i telnet. Powszechnie jest używany głównie HTTPS (HTTP na SSL). SSL zapewnia:

- prywatność — połączenie jest szyfrowane;
- autoryzację — klient i serwer określają swoją tożsamość;
- integralność przesyłanych danych — przez sumy kontrolne.

Ze względu na sposób dokonywania autoryzacji SSL jest protokołem scentralizowanym, inaczej niż na przykład PGP. Opiera się na grupie instytucji certyfikujących CA (ang. *Certyfing Authorities* — centrum certyfikacji), które opatrują swoim podpisem certyfikaty poszczególnych serwerów. CA z założenia są godne zaufania, dzięki czemu wchodząc na serwer legitymujący się certyfikatem jednego ze znanych CA, mamy pewność, że serwer rzeczywiście jest tym, za który się podaje.

W skrócie jego działanie można przedstawić następująco:

- Komputer chcący się połączyć z serwerem wysyła wiadomość zawierającą informacje o metodzie szyfrowania, metodzie kompresji danych i wersji protokołu. Wysyłany jest również klucz publiczny z komputera na serwer.
- Serwer wysyła klientowi swój klucz publiczny oraz informacje potwierdzające, że parametry połączenia są już uzgodnione na warunkach klienta.
- Serwer wysyła do klienta swój certyfikat, który należy zweryfikować. Od tego momentu wszystkie dane przesyłane pomiędzy tymi hostami uważane są za autentyczne, co więcej, dodatkowe klucze służą do zaszyfrowania wymienianych informacji między klientem a serwerem.

Sesja SSL zostanie zakończona w razie wylogowania lub gdy klient odłączy się od serwera — na przykład zamknie okno przeglądarki lub wpisze inny adres URL. Obecnie większość banków działających w Polsce, które oferują usługi bankowości internetowej, używa protokołu *SSL 3.0 (V3)*, a klucz do szyfrowania informacji jest *2048-bitowy*, co ilustruje rysunek 3.62.

Bezpieczeństwo dostępu do sieci bezprzewodowej

Kolejnym przykładem zastosowania szyfrowania jest komunikacja z punktem dostępowym bezprzewodowej sieci internetowej (ang. *hotspot* — gorący punkt). Aby połączenie było bezpieczne, należy je tak skonfigurować, by zapewniało autoryzowany dostęp do sieci bezprzewodowej oraz bezpieczeństwo przesyłania danych. W tym przypadku najważniejszą rolę odgrywa prawidłowe skonfigurowanie punktu dostępowego **Wi-Fi**. W pierwszej kolejności należy zalogować się do routera przy użyciu połączenia kablowego i zmienić hasło administratora na inne niż fabryczne. Przed uaktywnieniem opcji Wi-Fi należy ustanowić sposób szyfrowania (najlepiej na *WPA2*). Ten standard musi być obsługiwany przez wszystkie urządzenia działające w sieci bezprzewodowej (Wi-Fi). Jeśli jest taka możliwość, warto włączyć opcję odpowiedzialną za *firewall* (*zaporę sieciową*). Można również zablokować obwieszczanie identyfikatora *SSID*, co spowoduje, że nasza sieć nie będzie widziana przez innych. Zmniejszy to liczbę prób włamań, co oczywiście podniesie poziom bezpieczeństwa.

Bardzo ważną sprawą jest regularna aktualizacja oprogramowania routera. Nawet najdroższy router nie zapewni bezpieczeństwa, jeśli w jego oprogramowaniu znajduje się luka. Aktualizacja oprogramowania usuwa wykryte błędy oprogramowania, dlatego powinniśmy regularnie odwiedzać stronę internetową producenta.

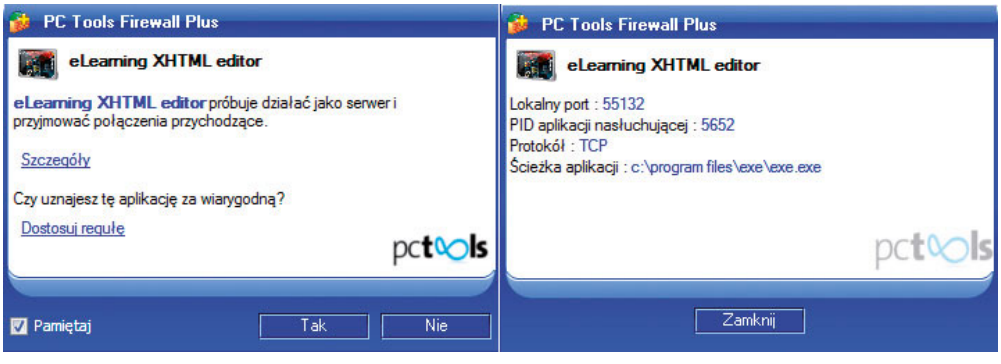
Program antywirusowy i zapora sieciowa

Każdy z komputerów podłączonych do sieci powinien również być zabezpieczony przez swoje oprogramowanie zapory sieciowej oraz program antywirusowy, który powinien być jak najczęściej aktualizowany i cały czas aktywny.

Zapora sieciowa (ang. *firewall*) składa się z pewnej liczby komponentów sieciowych (sprzętowych i programowych) w punkcie styku dwóch sieci. Głównym jej zadaniem jest zachowanie reguł bezpieczeństwa między siecią prywatną a niezabezpieczoną siecią publiczną, na przykład internetem.

Właśnie ta zapora decyduje, które z usług w sieci prywatnej dostępne są z zewnątrz i z jakich usług niezabezpieczonej sieci publicznej można korzystać z poziomu sieci prywatnej. Aby *firewall* był skuteczny, cały ruch danych między siecią prywatną a internetem musi przez niego przechodzić.

Podczas uruchamiania niektórych aplikacji może pojawić się komunikat pochodzący od zapory sieciowej (na przykład *PC Tools Firewall Plus*). Komunikaty te mogą pochodzić zarówno od programów, które mają być uruchomione i nie stanowią zagrożenia, jak i od koni trojańskich, które nie powinny być uruchomione na naszych komputerach. To użytkownik decyduje, czy aplikacja jest wiarygodna i będzie uruchomiona, czy też nie. Trzeba też zdecydować, które usługi (aplikacje) powinny mieć dostęp do internetu, a które nie, tak jak to pokazano na rysunku 3.63.



Rysunek 3.63. Próba uzyskania dostępu do internetu przez edytor zasobów zdalnego nauczania

Aplikacje, które nie powinny mieć dostępu do internetu, zostaną zablokowane. Jeśli polecenie *Tak* lub *Nie* ma być zapamiętane na stałe, należy zaznaczyć opcję *Pamiętaj*. Również dostęp z zewnątrz do naszego komputera jest znacznie utrudniony przez działanie zapory sieciowej. Zapora sieciowa blokuje lub udostępnia porty przypisane do usługi sieciowej.

Złośliwe oprogramowanie malware (z ang. *malicious software*)

Kolejnymi zagrożeniami, które możemy napotkać podczas podłączenia sieci do internetu, są te związane z różnymi atakami na sieć. Do złośliwego oprogramowania zaliczyć należy wszelkie aplikacje, skrypty i ingerencje mające szkodliwe, przestępcze lub złośliwe

działanie w stosunku do użytkownika komputera. Rozróżniamy następujące, omówione poniżej typy ataków i zagrożeń.

- **Wirus.** To program lub fragment wrogiego wykonalnego kodu, który bez zgody użytkownika dołącza się, nadpisuje lub zamienia zainfekowany program w celu reprodukcji samego siebie. Ze względu na różne rodzaje infekcji wirusy dzielą się na: wirusy gnieżdżące się w boot sektorze dysku twardego, wirusy pasożytnicze, wirusy wielocześciowe, wirusy towarzyszące oraz makrowirusy.
- **Robak.** Ten wirus rozmnaża się tylko w sieci. Nie potrzebuje programu **żywiciela** tak jak typowe wirusy. Robaki często powielają się poprzez pocztę elektroniczną.
- **Trojan.** „Koń trojański" to określenie oprogramowania, które podszywając się pod przydatne lub ciekawe dla użytkownika aplikacje, dodatkowo implementuje niepożądaną, ukrytą przed użytkownikiem działalność.
- **Inżynieria społeczna** (ang. *social engineering*). Częstym błędem popełnianym przez wielu internautów jest otwieranie poczty elektronicznej od osób lub instytucji, z którymi zwykle nie prowadzą żadnej korespondencji. Nigdy nie powinniśmy korzystać z linków, które znajdują się wewnątrz listu, w celu połączenia się ze stroną na przykład banku. Żaden bank nigdy do nas takiego maila nie skieruje i nigdy nie poprosi o uzupełnienie w ten sposób jakichkolwiek danych. Nieostrożne kliknięcie linku przekieruje nas na stronę hakera. Próby wyłudzenia ważnych informacji są realizowane nie tylko poprzez pocztę elektroniczną — w praktyce inżynieria społeczna sprowadza się do świadomego wprowadzenia w błąd pracownika konkretnej organizacji lub firmy w celu zdobycia poufnych danych, najczęściej haseł dostępu.
- **Skanery i exploity.** Kolejnym problemem mogą być programy typu skaner i exploit. **Exploit** to program zmierzający do przejęcia kontroli nad atakowanym systemem. W programach takich definiowane są rodzaje rozpoznawanych luk. Exploity mogą być wykorzystywane przez osoby niemające elementarnej wiedzy z zakresu technik hakerskich, co powoduje, że tego typu ataki są niezwykle popularne. **Skanery** to programy służące do testowania i wykrywania luk w serwerze. Są to narzędzia przeznaczone dla administratora sieci, który powinien dbać o bezpieczeństwo serwera, jednak często są używane przez osoby chcące włamać się do serwera poprzez niezabezpieczone luki. Skanery to jedno z najpopularniejszych narzędzi do naruszania bezpieczeństwa systemów przez użytkowników mających dostęp do internetu. Każdy złożony atak jest poprzedzony dokładnym zebraniem informacji o celu ataku.
- **Sniffing.** To kolejne zagrożenie dla naszej sieci. Polega na dokładnym sprawdzeniu, jakiego typu dane są przesyłane w sieci ofiary. „Podsłuchiwanie ruchu" (ang. *sniffing*) jest metodą niezwykle skuteczną. Liczba informacji, które tą drogą można zdobyć, wystarcza do uzyskania pełnej kontroli nad komputerem ofiary. Lekarstwem na to, by wszelkie przechwytywane pakiety były nieużyteczne z punktu widzenia podsłuchującego, jest korzystanie z szyfrowanych protokołów. Mowa tu o takich protokołach jak *https* bądź szyfrowanych protokołach pocztowych — *pop* oraz *smtp*. Należy również wspomnieć, że szanse skutecznego podsłuchu mogą zostać ograniczone przez stosowanie

odpowiednich urządzeń sieciowych, takich jak switch. Zapewniają one, że każdy pakiet jest wysyłany wyłącznie z punktu A do punktu B, z pominięciem innych hostów. Co za tym idzie, wszelkie próby podsłuchu nie mają sensu.

- **Spoofing.** To szereg technik zmierzających do podszycia się pod kogoś innego w sieci. Idea jest prosta. Gdy następuje uwierzytelnienie klienta, intruz łączy się z sesją, udając uwierzytelnioną maszynę. Później aplikacje uznają, że komunikują się ciągle z tym samym klientem. Jeżeli potencjalny intruz zdoła wysłać pakiety w imieniu autoryzowanego wcześniej klienta, to może na przykład zerwać połączenie, zmodyfikować sesję (przykładowo zmieniając kwotę 100 zł na 100 000 zł) lub też doprowadzić do przejęcia całości połączenia. Każde z tych działań niesie ze sobą potencjalne niebezpieczeństwo. Jak się chronić przed takim atakiem? Ochrona polega głównie na skorzystaniu z połączenia szyfrowanego. Można również zastosować zaawansowany *firewall*, który potrafi skojarzyć adres IP hosta z wysyłanymi do niego pakietami.
- **Rootkity.** To programy, które po zainstalowaniu w systemie tak go modyfikują, aby możliwe było wykonanie zaprogramowanych czynności bez ich wykrycia. *Rootkity* pomagają ukryć się innym procesom, które wykonują niepożądane czynności w systemie. Na przykład, jeżeli w systemie znajdują się *backdoors* (tylne drzwi), które śledzą wykonywane czynności, to *rootkit* ukrywa otwarte porty, mogące być ostrzeżeniem przed niepożądaną komunikacją. A gdy program rozsyłający spam rozpoczyna wysyłanie wiadomości poczty elektronicznej, wówczas *rootkit* ukrywa wszelkie czynności związane z ich wysyłaniem.
- **Adware.** To rodzaj oprogramowania rozpowszechnianego bezpłatnie. W zamian za możliwość darmowego użytkowania oprogramowanie wyświetla reklamy, zazwyczaj w postaci graficznych banerów. Programy adware to również element tzw. złośliwego oprogramowania, instalującego w systemie operacyjnym dodatkowe, trudne do usunięcia moduły, których zadaniem jest wyświetlanie niechcianych przez odbiorcę reklam (niezależnie od reklam w darmowo użytkowanym programie). Złośliwe adware bardzo często występują wspólnie z programami szpiegowskimi *spyware*.
- **Spyware.** To programy komputerowe, których celem jest szpiegowanie działań użytkownika. Gromadzą one informacje o użytkowniku i wysyłają je, często bez jego wiedzy i zgody, autorowi programu. Do takich informacji mogą należeć: adresy WWW stron internetowych odwiedzanych przez użytkownika, dane osobowe, numery kart płatniczych, hasła, zainteresowania użytkownika (na przykład na podstawie słów wpisywanych w oknie wyszukiwarki), adresy e-mail, archiwum. Programy te czasami mogą wyświetlać reklamy lub rozsyłać spam.

Aby przedostać się do komputera, złośliwe oprogramowanie korzysta z różnych kanałów. Może być pobrane z internetu i zainstalowane przez *trojany*. Podczas próby uzyskania dostępu do witryny WWW może wyświetlić się prośba o wyrażenie zgody na zainstalowanie niezbędnego lub aktualnego oprogramowania pochodzącego z niepewnego źródła. Gdy wyrazimy zgodę, zostanie zainstalowane niebezpieczne oprogramowanie.

Wskazówka

Formanty *ActiveX* uatrakcyjniają przeglądane strony, dostarczając materiały wideo, animowaną zawartość itp. Programy te mogą jednak działać nieprawidłowo lub pobierać niepożądaną zawartość. W niektórych przypadkach służą do dyskretnego zbierania informacji o systemie, niszczenia znajdujących się w nim danych, instalowania oprogramowania bez wiedzy użytkownika lub zezwalania innej osobie na zdalne przejęcie kontroli nad komputerem. Uwzględniając te zagrożenia, należy instalować wyłącznie programy, których wydawcom można całkowicie zaufać. **Jak zainstalować formant** *ActiveX?* Przy pierwszym uruchomieniu strony z formantem *ActiveX* w górnej części okna przeglądarki pojawi się pasek z komunikatem *Ta witryna sieci Web chce zainstalować następujący dodatek:*. Należy kliknąć ten pasek, a następnie wybrać *Zainstaluj formant ActiveX*.

Efektów działania szkodliwego oprogramowania jest wiele. Może to być uruchomienie niegroźnego programu lub utrata wszelkich plików z dysków. Niektóre programy mogą wykraść ważne osobiste dane (takie jak numer i hasło do konta bankowego) i przekazać je niepowołanym osobom, umożliwić włamanie do komputera i przejęcie nad nim kontroli (tzw. *zombie*). Zombie to komputer zainfekowany i działający w sieci, który może służyć między innymi do masowej wysyłki e-maili, a także jako jedno z ogniw do przeprowadzania rozległych ataków **DDoS**.

Atak DDoS (ang. *Distributed Denial of Service* — rozproszony atak odmowy usługi) polega na jednoczesnym atakowaniu ofiary z wielu miejsc. Służą do tego najczęściej komputery, nad którymi przejęto kontrolę przy użyciu specjalnego oprogramowania (zainfekowane sieci mogą liczyć nawet 100 000 komputerów *zombie*). Na dany sygnał komputery zaczynają jednocześnie atakować system ofiary, zasypując go fałszywymi próbami skorzystania z usług, jakie ten system oferuje. Dla każdego takiego wywołania atakowany komputer musi przydzielić pewne zasoby (pamięć, czas procesora, pasmo sieciowe), co przy bardzo dużej liczbie żądań prowadzi do wyczerpania dostępnych zasobów, a w efekcie do przerwy w działaniu lub nawet zawieszenia systemu. Użytkownicy zainfekowanych maszyn najczęściej nie zdają sobie sprawy, że ich komputery zostały wykorzystane do nielegalnych celów. Niektóre ze skryptów przeprowadzających atak po jego zakończeniu mogą samoczynnie odinstalować się, nie pozostawiając żadnych śladów swojej obecności w systemie.

Groźba ataku DDoS bywa czasami używana do szantażowania firm, dla których przerwa w działaniu systemu przekłada się na bezpośrednie straty finansowe firmy i jej klientów. W takich przypadkach osoby stojące za atakiem żądają okupu za odstąpienie od niego lub jego przerwanie. **Taki szantaż jest przestępstwem!**

Atak typu DoS (ang. *Denial of Service* — odmowa usługi) polega na wysłaniu do serwera-ofiary bardzo dużej liczby zapytań (na przykład kilku tysięcy odwołań do tej samej strony internetowej). Powoduje to przeciążenie i znaczne spowolnienie pracy atakowanego serwera lub zawieszenie pojedynczej usługi sieciowej. W wielu przypadkach pro-

wadzi to do zdestabilizowania pracy serwera. Atak DoS jest dokonywany z pojedynczego komputera, więc zlokalizowanie sprawcy sprowadza się do określenia jego numeru IP. Reszta należy do organów ścigania.

Zatruwanie DNS (ang. *DNS cache poisoning*) to technika *phishingu* polegająca na wysłaniu do serwera DNS fałszywego rekordu kojarzącego nazwę domeny z adresem IP. Serwer DNS zapamiętuje go na pewien czas (do kilku godzin) i zwraca klientom zapamiętany adres IP, czego skutkiem jest przekierowanie na fałszywą stronę.

3.4.9. Zadania

Zadanie 3.3. Co nazywamy siecią komputerową? Wymień i opisz rodzaje sieci komputerowych. Podaj przykłady różnych sieci.

Zadanie 3.4. Odpowiedz na poniższe pytania.

1. Jakie zadania pełnią poszczególne warstwy modelu OSI?
2. Co to jest protokół sieciowy?
3. Na czym polega komunikacja między hostami w ramach modelu OSI?
4. Co to jest pakiet i ramka — do jakich warstw modelu OSI należą?
5. Scharakteryzuj transmisję unicast, multicast i broadcast.
6. Kiedy można powiedzieć, że użytkownik jest za NAT-em?
7. Co oznacza, że protokół jest nierutowalny? Podaj przykład protokołu nierutowalnego.
8. Jaką funkcję pełnią serwery DNS?
9. Co to jest adres URL? Podaj przykłady.
10. Podaj przykłady adresów specjalnych.
11. Porównaj zapis adresów IPv4 oraz IPv6.
12. W jaki sposób działa protokół SSL?

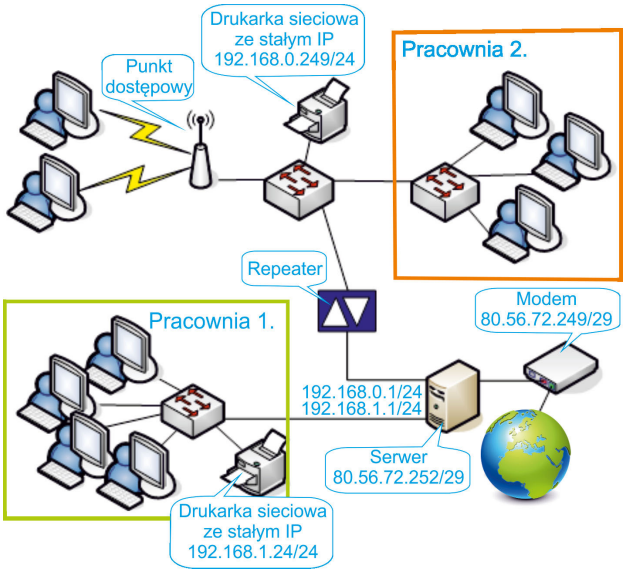
Zadanie 3.5. Przedstaw kilka urządzeń sieciowych — omów je i podaj przykładowe zastosowania.

Zadanie 3.6. Wymień klasy adresowe IPv4. Ile hostów można zaadresować w poszczególnych klasach?

Zadanie 3.7. Wymień trzy topologie sieci LAN. Narysuj sieć LAN w tych topologiach. W jakiej topologii są połączone komputery w szkolnej pracowni komputerowej?

Zadanie 3.8. Masz adres IP `31.11.0.0/24`. Zaproponuj projekt sieci LAN w topologii drzewiastej, obsługującej: dwie pracownie komputerowe (po 15 stanowisk) Twojej szkoły, dwie pracownie gimnazjum (po 10 komputerów), część administracyjną szkoły (5 komputerów), dwa centra multimedialne (po 4 komputery), kawiarenkę (6 stanowisk), pokój nauczycielski (3 stanowiska).

Zadanie 3.9. Przyjrzyj się uważnie zaprojektowanej sieci, której schemat zamieszczono na rysunku 3.64. Określ, jakie adresy IP przypiszesz komputerom w pracowniach 1. i 2.



Rysunek 3.64. Projekt sieci do zadań od 3.9 do 3.12

Zadanie 3.10. Na rysunku 3.64 zamieszczono *repeater*. Jak sądzisz, co spowodowało jego zastosowanie? Dlaczego nie ma drugiego takiego urządzenia w segmencie sieci obsługującej pracownię 1.?

Zadanie 3.11. Ile kart sieciowych znajduje się w serwerze pokazanym na rysunku 3.64? Podaj ich adresy sieciowe.

Zadanie 3.12. Napisz adresy rozgłoszeniowe obu utworzonych podsieci w projekcie zamieszczonym na rysunku 3.64.

Zadanie 3.13. Co przyczyniło się do powstania protokołu nowej generacji IPv6? Scharakteryzuj ten protokół.

Zadanie 3.14. Uproszczony zapis adresu IPv6 podano następująco: 2004:ac5::428:35a2. Podaj w zapisie heksadecymalnym, dziesiętnym oraz binarnym pełny zapis 8 członów tego adresu.

Zadanie 3.15. Wymień podstawowe zagrożenia, które powstają po podłączeniu komputera do sieci. W jaki sposób zabezpieczysz swój komputer podłączony do internetu?

3.5. Podstawy tworzenia stron WWW

Definicja

Hipertekst to sposób gromadzenia i prezentowania informacji umożliwiający tworzenie struktury danych i pozwalający na wyszukanie potrzebnej informacji, bez konieczności czytania całych tekstów.

Hipertekst cechuje organizacja danych w postaci niezależnych dokumentów hipertekstowych (*węzłów*) zawierających porcję informacji i połączonych *hiperlinkami* (odsyłaczami, odnośnikami, hiperłączami).

Definicja

WWW to hipertekstowy, multimedialny, sieciowy system informacyjny, którego podstawowym zadaniem jest publikowanie informacji.

W systemie WWW węzłem jest strona WWW.

3.5.1. Podstawowe elementy usługi WWW

Język HTML (ang. *HyperText Markup Language* — hipertekstowy język znaczników) — system formatowania dokumentów.

Protokół HTTP (ang. *HyperText Transfer Protocol* — protokół przesyłania dokumentów hipertekstowych) określa sposób komunikacji między przeglądarką i serwerem w internecie.

Przeglądarka internetowa — program zainstalowany na komputerze użytkownika, tzw. **klient** wysyłający do serwera żądanie przesłania dokumentu (wybranego przez adres URL). Przeglądarka interpretuje kod HTML i wyświetla otrzymaną stronę w odpowiednio sformatowanej postaci.

Serwer WWW — program, który udostępnia w internecie dokumenty i w odpowiedzi na żądanie wysyła je do przeglądarki.

3.5.2. Projektowanie witryny WWW

Na etapie projektowania należy przemyśleć wiele zagadnień, tak aby przystąpić do tworzenia strony ze szczegółowym planem pracy. Trzeba zatem:

- określić cele i grupy adresatów, czyli ustalić, do kogo kierujemy stronę;
- odpowiednio dobrać zawartość witryny;
- zaplanować układ stron;
- zaplanować graficzną szatę witryny;
- zbadać funkcjonalność i działanie witryny (sprawdzić hierarchię strony) już na etapie projektu.

Przy projektowaniu witryny należy zadbać o czytelność zamieszczanych komunikatów i treści, czytelność struktury witryny (odpowiedni podział na funkcjonalne bloki, elementy), właściwą konstrukcję systemu nawigacji witryny.

3.5.3. Popularne formaty hipertekstowe w sieci WWW

HTML (ang. *HyperText Markup Language*)

Definicja

Dokument HTML jest zwykłym dokumentem tekstowym składającym się z elementów (znaczników) HTML.

Znaczniki HTML opisują sposób formatowania dokumentu. Zestaw znaczników dla języków HTML i XHTML (w przeciwieństwie do języka XML) jest ustalony.

Ciekawostka

Zaprojektowany przez organizację *W3C* rozszerzalny język znaczników *XML* (ang. *eXtensible Markup Language*) ma inne zadanie. Przede wszystkim nie jest to język do tworzenia dokumentów, lecz… innych języków. Dopiero one służą do budowania dokumentów w internecie. Języki takie nazywamy *aplikacjami XML* — zapewne niektórym znana jest choćby nazwa MathML, oznaczająca aplikację XML służącą do tworzenia dokumentów zawierających formuły matematyczne. Inną znaną aplikacją jest WAP służący do wyświetlania internetowej informacji w telefonach komórkowych. Są też takie narzędzia jak: GedML (obsługa danych genealogicznych), MusicML (notacja muzyczna), VoiceML (dane głosowe), ThML (teksty teologiczne), VML i SVG (grafika wektorowa), XMLNews (wymiana aktualnych wiadomości), RSS (kanały informacyjne) i inne. Powstało mnóstwo takich aplikacji i są to języki wykorzystywane w biznesie, nauce, wyspecjalizowanych mediach itp.

Oczywiście najbardziej znaną aplikacją XML jest język XHTML, czyli następca HTML-a.

Prawdopodobnie każda osoba może być twórcą aplikacji XML, jeśli tylko starczy jej wiedzy, czasu i cierpliwości. Jeżeli będzie miała dobry pomysł, jej język może stać się standardem w jakiejś wybranej dziedzinie.

Element HTML zazwyczaj składa się z trzech części:
- znacznika otwierającego: `<znacznik>`;
- zawartości elementu, może nim być na przykład tekst, obraz;
- znacznika zamykającego: `</znacznik>`.

Zalecenia i wskazówki dla autorów stron WWW.
- W języku HTML wielkość liter w znacznikach jest dowolna. Zaleca się jednak stosowanie małych liter, które obowiązują w XHTML-u.
- Należy domykać znaczniki.
- Znaczniki powinny być poprawnie zagnieżdżane.
- Dokument HTML można utworzyć za pomocą prostego edytora tekstów, nawet Notatnika, zaleca się jednak stosowanie specjalnego edytora, jakim jest na przykład edytor kED, który spełnia powyższe wymogi dotyczące tworzenia kodu (*http://www.dobreprogramy.pl/kED,Program,Windows,11810.html*). Jest to prosty edytor tekstu udostępniony jako freeware, z obsługą kolorowania składni głównie w językach: (X)HTML, CSS, PHP, JavaScript oraz w takich językach jak: VBScript, XML, SQL, Perl, Python i WML. Podobne usługi oferują również inne edytory, na przykład coraz popularniejszy Notatnik+, polecany jako edytor przydatny dla webmasterów.

Pierwsza oficjalna odsłona języka HTML 4.0 odbyła się w 1997 roku, a ostatnia poprawka do wersji 4.01 została wprowadzona w 1999 roku. Można pomyśleć, że język HTML to już przeszłość. Na szczęście niezupełnie jest to prawdą.

XHTML (ang. *Extensible HyperText Markup Language*)

XHTML — czyli rozszerzalny język znaczników hipertekstowych — reformuje język HTML 4 w taki sposób, aby był zgodny z XML-em (ang. *Extensible Markup Language* — rozszerzalny język znaczników).

Język XHTML, który jest aktualnym standardem w tworzeniu serwisów WWW, opiera się na specyfikacji HTML 4.01, tzn. definiuje dokładnie te same znaczniki, atrybuty i sposoby budowania dokumentów. Jednak wprowadza przy tym pewne reguły, które pozwalają dostosować składnię do języka XML. Zatem zdobyta wcześniej wiedza o języku HTML nie pójdzie na marne i nie musisz się uczyć wszystkiego od nowa. Aby pisać dokumenty zgodne z aktualnymi standardami i trendami, wystarczy że przyswoisz sobie kilka dodatkowych, prostych zasad.

Zacznijmy od podobieństw do języka HTML:

- Dane mają formę tekstową i opisywane są znacznikami (tagami), na przykład: `<img>`.
- Znaczniki mogą mieć atrybuty, na przykład: `src="rysunek.jpg"`. Wartości atrybutów są ujmowane w cudzysłów, na przykład: `"rysunek.jpg"`.
- Znaczniki rozpoczynają się znakiem otwierającym, na przykład: `<html>`, a kończą (jeśli jest taka potrzeba) znakiem zamykającym `</html>`.
- Komentarze zawierają się w znaczniku `<!--komentarz-->` i mogą być używane w całym dokumencie.

Niektóre nowości w porównaniu z HTML-em:

- Znaczniki puste (logiczne) muszą być zawsze zakończone `/>`, na przykład: `<br />`, `<hr />`.
- Wszystkie atrybuty i znaczniki muszą być pisane małymi literami, na przykład: `<img src="obrazek.jpg" />`.
- Zlikwidowano wiele znaczników — (zamiast nich należy używać kaskadowych arkuszy stylów), na przykład: `<font>`, `<basefont>`, `<strike>`, `<s>`, `<u>`, `<center>`.
- Zlikwidowano wiele znaczników, które należy zastąpić innymi, na przykład zamiast `<menu>` należy używać znacznika `<ul>`, `<dir>` — `<ul>`, `<isindex>` — `<input>`, `<xmp>` — `<pre>`, `<plaintext>` — `<pre>`, `<listing>` — `<pre>`, `<applet>` — `<object>`.
- Zlikwidowano szereg atrybutów, które należy zastąpić stylami, jednakże wielu z nich można używać do tej pory, na przykład `border` w dalszym ciągu określa grubość krawędzi w tabeli i jest stosowany w znaczniku `<table>`, natomiast atrybut `width` może być stosowany w celu określenia szerokości w znacznikach `<colgroup>`, `<col>`, `<table>`, `<img>`, `<object>` i `<iframe>`. Mimo że tagi, takie jak `<b>`, `<i>`, znajdują się w specyfikacji, zaleca się stosowanie stylów. W przyszłości wszystkie atrybuty i znaczniki opisujące stronę pod względem wizualnym zostaną zlikwidowane (na przykład: `size`, `width`, `height`, `border`, `align`).
- Znaczniki nie mogą być skrzyżowane. W XHTML-u ważna jest kolejność otwierania i zamykania znaczników. Przykład niepoprawnego zapisu tekstu: `Wlazł <i>kotek <b>na </i>płotek.</b>`.

3.5.4. Struktura dokumentu

Każdy dokument (X)HTML musi mieć znaczniki początku dokumentu, jego końca oraz wydzielone dwie sekcje: nagłówka i ciała (treści) strony. Struktura dokumentu w największym uproszczeniu wygląda następująco:

```
<html>
<head>
</head>
<body>
</body>
</html>
```

Dołóżmy jeszcze informację na początku dokumentu i kilka słów wyjaśnienia w HTML-u:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html401/strict.dtd"> <!--Prolog — ten tekst jest komentarzem-->
<html>
<head>
<!--Nagłówek dokumentu, tu podajemy informacje dodatkowe i tytuł stron -->
</head> <!--Koniec sekcji nagłówka-->
<body>
<!--Treść dokumentu -->
</body> <!--Koniec sekcji treści dokumentu -->
</html> <!--Koniec dokumentu -->
```

I krótki przykład w XHTML-u:

```
<?xml version="1.0" encoding="iso-8859-2"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Przykładowa strona</title>
</head>
<body>
<p>Przykładowy akapit</p>
<hr />
<img src="rysunek.jpg" alt="tekst" /> <!--Koniecznie używaj atrybutu alt i zwróć uwagę na zakończenie pustego znacznika-->
</body>
</html>
```

Prolog

W dokumentach WWW często stosuje się specjalny prolog, który identyfikuje poziom używanego języka. Jest on wstawiany jako pierwszy element dokumentu, jeszcze przed otwarciem szkieletu strony.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html401/strict.dtd">
```

Ten dokument jest zgodny ze specyfikacją HTML 4.0.

```
<?xml version="1.0" encoding="iso-8859-2"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/
xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

A ten dokument jest zgodny ze specyfikacją XHTML.

Nagłówek

`<meta>` to opcjonalne polecenia, które szczegółowo informują o zawartości dokumentu. Są wykorzystywane przez odpowiednie programy, na przykład przez wyszukiwarki sieciowe. Podawanie tych dodatkowych informacji jest zalecane. Oto krótkie podpowiedzi dotyczące kilku atrybutów znacznika `<meta>`:

```
<head>
<meta http-equiv="Content-type" content="text/html; charset=iso-8859-2">
<!--Kodowanie polskich liter -->
<meta name="Description" content=" [tu wstaw krótki opis strony] ">
<meta name="Keywords" content=" [wstaw tu słowa kluczowe] "> <!--Ważne dla
programów wyszukiwawczych-->
<meta name="Author" content=" [dane autora] ">
<meta http-equiv="content-language" content="pl">
<meta name="Generator" content="kED2">
<title> [tytuł strony] </title> <!--Do 40 znaków, koniecznie go używaj, pokaże się na
pasku tytułowym przeglądarki-->
</head>
```

Wskazówka

W dokumentach należy stosować standard kodowania polskich liter według normy *ISO-8859-2* lub *UTF-8*. Są to standardy dostępne w wielu systemach operacyjnych. Standard kodowania *win-1250* jest typowy tylko dla Windows. Nie należy stosować go na stronach WWW. (W edytorze kED, w zakładce *Plik/Zmień stronę kodową...*, można zmienić sposób kodowania polskich znaków).

Treść dokumentu (sekcja body)

Tekst na stronie WWW

Ta sekcja jest najważniejszą częścią dokumentu. To tutaj pojawia się treść strony.

Gdy redagujemy tekst w edytorze tekstu, jesteśmy przyzwyczajeni do używania klawisza *Enter* w celu oddzielenia od siebie kolejnych akapitów. Jednak przeglądarki ignorują ten znak i na ekranie akapity zostaną złączone w ciągły tekst. Dlatego każdy akapit powinien być umieszczony między parą znaczników `<p> </p>`. Akapity są oddzielane dodatkową interlinią, która podkreśla ten podział. Można również użyć pojedynczego (nazywanego otwartym) znacznika `<br />`, który zapewnia efekt miękkiego przejścia do nowej linii.

Należy unikać powiększania i wyróżniania tekstu tytułów przez zmianę wyglądu tekstu. Jeżeli chcesz zamieścić na stronie tytuł rozdziału lub podrozdziału, koniecznie wykorzystaj do tego celu znaczniki nagłówkowe, na przykład: `<h1>tu wpisz tytuł pierwszego stopnia</h1>`.

Ciekawostka

Osoby niewidome i niedowidzące również przeglądają strony WWW w poszukiwaniu informacji. Korzystają z dużych monitorów, ustawionych na małą rozdzielczość ekranu, na przykład 800×600 pikseli. Siedzą przy monitorze z dużą lupą i śledzą to, co czyta im program dla niepełnosprawnych. Syntezator mowy wyszukuje najważniejsze treści, a te są ujęte w znaczniki `<h1>` do `<h6>`. Jeżeli więc na naszej stronie nie wskażemy tego, co jest ważne, niepełnosprawna osoba nic nie usłyszy. Dla niej nasza strona będzie pusta (cicha). To samo dotyczy omawianej nieco później grafiki wstawianej na stronę. Niewidoma osoba usłyszy tekst opisujący obraz. Będzie więc poinformowana o treści strony.

Elementy blokowe służą do formatowania całych bloków tekstu. Oto kilka z nich:

`<h1> </h1>` — nagłówki; hierarchia tekstu jak w spisie treści (od `h1` do `h6`),

`<p> </p>` — akapit,

`<div> </div>` — wydzielenie bloku tekstu w celu sformatowania,

`<br />` — znak końca wiersza,

`<hr />` — linia pozioma,

`<ul><li> </li></ul>` — listy nieuporządkowane,

`<ol><li> </li></ol>` — listy uporządkowane,

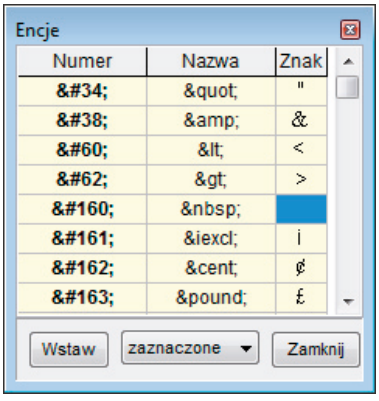
`<blockquote> </blockquote>` — cytat.

Znaczniki wizualne (prezentacyjne) służą do jawnego określenia ostatecznego i jednoznacznego wyglądu tekstu na stronie.

Przykłady znaczników prezentacyjnych: `<b>`, `<i>`, `<tt>`, `<sub>`, `<sup>`, `<u>`, `<font>`.

Jednak znaczniki te (ujęte w specyfikacji HTML 4) uznawane są obecnie za schyłkowe — skutecznie wyparły je *arkusze stylów*, w których opisujemy szczegóły takie jak: kroje czcionek, odstępy między wierszami czy kolory.

Skoro przeglądarki ignorują dodatkową spację, to wykorzystać można tzw. *znaki specjalne*. Można stosować opisowy sposób zapisu tych znaków (i znaków narodowościowych innych języków) z wykorzystaniem tzw. *encji*. Oto przykłady: `<`, ` `, `&duml;`, `&copy`. Całą listę encji można zobaczyć w programie kED po wybraniu z menu *Opcje/Tabela encji* (rysunek 3.65).



Rysunek 3.65. Tabela encji w programie kED

Grafika na stronie WWW

Do zamieszczania grafiki na stronach WWW służy znacznik `<img>`, który ma następujące atrybuty: źródło obrazu (obowiązkowy atrybut), szerokość obrazu, wysokość obrazu i opis (koniecznie dołączaj opis ze względu na osoby niedowidzące — usłyszą ten opis dzięki syntezatorowi mowy).

```
<img src="obrazek.png" width="120" height="160" alt="opis obrazu" />
```

Wskazówka

Niedopuszczalne jest, aby dużych rozmiarów obraz był zmniejszany w kodzie WWW. Należy najpierw przygotować obraz o odpowiednio zmniejszonych wymiarach, a następnie zamieścić go na serwerze WWW. Strona z kilkoma wielkimi obrazami będzie się bardzo długo wczytywała i może spowodować zniechęcenie osoby odwiedzającej, a w końcu — rezygnację z otwarcia strony.

Scalable Vector Graphics (SVG) — nowy standard grafiki w sieci WWW

Format SVG (ang. *Scalable Vector Graphics* — skalowalna grafika wektorowa) to stworzona w 1999 roku przez W3C aplikacja XML opisująca dwuwymiarową (2D) statyczną i animowaną grafikę wektorową na stronach WWW. Szerzej format ten jest opisany w rozdziale 4. Tam też zostały przedstawione ćwiczenia w programie Inkscape, których wynikiem są proste grafiki wektorowe *.svg*. Tutaj natomiast opisujemy sposoby zamieszczenia tego formatu na stronie WWW.

Niektóre przeglądarki, na przykład Amaya, której rozwój wspiera organizacja W3C, a także Chrome i Opera (począwszy od wersji 8 beta 3), gwarantują jego pełną obsługę. Konqueror i Mozilla Firefox (od wersji 1.5) radzą sobie z nim coraz lepiej. Inne przeglądarki, jak na przykład Internet Explorer, mogą wyświetlać pliki SVG, jednak wymagają zainstalowania dodatkowej wtyczki, na przykład firmy Adobe SVGViewer (*http://www.adobe.com/svg/viewer/install/*).

Jest również możliwe, że SVG zastąpi format grafiki wektorowej Macromedia Flash. Do tworzenia grafiki w tym formacie można użyć bezpłatnych programów, na przykład Inkscape. Format SVG nie podlega prawom patentowym.

Jak zamieścić plik .svg na stronie WWW?

Grafikę SVG możemy wyświetlić na kilka różnych sposobów.

Można zastosować odwołanie do zewnętrznego dokumentu *obraz.svg*:

```
<a href="obraz.svg">kliknij tutaj i zobacz plik obraz.svg</a>
```

Z tym rozwiązaniem obecnie radzi sobie każda przeglądarka, prawidłowo wyświetlając grafikę SVG. Niektóre przeglądarki pozwalają również na ich skalowanie, dokładając do podręcznego menu stosowne polecenia: *powiększ*, *zmniejsz*.

Można użyć znacznika `<img>`:

```
<img src="svg/obraz.svg" height="100" width="100" alt="rysunek svg"/>
```

To rozwiązanie wydaje się najlepsze. Poprawnie działa w przeglądarkach Opera i Amaya. Inne przeglądarki z pewnością niebawem do nich dołączą.

Można użyć znacznika `<object>`:

```
<object data="svg/obraz.svg" height="100" width="100" type="image/
svg+xml"/>[Twoja przeglądarka nie obsługuje formatu SVG]
</object>
```

Dokumenty SVG mają przypisany typ MIME (ang. *Multipurpose Internet Mail Extensions* — wielozadaniowe rozszerzenia internetowej poczty elektronicznej): `image/svg+xml` (dawniej `image/svg-xml`).

Można użyć znacznika `<embed>`:

```
<embed src="svg/obraz.svg" height="100" width="100" alt="rysunek svg"
pluginspace="http://www.adobe.com/svg/viewer/install/">
```

Znacznik `<embed>` działa, jednak nie jest zalecany przez W3C.

SVG to język znacznikowy zgodny ze specyfikacją XML. Pliki *.svg* można edytować zarówno w edytorze graficznym, jak i tekstowym. Narysowane w Inkscape koło i prostokąt, zapisane w pliku *figury.svg*, można edytować w Notatniku. Poniżej zamieszczono najważniejszy fragment tego pliku:

```
<svg>
<circle style="fill: red; fill-opacity: 3; stroke: green" cx="100" cy="100"
r="50" />
<rect style="fill: yellow; fill-opacity: .5; stroke: black"  x="10" y="10"
width="100"
height="50" />
</svg>
```

gdzie:

`fill` — wypełnienie,

`stroke` — kontur (krawędź).

Sprawdź to samodzielnie. Dzięki temu możliwe jest pisanie kodu rysunku SVG bezpośrednio w kodzie strony WWW.

Odsyłacze, hiperłącza, linki

Odsyłacze — inaczej hiperłącza, odnośniki hipertekstowe, linki — stanowią istotę internetu. Bez nich dokumenty znajdujące się w sieci nie byłyby w żaden sposób ze sobą powiązane. Praktycznie na każdej stronie WWW umieszczone są odsyłacze. Najczęściej stanowi je specjalnie wyróżniony krótki tekst (lub obrazek), po którego kliknięciu następuje na przykład przeniesienie do innej strony, przy czym strona taka może wchodzić w skład tego samego serwisu, ale równie dobrze może znajdować się na drugim końcu świata. Przeglądarki internetowe wyświetlają odsyłacze najczęściej jako tekst podkreślony i w innym kolorze — dzięki temu odsyłacze są lepiej widoczne i odróżniają się od zwykłego tekstu.

Użytkownika można odsyłać również do cudzych stron, a nie tylko do własnych — i nie jest to łamanie praw autorskich, przeciwnie, to darmowa reklama strony, do której następuje odwołanie. Ważne jest jedynie, aby dokumenty były w jakiś sposób powiązane ze sobą tematycznie (chociaż nie jest to wymóg). To właśnie istota hipertekstu — pozwala on uzyskać system połączeń pomiędzy różnymi dokumentami w sieci.

Odsyłacz do adresu internetowego

```
<a href="http://adres strony">opis</a>
```

W miejsce `"http://adres strony"` należy podać pełny adres internetowy strony, do której chcemy się przenieść (na przykład *www.onet.pl*). Adres strony internetowej zawsze musi rozpoczynać się od `http://` lub `https://`.

Jeśli podczas surfowania w sieci trafimy na jakąś ciekawą stronę, możemy skopiować jej adres (z paska adresu przeglądarki) i umieścić w odsyłaczu. Dzięki temu wszyscy użytkownicy odwiedzający naszą witrynę będą mogli przenieść się bezpośrednio do wskazanej strony, bez potrzeby wpisywania jej adresu.

Odsyłacz do podstrony (w obrębie tej samej witryny)

```
<a href="względna ścieżka dostępu">opis</a>
```

Tak jak każda książka składa się z rozdziałów, tak samo witryna internetowa i serwisy internetowe składają się zwykle z podstron. Każda podstrona jest osobnym plikiem HTML i zawiera treść różniącą się tematycznie od treści pozostałych podstron. Taka organizacja ułatwia użytkownikowi nawigację po witrynie oraz zapobiega niepotrzebnemu wczytywaniu całego serwisu od razu (co trwałoby prawdopodobnie bardzo długo).

Odsyłacz do pliku

```
<a href="ścieżka_do_pliku">Tekst będący hiperłączem</a>
```

Pliki filmowe, muzyczne, dokumenty PDF czy archiwa ZIP zamieszczone jako dokument w odnośniku spowodują pojawienie się okna dialogowego z zapytaniem, czy plik ma zostać otwarty w domyślnej aplikacji, czy też zapisany na dysku.

```
<a href="ścieżka_do_pliku.jpg">zobacz zdjęcie</a>
```

Jeżeli hiperłącze do grafiki odwołuje się do obszernego pliku, powinno być połączone z dodatkową informacją w atrybucie `title` znacznika `<a>`, w którym — oprócz krótkiego opisu obrazu — należy podać rozmiar pliku. Atrybut ten pojawi się w dymku przy zbliżeniu wskaźnika myszy do odnośnika.

```
<a href="duzy.jpg" title="4,5 MB"><img src="maly.jpg"/></a>
```

Można również zastosować znacznik `<img>` i utworzyć z niego hiperłącze do większego zdjęcia, co często jest stosowane w galeriach zdjęć. Pamiętaj jednak, że zdjęcie pokaże się na ekranie w lewym górnym rogu przeglądarki na białym tle. Nie ma tu bowiem możliwości ustalania sposobu jego wyświetlenia.

Odsyłacz do etykiety (zakładki)

Odsyłacz do etykiety, czyli do miejsca na stronie WWW, jest najczęściej stosowany przy tworzeniu spisu treści serwisu. Odwołanie do etykiety wymaga dwóch czynności: utworzenia docelowej etykiety (zakładki) oraz wpisania na tej samej lub innej stronie odnośnika do utworzonej zakładki.

Krok 1. Utworzenie etykiety:

```
<a name="nazwa_etykiety"></a>
```

Krok 2. (Wariant pierwszy). Wstawienie odsyłacza na tej samej stronie, na której utworzono etykietę:

```
<a href="#nazwa_etykiety">Tu kliknąć</a>
```

Krok 2. (Wariant drugi). Wstawienie odsyłacza na innej stronie:

```
<a href="strona_z_etykietą.html#nazwa_etykiety">Tu kliknąć</a>
```

Odsyłacz pocztowy

```
<a href="mailto:adres e-mail">napisz do mnie</a>
```

Dzięki temu odsyłaczowi użytkownicy będą mogli wysłać e-mail na adres podany w poleceniu. Po kliknięciu takiego odnośnika list nie zostanie wysłany natychmiast, lecz nastąpi otwarcie domyślnego programu pocztowego użytkownika (na przykład poczta usługi Windows Live). Jeśli na opublikowanej przez siebie stronie w odsyłaczu pocztowym podasz adres swojej poczty elektronicznej, będziesz w stałym kontakcie z użytkownikami odwiedzającymi stronę.

3.5.5. Kaskadowe arkusze stylów

Definicja

Arkusz stylów to zestaw reguł, które znacznikom języka HTML przypisują wybraną charakterystykę. Jest przeznaczony do formatowania dokumentów opracowanych w języku HTML.

W 1996 roku opracowano standard CSS (ang. *Cascading Style Sheets* — kaskadowe arkusze stylów), którego specyfikację można znaleźć pod adresem: *http://www.w3.org/Style/CSS/*.

Zalety CSS:

- przeniesienie opisów formatowania elementów do arkuszy stylów, co pozwoliło na oddzielenie struktury i treści dokumentu od jego formy;
- zwiększenie przejrzystości kodu;
- zwiększenie szybkości wczytywania witryn;
- skrócenie kodu HTML;
- uzyskanie większych możliwości formatowania widoku strony.

Reguła stylu ma następujące składniki:

```
selektor {własność: wartość}
```

gdzie:

selektor = znacznik HTML (na przykład `p`, `li`, `td`, `body`).

Przykład pochodzi z szablonu arkusza CSS w programie kED (z menu *Plik* trzeba wybrać *Szablon arkusza CSS*):

```
h3
{font-size: 2em;
 font-weight: bold;
}
```

Inne przykłady cech i wartości, oprócz zamieszczonych w szablonie arkusza stylów (w kED), znaleźć można na stronach opisujących specyfikacje CSS, na przykład *http://www.w3.org/TR/CSS2*.

Położenie (lokalizacja) reguł stylów

Są trzy sposoby umieszczenia reguł stylów CSS w dokumentach HTML.

1. **Styl wpisany w linii** (*inline*) jest sposobem wykorzystującym atrybut `style`. Poniżej umieszczony jest przykład użycia tej metody do zmiany koloru tła na czerwony i formatowania dwóch akapitów tekstu:

```
<html>
<head>
<title>Strona z CSS w linii - z atrybutem style</title>
</head>
<body style="background-color: #FF0000;">
<p>Tło tej strony jest czerwone</p> <!--Styl dla tego akapitu nie został zmieniony-->
<p style="color: blue; font-size: 12pt">tekst niebieski wysoki 12 pt</p> <!--Zmieniony styl akapitu-->
<p style="color: green">Zielony napis</p> <!--Zmieniony styl akapitu-->
</body>
</html>
```

   Ze względu na niewydajność nie jest to metoda zalecana. Zadaniem stylów jest przecież ułatwianie pracy i maksymalne jej skracanie. Jedna z ważnych funkcji CSS

to możliwość wykorzystywania raz zdefiniowanych stylów w kilku dokumentach, czego ta metoda nie oferuje.

2. **Styl osadzony** dotyczy całego dokumentu i jest umieszczony na początku strony, w sekcji `head`. Opis formatowania strony zamieszcza się w znaczniku `<style>`.

```
<html>
<head>
<title>Strona z CSS osadzonym wewnątrz dokumentu</title>
<style type="text/css"> <!--Początek deklaracji stylów osadzonych dla całego dokumentu-->
body {background-color: #FF0000;}
p {font-size: 12px; font-family: monospace} <!--Ustalenie wyglądu czcionki dla akapitu-->
h1,h2 {text-align: center; color: green} <!--Ustalenia dla nagłówków 1. i 2. stopnia-->
</style> <!--Koniec stylów osadzonych-->
</head>
<!-- Dalsza część dokumentu -->
```

Ta metoda nie jest zalecana. Stosuje się ją jedynie w przypadku jednostronicowego dokumentu.

3. **Zewnętrzny arkusz stylów** znajduje się na serwerze, jako plik tekstowy z rozszerzeniem *.css*. Zawiera on definicje stylów. Natomiast w dokumencie musi pojawić się odwołanie do tego zewnętrznego arkusza stylów, co realizujemy przez wpisanie w sekcji `head` następującej linii kodu:

```
<link rel="stylesheet" href="styl.css" type="text/css" /> <!--Nazwa pliku może być inna, na przykład: moj_styl.css.-->
```

W celu zachowania zgodności z HTML-em w dokumentach XHTML zaleca się umieszczanie podwójnej deklaracji wywołania arkusza stylów:

```
<head>
<link rel="stylesheet" type="text/css" href="styl.css" />
<?xml-stylesheet type="text/css" href="styl.css" ?>
</head>
```

A w pliku *styl.css* należy zamieścić definicje stylów. Oto krótki przykład zawierający te same definicje, co w przykładzie stylów osadzonych:

```
p { font-size: 12px; font-family: monospace }
h1,h2 { text-align: center; color: red }
```

Jest to metoda najczęściej spotykana. Daje ona najwięcej możliwości i w pełni wykorzystuje funkcje stylów. Po stworzeniu zewnętrznego pliku CSS można go wykorzystać w kilku stronach. Daje to wymierne korzyści wtedy, gdy należy na przykład wprowadzić korekty w wyglądzie wielostronicowej witryny. Łatwiej jest wówczas dokonać zmian w jednym miejscu, a przy tym można oszczędzić czas i zyskać pewność, że wprowadzone zmiany zostaną uwzględnione na wszystkich stronach.

Kaskadowość, czyli hierarchia i dziedziczenie

Kaskadowość polega na tym, że każdy typ stylu (ze względu na swoje położenie) ma inny priorytet. Istnieje ustalona hierarchia źródeł stylów. I tak, styl wpisany ma najwyższy priorytet, drugi w kolejności jest styl osadzony, a najniższy priorytet przypisuje się zewnętrznemu arkuszowi.

Kaskadowa (hierarchiczna) struktura pozwala utrzymać jednolity wygląd całej witryny i kontrolować wybrane elementy strony przy użyciu stylów położonych wyżej w hierarchii ważności.

Dziedziczenie polega na tym, że elementy niższe w hierarchii drzewa dokumentu dziedziczą formatowanie elementów leżących wyżej w hierarchii (tak jak dzieci dziedziczą po rodzicach), chyba że wyraźnie nadamy im inne formatowanie. Jeśli cała tabela ma zdefiniowaną czcionkę, wszystkie komórki, jako położone niżej w hierarchii, mają taką właśnie czcionkę, chyba że zdefiniujemy wyjątek dla jednej konkretnej komórki. Będzie ona miała inne formatowanie ze względu na kaskadowość stylów.

Jak widać z tego przykładu, kaskadowość i dziedziczenie uzupełniają się, tworząc system sterujący działaniem stylów. Kaskadowość obrazuje hierarchię źródeł stylów, a dziedziczenie wpływa na wygląd strony z punktu widzenia hierarchii elementów w tym dokumencie.

Przykłady kodu

Dla porównania prześledź kilka przykładów, w których pokazano dwie metody deklarowania tych samych parametrów. Jedna metoda jest już przestarzała (ale nadal często spotykana), a druga wykorzystuje styl definiowany i jest metodą zalecaną.

```
<body>
<div> <!--Początek spójnego bloku tekstu -->
Tekst prosty <br/> <!--Tekst normalny, prosty-->
<i>pochylony</i><br/> <!--Pochylony tekst (przestarzały)-->
<b>pogrubiony</b><br/> <!--Pogrubiony tekst (przestarzały)-->
<span style="font-style: italic;">kursywa</span> <!--Pochylony tekst z zastosowanym stylem CSS-->
<span style="font-weight: bold;">pogrubienie</span> <!--Pogrubiony tekst z zastosowanym stylem CSS-->
<br > <!--Znak końca linii (przestarzały)-->
<br /><!--Znak końca linii (rekomendowany)-->
<font color="Blue">niebieski</font> <!--Niebieski tekst (przestarzały)-->
<span style="color: #0000ff;">niebieski</span> <!--Niebieski tekst z użytym stylem CSS-->
</div> <!--Koniec spójnego bloku tekstu-->
</body>
```

3.5.6. Publikowanie witryny w sieci WWW

Gotowe dokumenty HTML przesyłamy na serwer. Jest wiele serwerów, które udostępniają w ramach konta poczty elektronicznej możliwość opublikowania strony WWW. Proces przesyłania odbywa się z wykorzystaniem protokołu FTP. Jak przesłać pliki na serwer?

Wskazówka

Jeżeli witryna jest nieprawidłowo przygotowana, tzn. pliki są porozrzucane w różnych miejscach dysku albo ich nazwy zawierają polskie znaki diakrytyczne, spacje lub występuje niezgodność wielkości liter między nazwami plików a ich odnośnikami w kodzie strony, to może się okazać, że zamieszczona na serwerze strona, która pięknie działała na lokalnym dysku, teraz nie działa. Zamiast obrazków mamy miejsce na grafikę, a odnośnik, zamiast otwierać kolejną stronę, wyświetla komunikat o nieodnalezionej stronie.

Ćwiczenie 3.15. Opublikuj przygotowaną stronę na serwerze WWW.

Aby opublikować swoją stronę, nie musisz mieć w domu dostępu do internetu. Wszystkie operacje możesz wykonać na przykład w szkole. Twój komputer nie musi być ciągle włączony, ponieważ strona będzie znajdowała się na dysku serwera. Jedyne, co musisz zrobić, to wysłać swoją stronę na serwer. Możesz wykupić konto w dowolnej firmie zapewniającej tzw. hosting WWW. Obecnie w Polsce działa wiele firm, które oferują swoje usługi za kilka złotych miesięcznie. Jest to przydatne dla twórców stron komercyjnych, którym zależy przede wszystkim na prestiżu. Przed wykupieniem tego typu konta zwróć uwagę, czy masz do czynienia z firmą godną zaufania. Legalnie działająca firma powinna podać dokładne dane kontaktowe do tradycyjnej korespondencji, numer telefonu, NIP oraz REGON.

Jeśli chcesz opublikować niekomercyjną stronę WWW, możesz skorzystać z usług jednej z wielu firm oferujących nieodpłatnie miejsce na swoim serwerze. Darmowe konta WWW mają pewne ograniczenia, a jednym z najdokuczliwszych są reklamy, zwykle dołączane do strony automatycznie. Przeczytaj dokładnie regulamin, żeby nie trafić na jakiś haczyk.

Na płycie CD dołączonej do podręcznika, w pliku *publikowanie strony WWW.pdf*, przedstawiono kolejne etapy publikowania strony w portalu *www.onet.pl*.

Sprawdzenie poprawności dokumentu

Zaleca się sprawdzanie poprawności dokumentu w celu wyeliminowania błędów i tym samym uzyskania pewności, że dokument będzie poprawnie wyświetlany przez różne przeglądarki. Doskonałym narzędziem do tego celu jest **Validator**, dostępny pod adresem: *http://validator.w3.org/*. Wystarczy wpisać tam adres dowolnej strony WWW, aby zobaczyć raport. Marzeniem każdego webmastera jest otrzymanie raportu zwrotnego takiego jak na rysunku 3.66.



Rysunek 3.66. Raport sprawdzenia poprawności strony WWW

Jeżeli natomiast pojawią się błędy, zobaczymy raport i króciutki opis ze wskazówkami prowadzącymi do ich usunięcia. Poprawiamy więc błędy i ponownie wysyłamy poprawiony plik na serwer. Sprawdzamy stronę raz jeszcze, aż do zielonego komunikatu o poprawności wykonania strony i braku błędów. Powodzenia.

Teraz pozostaje już tylko zgłoszenie strony do katalogów wyszukiwawczych, aby jak największa liczba internautów mogła do niej dotrzeć.

3.5.7. Systemy zarządzania treścią

W czasach ogromnej popularności i powszechnej dostępności internetu założenie własnej strony internetowej nie jest wielkim wyzwaniem. Nie trzeba też przekonywać o konieczności jej posiadania. Prawie każda firma ma już własną stronę WWW albo jest w trakcie jej tworzenia.

Jeżeli strona internetowa składa się z niewielu elementów, których nie trzeba ciągle zmieniać i uaktualniać, problem z jej modyfikacją jest niewielki. Trudność sprawia dopiero zapanowanie nad serwisem z bardzo dużą ilością ciągle zmieniających się informacji, wieloma podstronami i rozbudowaną nawigacją oraz elementami interaktywnymi. Ciągła aktualizacja witryny okazuje się wtedy bardzo kłopotliwa, a zlecanie częstych modyfikacji — dosyć kosztowne. W takim przypadku najlepiej generować kod stron dynamicznie, za pomocą specjalnie przygotowanych do tego celu szablonów graficznych. Samodzielne przygotowanie tego typu oprogramowania wymaga olbrzymiej wiedzy i umiejętności oraz dużego nakładu pracy, a efekt naszych starań może być różny i nie zawsze zadowalający.

Na szczęście możemy skorzystać z gotowych i sprawdzonych narzędzi, które ułatwią nam pracę, czyli tzw. **systemów zarządzania treścią**, w skrócie zwanych **CMS** (ang. *Content Management System*). Umożliwiają one szybkie tworzenie rozbudowanych serwisów internetowych o różnym stopniu trudności bez umiejętności programowania.

System CMS pozwala na łatwe administrowanie serwisem, nie wymagając przy tym od zespołu redakcyjnego nawet znajomości języka HTML. Umożliwia kontrolowanie praktycznie wszystkich elementów witryny internetowej. CMS można wykorzystać przy tworzeniu stron WWW oraz portali, a nawet rozbudowanych sklepów internetowych.

Wszelkie działania polegające na dodawaniu, usuwaniu lub jakiejkolwiek zmianie danych wykonuje się w panelu administratora, wypełniając odpowiednio przygotowane formularze. Rola administratora sprowadza się do instalacji systemu oraz ustalenia zawartości witryny. Najpopularniejsze systemy CMS to Joomla! (*www.joomla.pl/*) oraz opisany w podręczniku Drupal.

System Drupal

Drupal jest oprogramowaniem open source rozpowszechnianym na licencji GNU GPL (ang. *GNU General Public License* — powszechna licencja publiczna), utrzymywanym i rozwijanym przez społeczność tysięcy użytkowników i programistów.

Jest to system zarządzania treścią umożliwiający użytkownikowi łatwe publikowanie i organizowanie zawartości strony, a także zarządzanie nią. Wyposażony jest w funkcje, które oferują między innymi:

- środowiska przeznaczone do wspólnej pracy wielu redaktorów nad projektem,
- galerie grafik i zdjęć,
- możliwość wysyłania i pobierania plików,
- fora,
- biuletyny,
- system blogów.

Do prawidłowego działania Drupal wymaga serwera WWW z obsługą skryptów PHP oraz bazy danych MySQL lub PostgreSQL. Warunek ten zostanie spełniony dzięki zainstalowaniu lokalnego środowiska WAMP, w którego skład wchodzą: Apache (serwer HTTP), PHP (skryptowy język programowania, służący do tworzenia dynamicznych stron WWW), MySQL (serwer relacyjnych baz danych) oraz PhpMyAdmin (umożliwia wizualną i uproszczoną w stosunku do konsolowej pracę z bazą danych MySQL).

Ciekawostka

WAMP (ang. *Windows — Apache — MySQL — PHP*) to pakiet wolnego oprogramowania do obsługi witryn internetowych w środowisku MS Windows, zawierający programy open source: serwer Apache, język skryptowy PHP, bazę danych MySQL i oprogramowanie uzupełniające.

Zestaw samego oprogramowania (AMP) działa również na innych systemach operacyjnych, takich jak: Linux (LAMP), Mac OS (MAMP), Solaris (SAMP) lub OpenBSD (Oamp). Kombinacja oprogramowania zawartego w pakiecie może się różnić, zwłaszcza w odniesieniu do oprogramowania skryptowego. PHP może być zastąpiony lub uzupełniony przez języki Perl i (lub) Python.

Ćwiczenie 3.16. Na płycie CD dołączonej do podręcznika, w pliku *CMS Drupal.pdf*, zamieszczona została instrukcja dotycząca instalacji, konfiguracji i zarządzania lokalnym serwisem Drupal. Znajdują się tam następujące szczegółowe instrukcje:

- instalacja środowiska WAMP,
- Drupal — przygotowanie do instalacji,

- instalacja serwisu Drupal,
- spolszczenie serwisu,
- podstawowa konfiguracja i instalacja dodatkowych modułów,
- strona WWW,
- baza danych MySQL serwisu Drupal,
- zmiana wyglądu serwisu,
- administracja serwisem — użytkownicy i ich prawa.

Zainstaluj lokalny serwis Drupal na swoim komputerze (środowisko Microsoft Windows) zgodnie z kolejnymi punktami instrukcji.

Stwórz prosty tematyczny serwis, w którym prace redakcyjne prowadzą administrator oraz redaktor, czyli użytkownik uwierzytelniony z prawami edycji. Zwróć uwagę na zawartość bazy danych oraz prawa do tworzenia, edycji i usuwania treści. Sprawdzaj, jak widzi serwis użytkownik anonimowy (odwiedzający).

Do wykonania tego ćwiczenia w pracowni komputerowej należy stworzyć dwu- lub trzyosobowe zespoły. Każdy uczeń jest administratorem swojego serwisu i udostępnia prawa edycji przynajmniej jednej osobie. Stwórzcie w ten sposób serwisy tematyczne. Pamiętajcie o tym, by administratorzy swoich serwisów przydzielili odpowiednie prawa użytkownikom uwierzytelnionym (redaktorom).

Wskazówka

Aby zalogować się do serwisu innego komputera w lokalnej sieci, należy poznać adres IP tego komputera i wpisać go w polu adresu przeglądarki, na przykład *http://192.168.0.165/drupal*.

Projekt edukacyjny

Założenia projektu edukacyjnego pt. *Prezentacja w sieci internet oparta na systemie zarządzania treścią*

Projekt powinien być wykonany przez zespoły uczniów (od 4 do 5 osób). Jego realizacja będzie się opierała na systemie zarządzania treścią Drupal (wraz z odpowiednimi modułami, na przykład Gallery2). Projekt powinien rozpocząć się od wybrania tematu projektu. Następnie należy opracować program projektu (założenia) wraz z przydziałem zakresu prac dla poszczególnych członków zespołu. Nie może zabraknąć harmonogramu działań niezbędnych do osiągnięcia sukcesu.

Przykładowe tematy witryn internetowych opartych na CMS Drupal:
- To dzięki matematyce i fizyce mamy komputery.
- Inżynierowie szukają inspiracji w świecie przyrody.
- Wykorzystanie światła w komputerze (monitory, wyświetlacze, napędy optyczne).

Wybór tematu projektu to początek procesu związanego z ustaleniem treści, które będą publikowane w witrynie Drupal. Następnie należy określić, za jaki zakres prac w projekcie jest odpowiedzialna każda osoba i w jakim czasie prace te muszą zostać wykonane.

Praktyczną część projektu należy rozpocząć od wykonania ćwiczenia 3.16. Dalsza praca jest uzależniona od tematu, który wymusi instalację dodatkowych modułów. Podczas gromadzenia materiałów do swojej publikacji należy bezwzględnie przestrzegać praw autorskich. Dlatego nie należy zapominać o zapisaniu informacji o tym, skąd pochodzą teksty, obrazy czy filmy.

Po upływie terminu realizacji projektu nastąpi jego prezentacja przez zespoły na forum klasy lub szkoły.



Na płycie CD znajdziesz plik *hosting CMS.pdf*, który zawiera informacje ułatwiające wykonanie projektu edukacyjnego.

3.5.8. Czym jest PHP?

Definicja

PHP to język skryptowy. Kod PHP osadzony na stronach HTML pozostaje niewidoczny dla użytkowników (w przeciwieństwie do na przykład skryptów JavaScript). Jest interpretowany przez serwer WWW, który na podstawie kodu tworzy stronę HTML lub inne wyniki. **Program napisany w PHP jest wykonywany po stronie serwera.**

Pierwotna wersja PHP powstała jako dzieło Rasmusa Lerdorfa w 1994 roku i służyła do monitorowania odwiedzin na stronie domowej autora. Obecna wersja to już poważny produkt o szerokich zastosowaniach. Początkowo skrót PHP oznaczał osobistą stronę domową (ang. *Personal Home Page*), został jednak zmieniony i oznacza *PHP HyperText Preprocessor*.

Główna strona PHP to *http://www.php.net*; funkcjonuje również wersja przetłumaczona na język polski: *http://php.net/manual/pl/index.php*.

Zalety PHP

- **Brak kosztów.** PHP jest bezpłatny i dostępny pod adresem: *http://www.php.net*.
- **Wysoka wydajność.** PHP jest w stanie obsłużyć na jednym serwerze wiele odwiedzin dziennie. Dokładne testy wydajności dostępne są na stronie *http://www.zend.com*.
- **Integracja z bazami danych.** PHP został wyposażony w interfejs do wielu różnych systemów baz danych, w tym do najpopularniejszej — MySQL, a także do PostgreSQL, mSQL, Oracle, DBM, filePro i innych.
- **Wbudowane biblioteki.** PHP udostępnia wiele wbudowanych funkcji do rozwiązywania różnych popularnych zadań w sieci WWW. Dzięki bibliotekom PHP bez problemu realizuje nawet takie zadania, jak tworzenie obrazków GIF, łączenie się z innymi usługami sieciowymi, wysyłanie poczty, praca z cookies, generowanie dokumentów PDF.
- **Przenośność.** PHP jest dostępny dla wielu systemów operacyjnych. Kod PHP możemy tworzyć zarówno w systemach Linux, Solaris, jak i w różnych wersjach Windows.

Wygląd skryptu PHP

Dokumenty, które zawierają skrypty PHP, składają się zazwyczaj z kodu HTML z wplecionym tekstem programu. Oznacza to, że tekst skryptu PHP umieszczamy bezpośrednio w kodzie HTML, oddzielając go specjalnymi znacznikami. Początek i koniec skryptu mogą być oznaczone na różne sposoby:

```
<? ?>
<?php ?> //Sposób używany w przykładach w podręczniku
<script language="php"> </script>
lub <% %> //Od wersji 3.0.4
```

Możesz dowolnie wybrać, którego znacznika używać. Warto jednak zachować konsekwencję w tym względzie.

Najprostszy skrypt będzie wyglądał następująco:

```
<html>
<head>
<meta http-equiv="Content-Type" content="tekst/html;
charset=iso-8859-2">
<title>Mój pierwszy skrypt PHP</title>
</head>
<body>
<?php
print("To mój pierwszy skrypt"); //To jest komentarz: Skrypt wypisze tekst, używając
                                 //funkcji print
?>
</body>
</html>
```

Zauważ, że za instrukcją `print` znajduje się średnik. Jest on bardzo ważny — jest wymagany po każdej instrukcji, w przeciwnym razie na ekranie zamiast oczekiwanych rezultatów wystąpi błąd.

Komentarze

Bardzo dobrym zwyczajem (nawykiem) jest stosowanie komentarzy w kodzie skryptów — nawet wówczas, gdy w trakcie pisania skryptu wydaje się to niepotrzebne. Komentarze mogą zaoszczędzić w przyszłości wiele cennego czasu. Możemy je zawrzeć wewnątrz znaczników na trzy różne sposoby:

```
/* Komentarz zawarty pomiędzy tymi znacznikami
może zajmować dowolną liczbę wierszy */
// Komentarz za tym znacznikiem może być zawarty tylko do końca tego wiersza
# Jak wyżej. Wszystko do końca tego wiersza jest komentarzem
```

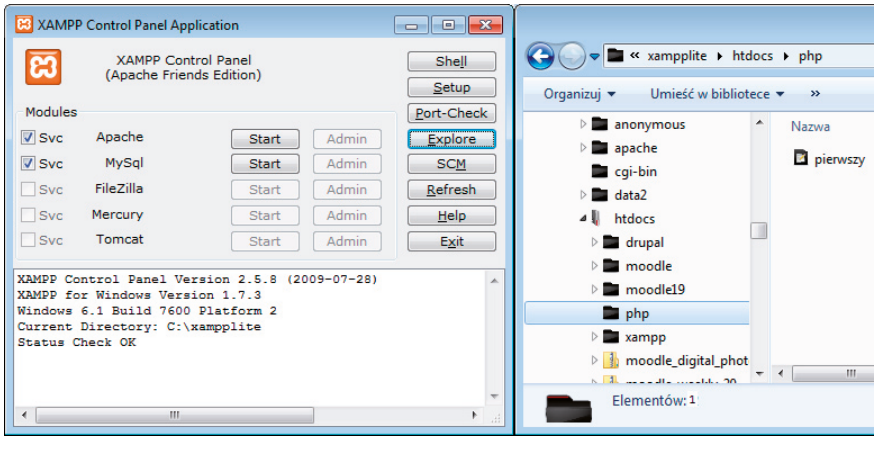
Oczywiście komentarz nie jest widoczny na stronie WWW i nie ma wpływu na wykonywanie programu. **Interpreter PHP** podczas wykonywania skryptu ignoruje wszystko, co zostało objęte znacznikami komentarza.

Rozszerzenie pliku

Pierwszy napisany skrypt, który wyświetli za pomocą wbudowanej funkcji `print` dowolny tekst (pamiętaj o średniku) opatrzony komentarzem, należy zapisać. Aby skrypt został wykonany, serwer WWW musi stwierdzić, że jest to właśnie skrypt PHP. W tym celu plik musimy zapisać z odpowiednim rozszerzeniem. Do wyboru mamy (w zależności od konfiguracji serwera): *.php*, *.php3*, *.php4*, *.php5* i *.phtml*. Utwórz folder *php* i zapisz w nim skrypt pod nazwą *pierwszy.php*. Można oczywiście użyć systemowego edytora plików tekstowych — Notatnika (przy zapisie plików koniecznie wybierz *Zapisz jako typ: Wszystkie pliki *.** oraz *Kodowanie UTF-8*), jednak lepiej użyć znanego już edytora kED.

Przed uruchomieniem skryptu *pierwszy.php* na swoim komputerze należy zainstalować i uruchomić środowisko WAMP. Przed instalacją systemu Drupal zainstalowany został serwer Apache z obsługą PHP — XAMPP Lite. Proces instalacji tego serwera jest przedstawiony na płycie CD dołączonej do podręcznika, w pliku *CMS Drupal.pdf*.

Jeżeli na Twoim lokalnym stanowisku komputerowym zostało zainstalowane i uruchomione środowisko WAMP (XAMPP Lite) z ćwiczenia 3.16, to masz możliwość wykonania skryptu PHP. Wystarczy do folderu *C:\xampplite\htdocs* skopiować folder *php* z plikami zawierającymi skrypty (rysunek 3.67).



Rysunek 3.67. Lokalizacja plików ze skryptami PHP, uruchamianych przez XAMPP Lite

Gdy pliki są już w odpowiednim folderze, wpisz w pasku adresu przeglądarki internetowej adres skryptu *http://127.0.0.1/php/pierwszy.php*. Działa?

Zmienne

Czym jest zmienna? Zmienna przechowuje dane w trakcie wykonywania skryptu. Każda zmienna ma nazwę i wartość, na przykład:

```
$imie="Jan";
$liczba1=5;
$liczba2=10.5;
```

Nazwy zmiennych mogą składać się z liter, cyfr i znaków podkreślenia. Nie mogą jednak rozpoczynać się od cyfry. Każdą nazwę zmiennej poprzedza się znakiem dolara — `$`. Oto prawidłowe nazwy zmiennych:

```
$imie; $Imie; $MojeImie; $imie4; $moje_imie;
```

Należy pamiętać, że w nazwach zmiennych jest uwzględniana wielkość liter. Oznacza to, że `$imie` i `$IMIE` to dwie różne zmienne, więc odwołując się do nich w programie, należy zwracać na to uwagę.

Typy zmiennych

W PHP nie jest wymagane wcześniejsze deklarowanie zmiennej, tj. wcześniejsze określanie typu zmiennej. Typ zmiennej jest automatycznie określany na podstawie wartości do niej przypisanej.

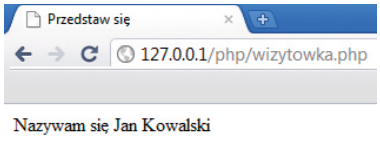
Zmienne w PHP dzielą się na typy, na przykład: łańcuchy znakowe (`string`), liczby całkowite (`integer`), liczby zmiennoprzecinkowe (`double`).

```
$imie="Jan"; //Zmienna typu string
$liczba1=5; //Zmienna typu integer
$liczba2=10.5; //Zmienna typu double
```

Czas zatem na prosty skrypt wykorzystujący zmienne:

```
<html>
<head>
<meta http-equiv="Content-Type" content="tekst/html;
charset=iso-8859-2">
<title>Przedstaw się</title>
</head>
<body>
<?php
$imie="Jan"; //Imię zostało zapamiętane w zmiennej imie
$nazwisko="Kowalski"; //Nazwisko zostało zapamiętane w zmiennej nazwisko
print("Nazywam się ".$imie." ".$nazwisko); //Kilka słów wyjaśnienia poniżej
?>
</body>
</html>
```

Zapisz ten skrypt pod nazwą *wizytowka.php*. Po wykonaniu skryptu i przesłaniu go do przeglądarki na ekranie pojawi się tekst: *Nazywam się Jan Kowalski* (rysunek 3.68).



Rysunek 3.68. Widok wyniku wykonania skryptu w oknie przeglądarki

Zwróć uwagę na pasek tytułowy przeglądarki i tytuł karty.

Oczywiście ten sam efekt można uzyskać bez użycia zmiennych — wystarczy napisać: `print("Nazywam się Jan Kowalski");`. Jednak z użycia zmiennych płyną inne korzyści — można zmienić ich wartości.

Przypisywanie wartości do zmiennych

Aby określić wartość danej zmiennej, używamy znaku równości — operatora przypisania.

```
$imie="Jan";
```

Wartość zmiennej typu `string`, czyli dowolny ciąg znaków, zapisujemy w cudzysłowach:

```
$napis="To jest zmienna typu string";
$imie="Jan";
$nazwisko="Kowalski";
$tekst="Nazywam się $imie $nazwisko";
```

Zmienna `$tekst` przyjmuje teraz wartość: `Nazywam się Jan Kowalski`.

Przyjrzyj się jeszcze wykorzystaniu funkcji `print`:
`print("Nazywam się ".$imię." ".$nazwisko);`.

W celu połączenia ze sobą łańcuchów został tu użyty znak kropki, zwany operatorem konkatenacji łańcuchów. Teksty można połączyć w następujący sposób:

`$tekst=$imie.$nazwisko;` oraz `$tekst="$imie $nazwisko";`

Wartości zmiennych typu `integer` (liczba całkowita) oraz `double` (liczba zmiennoprzecinkowa) zapisujemy bez cudzysłowów:

```
$liczba=21;
$liczba_dni=365;
$liczba_pi=3.14;
```

Na zmiennych obu typów można przeprowadzać działania matematyczne za pomocą operatorów: +, -, *, /.

Operatory

Operator informuje PHP, że ma wykonać jakieś matematyczne lub logiczne działanie. Większość operatorów w PHP jest całkowicie intuicyjna.

Operatory matematyczne

Operator	Przeprowadzana operacja
`+`	Dodawanie `$a=$b+3;`
`-`	Odejmowanie `$a=$b-3;`
`*`	Mnożenie `$a=$b*3;`
`/`	Dzielenie `$a=$b/3;`
`%`	Dzielenie modulo (zwraca liczbę całkowitą stanowiącą resztę z dzielenia)
`++`	Inkrementacja `$a++` (dodaje do zmiennej wartość `1`)
`--`	Dekrementacja `$a--` (odejmuje od zmiennej wartość `1`)

Operatory można stosować zarówno do liczb całkowitych, jak i zmiennoprzecinkowych.

Przykład 3.5.

Napisz prosty skrypt obliczający pole prostokąta. Wypisz na ekranie wartości boków oraz wynik obliczeń. Skrypt zapisz pod nazwą *prostokat.php*.

W zmiennych zapisz boki prostokąta, a następnie za pomocą operatora mnożenia oblicz jego pole.

```
<html>
<head>
<meta http-equiv="Content-Type" content="tekst/html;
charset=iso-8859-2">
<title>Pole prostokąta</title>
</head>
<body>
<?php
$a=15;
$b=4;
$P=$a*$b; //Wprowadzona nowa zmienna P
print("Pole prostokąta o bokach $a i $b wynosi: $P");
?>
</body>
</html>
```

Operatory logiczne oraz operatory porównania

Operatory porównania, jak sama nazwa wskazuje, porównują wartości (na przykład zmiennych) i zwracają wartość `TRUE` (prawda) lub `FALSE` (fałsz).

Operator	Przeprowadzane porównanie
`<`	Mniejsze `$a<$b`
`>`	Większe `$a>$b`
`<=`	Mniejsze lub równe `$a<=$b`
`>=`	Większe lub równe `$a>=$b`
`==`	Równe `$a==$b`
`===`	Identyczne `$a === $b` — prawda, jeśli `$a` jest równe `$b` i są tego samego typu (tylko PHP4)
`!=`	Różne `$a!=$b`
`&&` (AND)	Koniunkcja (logiczne i) `$a==1 && $b==2` (nie potrzeba nawiasów)
`||` (OR)	Alternatywa (logiczne lub) `$a==2 || $b==1`
`!` (NOT)	Negacja `!($a==1)` (negacja powoduje zmianę wartości na przeciwną, na przykład prawda na fałsz — w tym przykładzie: jeśli `$a` jest równa `1`, wartość `TRUE` zostanie zamieniona na `FALSE`)

Należy tu zwrócić uwagę na operator przypisania (=) oraz operator równości (==) i nie mylić ich.

Operator przypisania

Można spotkać się z różnymi zapisami poprawnych instrukcji przypisania:

```
$a=10; //Zmiennej $a została przypisana wartość z prawej strony, czyli 10
$a+=5; //Do zmiennej $a=10 dodano 5, czyli zmienna $a=15. Ten zapis jest tożsamy z poniższym:
$a=$a+5;
```

Na szczególną uwagę zasługuje operator *konkatenacji* (łączenia łańcuchów znaków), czyli kropka (.).

```
<?php
$napis="Wlazł";
$napis.=" kotek";
$napis.=" na";
$napis.=" płotek.";
print($napis);
?>
```

Sprawdź efekt działania operatora konkatenacji.

Stałe

Stałe definiujemy za pomocą dwuargumentowej funkcji `define("nazwa_stałej", wartość)`, na przykład:

```
define("PROCENT", 0.19);
define("NAPIS", "Warto znać PHP!"); //Wartością stałej może być tekst
```

Nazwa stałej została zapisana wielkimi literami. Nie jest to konieczne, ale — dla odróżnienia od zmiennych — powszechnie przyjęte.

Przykład 3.6.

Oblicz kwotę podatku od dochodów. Stawka podatkowa wynosi 19%. Niech będzie to nasza stała.

```
<?php
define("PODATEK", 0.19); //Stała o wartości 0.19 (19%)
print("Podatek od kwoty: 2800 zł wynosi: ". (PODATEK*2800)." zł");
?>
```

Zapisz ten przykład z wykorzystaniem zmiennej, która przechowuje wartość dochodu.

Oczywiście większy sens definiowania stałej istnieje wówczas, gdy kilkakrotnie wykorzystujemy jej wartość. Gdy zmieni się stawka podatkowa, będziemy musieli uczynić to tylko raz (w miejscu definiowania stałej), a nie w każdym miejscu wywołania zmiennej.

Stałej **TRUE** przypisana jest wartość `1`, zaś stałej **FALSE** — wartość `0`.

Zmienne środowiskowe

Zmienne te zawierają specjalne informacje dotyczące konfiguracji serwera WWW czy konfiguracji PHP. Wykorzystując je, możemy się dowiedzieć czegoś więcej na temat osób odwiedzających stronę czy też na temat samego serwera. Listę zmiennych środowiskowych możemy uzyskać, wywołując funkcję *phpinfo()*.

Przykład 3.7.

W tym przykładzie wykorzystamy zmienną środowiskową `$HTTP_USER_AGENT` w celu uzyskania nazwy przeglądarki oraz systemu operacyjnego odwiedzającego. Za pomocą zmiennej `$REMOTE_ADDR` uzyskamy adres IP.

```
<?php
print("Używasz przeglądarki: ". $HTTP_USER_AGENT. "<br />"); //Zwróć uwagę na znacznik <br />
print("Twój adres IP: ".$REMOTE_ADDR."<br />");
?>
```

Oto efekt:

```
Używasz przeglądarki: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
Twój adres IP: 127.0.0.1
```

Zmienne środowiskowe są nieocenione przy konstruowaniu statystyk strony dostarczających niezbędnych informacji o gościach odwiedzających witrynę.

Instrukcja warunkowa if

Często będziemy chcieli wykonać jakieś działanie, ale tylko wtedy, gdy zostanie spełniony pewien warunek. W tym celu posłużymy się instrukcją `if`. Ma ona następującą postać:

```
if (warunek1) {blok instrukcji 1}
elseif (warunek2) {blok instrukcji 2}
elseif (warunek3) {blok instrukcji 3}
else {blok instrukcji_else}
```

i nieco krócej z jednym warunkiem:

```
if (warunek) {
 instrukcja_true;
} else {
 instrukcja_false;
}
```

lub forma uproszczona:

```
if (warunek) {
 blok instrukcji
}
```

Spójrz, jak to działa w praktyce. Sprawdź, jaką liczbę przechowuje zmienna `$liczba`:

```
if($liczba>0) {
 print ("$liczba jest większa od 0");
}
```

```
elseif ($liczba<0) {
 print ("$liczba jest mniejsza od 0");
}
else {
 print ("$liczba jest równa 0");
}
```

Klauzule `elseif` oraz `else` stosujemy w celu sprawdzenia wszystkich możliwych sytuacji. Zaczynamy od zapisania pierwszego warunku w instrukcji `if`, następnie sprawdzamy wszystkie pozostałe warunki, umieszczając każdy z nich w klauzuli `elseif`. Na końcu umieszczamy klauzulę `else` i w ten sposób jesteśmy przygotowani na wszystkie możliwe sytuacje.

Przykład 3.8.

Napisz skrypt, który będzie wyświetlał nazwę dnia tygodnia w języku polskim. Posłuż się, oprócz instrukcji `if`, funkcją `date()`, która będzie zwracała bieżący dzień tygodnia (w języku angielskim).

Funkcja `time()` zwraca w formie liczby `integer` aktualny znacznik czasu. *Znacznik czasu* określa liczbę sekund, jakie upłynęły od 1 stycznia 1970 roku.

Funkcja `date(format, znacznik_czasu)` zwraca łańcuch znaków opisujący datę. Ma ona dwa argumenty. Jeden z nich — `format` — informuje funkcję `date()` o postaci, w jakiej ma być wyświetlona data. Format `"l"` oznacza dzień tygodnia. Jeśli znacznik czasu nie zostanie podany, funkcja `date()` użyje aktualnej daty.

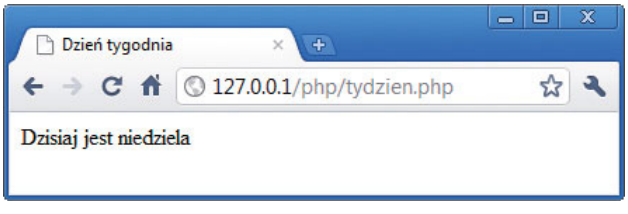
```
<?php
$dzien=date("l"); //Zmiennej $dzien przypisz aktualny dzień tygodnia
if ($dzien=="Monday") {print ("Jest poniedziałek");}
elseif ($dzien=="Tuesday") {print ("Jest wtorek");}
elseif ($dzien=="Wednesday") {print ("Jest środa");}
elseif ($dzien=="Thursday") {print ("Jest czwartek");}
elseif ($dzien=="Friday") {print ("Jest piątek");}
elseif ($dzien=="Saturday") {print ("Jest sobota");}
else {print ("Dzisiaj jest niedziela");}
?>
```

Zapis tego skryptu jest czytelny i nie wymaga wyjaśnień. Rzadko jednak pisze się skrypt w powyższy sposób. Częściej stosuje się taki zapis:

```
<?php
$dzien=date("l"); //Zmiennej $dzien przypisz aktualny dzień tygodnia
if ($dzien=="Monday") {
 print ("Jest poniedziałek");
```

```
} elseif ($dzien=="Tuesday") {
 print ("Jest wtorek");
} elseif ($dzien=="Wednesday") {
 print ("Jest środa");
} elseif ($dzien=="Thursday") {
 print ("Jest czwartek");
} elseif ($dzien=="Friday") {
 print ("Jest piątek");
} elseif ($dzien=="Saturday") {
 print ("Jest sobota");
} else {
 print ("Dzisiaj jest niedziela");
}
?>
```

Zapisz skrypt pod nazwą *tydzien.php* i sprawdź jego działanie. Czy efekt jest podobny do tego na rysunku 3.69?



Rysunek 3.69. Skrypt wyświetla aktualny dzień tygodnia

Pętla for

Często będziemy stawać przed zadaniem, aby jakąś czynność powtórzyć wiele razy. Wyobraźmy sobie na przykład, że chcemy wypisać liczby w kolejności od 1 do 100. Oczywiście moglibyśmy to uczynić, stosując choćby funkcję `print()`. Można to jednak zrobić o wiele prościej i szybciej — z wykorzystaniem pętli. Polega ona na wielokrotnym wykonywaniu bloku instrukcji. W PHP mamy kilka pętli. Najprostszą z nich jest pętla `for`. Oto jej postać:

```
for (inicjalizacja; warunek_pętli; zwiększanie_licznika) {
 blok instrukcji
}
```

`inicjalizacja` — tu nadajemy zmiennej sterującej w pętli wartość początkową.

`warunek_pętli` — jest to wyrażenie określające warunek wykonywania pętli.

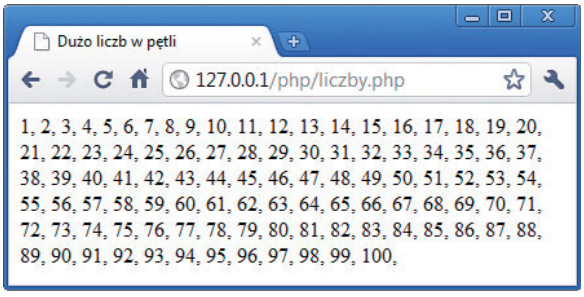
`zwiększanie_licznika` — tu określamy sposób zwiększania zmiennej sterującej przy każdym przebiegu pętli.

Przykład 3.9.

Wypisz kolejne liczby od 1 do 100:

```
<?php
for($i=1; $i<=100; $i++) {
 print("$i, "); //Zwróć uwagę na przecinek i spację oddzielające liczby
}
?>
```

Zapisz skrypt w pliku *liczby.php* i sprawdź jego działanie (rysunek 3.70).



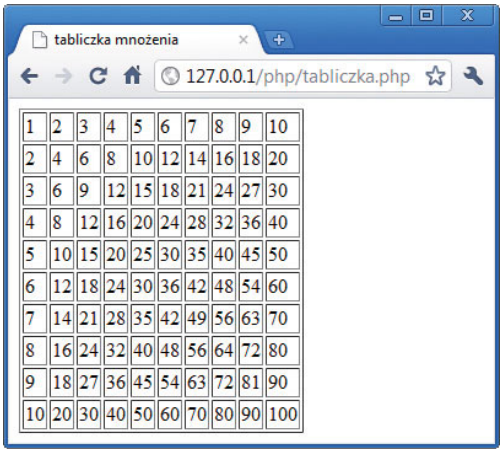
Rysunek 3.70. Zastosowanie pętli for do wypisania wielu liczb

Przykład 3.10.

Utwórz w tabeli tabliczkę mnożenia.

```
<?php
print("<table border>"); //Znacznik tabeli z ramką
for($i=1; $i<=10; $i++) {
 print("<tr>"); //Utwórz nowy wiersz tabeli
 for($j=1;$j<=10;$j++){ //Pętla w pętli tworzy komórki w wierszu
  print("<td>".$i*$j."</td>");
 }
 print("</tr>");
}
print("</table>");
?>
```

Skrypt zapisz pod nazwą *tabliczka.php* i sprawdź jego działanie (rysunek 3.71). Ile czasu i energii pochłonęłoby utworzenie takiej tabliczki w tradycyjny sposób, za pomocą wyłącznie znaczników HTML?



Rysunek 3.71. Tabliczka mnożenia — zastosowanie pętli for

Zauważ, że wartości mnożonych liczb możesz dowolnie zmieniać. Również wymiar tabeli może się zmienić. Dokonaj zmian w skrypcie, aby mnożone były liczby od 1 do 20 i od 100 do 110. Zmień również wielkość ramki oraz kolor tła tabeli.

Niestety, wadą pętli `for` jest to, że musi być określona skończona liczba jej powtórzeń. A co w przypadku, gdy nie wiemy, ile razy należy powtórzyć jakąś czynność w celu otrzymania oczekiwanego skutku? Tu przychodzi z pomocą pętla `while`.

Pętla while

Dopóki jest spełniony warunek, wykonuje instrukcje:

```
while (warunek) {
 blok instrukcji
}
```

Zastosujmy zatem pętlę `while` do wypisania 100 liczb z przykładu 3.9.

```
<?php
$i=1; //Przypisz wartość początkową zmiennej $i przed wejściem do pętli
while ($i<=100) { //Sprawdź warunek, a gdy jest prawdziwy, wykonaj instrukcję w pętli
 print("$i, ");
 $i++; //Zwiększ zmienną $i o 1
 } //Tu kończy się pętla
?>
```

Zapisz skrypt pod nazwą *liczby_while.php* i sprawdź jego działanie.

pętla do...while

Wykonuje instrukcje, dopóki jest spełniony warunek. Różnica pomiędzy pętlą `while` a pętlą `do...while` tkwi w miejscu sprawdzania warunku:

```
do{
 blok instrukcji
}
while (warunek)
```

Napisz skrypt wypisujący liczby od 1 do 100, wykorzystując pętlę `do...while`.

Tyle w ramach wstępu do tworzenia dynamicznych stron WWW. Masz już podstawową wiedzę dotyczącą tworzenia skryptów PHP. Na oficjalnej polskiej stronie PHP znajdziesz podręcznik i przewodnik po PHP: *http://www.php.net/manual/pl/langref.php*. Możesz również skorzystać z wielu różnych kursów internetowych na ten temat. Źródłem wiedzy są także fora dyskusyjne, na których uzyskasz pomoc. Polecam też fachową literaturę.

3.5.9. Zadania

Na płycie CD dołączonej do podręcznika znajduje się folder *Literatura* zawierający witrynę WWW — wyglądem jej stron zarządza arkusz stylów *styl_literatury.css*, natomiast sposób nawigacji reguluje napisana w języku HTML definicja ramek (*frames*), zawarta na stronie *index.html*. Funkcję strony głównej pełni *index.html*. Skopiuj ten folder na lokalny dysk.

Zadanie 3.16. Wybierz dowolną stronę, której nazwa jest tytułem utworu (tj. inną niż *index.html*, *menu.html*), i pozbaw ją stylu, usuwając z nagłówka całą linię wywołującą arkusz:

```
<link rel="stylesheet" href="styl_literatury.css" type="text/css" />
```

lub usuwając tylko samo odwołanie do arkusza *styl_literatury.css*. Zobacz efekt (odśwież okno przeglądarki). Po wykonaniu zadania przywróć stan poprzedni.

Zadanie 3.17. Oprócz stylu zdefiniowanego w pliku *styl_literatury.css*, w folderze *Literatura* znajduje się jeszcze kilka innych arkuszy stylów *.css*. Skopiuj kolejno zawartość innych arkuszy w miejsce dotychczasowej zawartości pliku *styl_literatury.css*, by uzyskać nowy wygląd witryny.

Zadanie 3.18. Przeanalizuj (najlepiej bezpośrednio na każdej stronie z osobna, bez użycia strony *index.html*) działanie stylów i udziel odpowiedzi na poniższe pytania.

1. Czy tekst jest wycentrowany w obrębie okna przeglądarki czy w obrębie bloku, czy też jest złożony według jeszcze innego szablonu?
2. Jak nazywa się lub nazywają się klasy stylu (`class`) odpowiedzialne za sposób wyświetlania wierszy?
3. Czy sposób składu wiersza zależy zawsze i wszędzie tylko od odpowiednich klas?

Zadanie 3.19. Zmień w arkuszu *styl_dla_menu.css* następujące elementy:

- kolor tła (*background-color*) na zielony;
- obecne wyrównanie tekstu na centrowanie (`text-align` w kilku różnych miejscach arkusza);
- wielkość i kolor czcionki (`font-size` i `color` występują w kilku miejscach).

Aby obejrzeć efekty, należy w nagłówku strony *menu.html* zmienić odwołanie do arkusza stylu ze *styl_literatury.css* na *styl_dla_menu.css*.

Wskazówka

Nie każda zmiana jest realizowana i widoczna w oknie przeglądarki. Przykładowo styl koloru `body {color: #bbeeff;}` (zmień ten kolor i zobacz efekt), zdefiniowany dla całej sekcji `body`, nie będzie obowiązywał wewnątrz znacznika `<a class="spistresci">` stylizowanego zgodnie z klasą `.spistresci` opisaną w arkuszu następująco: `.spistresci {color: #bdf7b3;}`. Właściwości nieokreślone w klasie będą dziedziczone z zewnątrz, na przykład kolor tła (`background-color`), który jest zdefiniowany w arkuszu przy znaczniku `body`, odziedziczy klasa `.spistresci`, gdyż nie zawiera deklaracji koloru tła.

Zadanie 3.20. Napisz skrypt PHP wykonujący na dwóch liczbach cztery podstawowe działania arytmetyczne. Skrypt zapisz pod nazwą *kalkulator.php*.

Zadanie 3.21. Oto kod skryptu wypisującego dni tygodnia w języku angielskim od dziś do niedzieli. Przeanalizuj ten skrypt.

```
<?php
$dzisiaj=time();
print ("Do niedzieli pozostały jeszcze dni: <br />");
while (date("l", $dzisiaj) !="Sunday") {
 print (date("l", $dzisiaj));
 $dzisiaj += (24*3600);
 print ("<br />");
}
?>
```

W zmiennej `$dzisiaj` przechowujemy aktualną datę określaną za pomocą liczby sekund, które upłynęły od 1 stycznia 1970 roku (wykorzystujemy tu funkcję `time()`). Następnie za pomocą funkcji `print()` wyświetlamy tekst, który opisze uzyskane wyniki. Pętlą `while` sprawdzamy, czy data zapisana w zmiennej `$dzisiaj` nie jest niedzielą. Jeśli warunek jest prawdziwy, to wyświetlamy komunikat z dniem tygodnia, a do wartości zapisanej w zmiennej `$dzisiaj` dodajemy 24 godziny w sekundach, czyli 24*3600 sekund. Czynności te powtarzamy, aż data zapisana w zmiennej `$dzisiaj` będzie niedzielą. Wtedy pętla `while` zostanie zakończona.

Napisz skrypty, które korzystając z funkcji daty i czasu, wypiszą kilka przydatnych informacji:

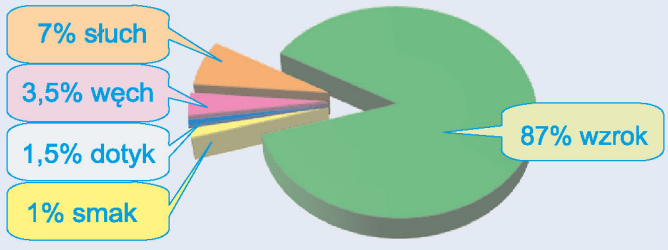
- Ile dni pozostało do matury?
- Ile dni chodzisz w tym roku szkolnym do szkoły?
- Który mamy dzień roku?
- Ile dni minęło od dnia Twoich urodzin?
- W którym dniu tygodnia się urodziłeś?

Rozdział 4.

MULTIMEDIA I GRAFIKA KOMPUTEROWA

Na początku dziejów współczesnej techniki komputerowej istniał jedynie alfanumeryczny (tekstowy) tryb pracy. Starsi użytkownicy komputerów z pewnością pamiętają rysunki wykonane wyłącznie ze znaków ASCII. Z kolei młodsi pamiętają czasy telefonów komórkowych, które miały możliwość wysyłania jedynie krótkich wiadomości tekstowych SMS. Na maleńkim wyświetlaczu były rysowane misie i serduszka tworzone ze znaków wprowadzanych z klawiatury telefonu. Tekstowe rysunki nierzadko zadziwiały pomysłowością i ogromem pracy włożonej w ich wykonanie.

Blisko 90% informacji, które docierają do odbiorcy, jest odbieranych przez zmysł wzroku. Kolejne 7% rejestruje słuch. Zaledwie kilka procent przypada na pozostałe zmysły: dotyku, smaku i powonienia. Jak ważna jest zatem informacja, która dociera do nas poprzez obraz, napis i dźwięk? Dla lepszego zobrazowania procentowego udziału poszczególnych zmysłów w odbieraniu informacji zestawiono te dane na wykresie zamieszczonym na rysunku 4.1. Widać na nim, jak ogromny wpływ ma na nas obraz.



Rysunek 4.1. Procentowy udział zmysłów w postrzeganiu świata przez człowieka według Jana Młodkowskiego (*Aktywność wizualna człowieka*, PWN 1998)

4.1. Grafika komputerowa

Czym zajmuje się ta dziedzina informatyki? Najogólniej można powiedzieć, że generowaniem, przetwarzaniem i prezentowaniem obrazu za pomocą komputera.

Grafikę komputerową dzielimy na dwie grupy: grafikę rastrową i grafikę wektorową. Te dwie odmiany stanowią podstawę wszystkich innych rodzajów grafiki komputerowej, począwszy od składu publikacji i tworzenia stron WWW, a skończywszy na montażu filmów, projektowaniu mebli czy obrazowaniu badania USG jamy brzusznej.

Ze względu na rodzaj sceny grafikę komputerową dzielimy na płaską (2D) i trójwymiarową, czyli przestrzenną (3D).

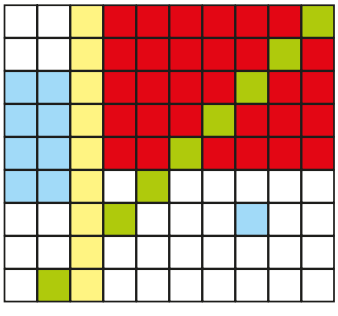
Grafika komputerowa znalazła zastosowanie w tak wielu dziedzinach, że nie można już sobie wyobrazić powrotu do komputerów jedynie z trybem tekstowym. Od graficznego interfejsu użytkownika (GUI), poprzez graficzną prezentację danych na wykresach, skład i montaż komputerowy dokumentów i filmów, prezentacje multimedialne, internet, aż po projekty inżynierskie, kartografię i medycynę — wszędzie mamy do czynienia z grafiką komputerową.

4.1.1. Grafika rastrowa

W grafice komputerowej stosuje się dwa sposoby reprezentacji obrazu. Jeden opisuje kolory obrazka punkt po punkcie (piksel po pikselu). Tak reprezentowaną grafikę nazywamy rastrową lub bitmapową. Drugi sposób odwołuje się do zasad matematycznych — każdy element obrazka opisany jest formułą matematyczną. Grafikę zapisaną w tej postaci nazywamy grafiką wektorową.

Przedstawiony na rysunku 4.2 maleńki fragment grafiki bitmapowej, pokazanej w powiększeniu 1600%, w umowny sposób można opisać następująco: na początku rozmiar bitmapy w poziomie i w pionie, a następnie informacja o kolorze każdego punktu obrazu. Można to zapisać w taki sposób:

```
10, 9;
B, B, Ż, C, C, C, C, C, C, Z;
B, B, Ż, C, C, C, C, C, Z, C;
N, N, Ż, C, C, C, C, Z, C, C;
N, N, Ż, C, C, C, Z, C, C, C;
N, N, Ż, C, C, Z, C, C, C, C;
N, N, Ż, B, Z, B, B, B, B, B;
B, B, Ż, Z, B, B, B, N, B, B;
B, B, Ż, B, B, B, B, B, B, B;
B, Z, Ż, B, B, B, B, B, B, B.
```



Rysunek 4.2. Przykładowa grafika rastrowa (bitmapa)

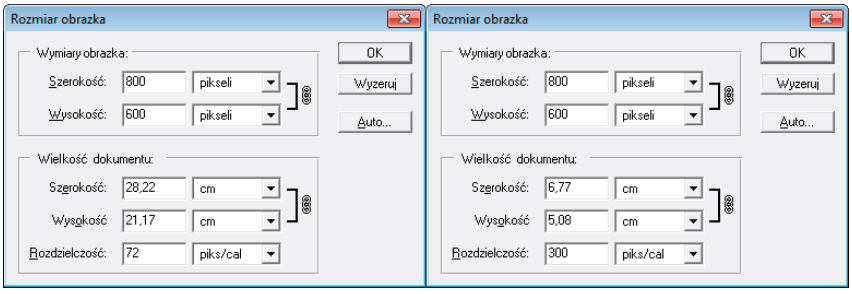
Gdzie: B — kolor biały, Ż — żółty, Z — zielony, C — czerwony, N — niebieski.

4.1.2. Rozdzielczość a wymiary obrazka

Rozdzielczość ma obecnie wiele znaczeń, często nieprawidłowo używanych i mylonych z wymiarami lub rozmiarem. *Rozdzielczością obrazka* (rysunku) jest gęstość upakowania pikseli na jednostce długości (cal, rzadziej cm), natomiast liczba pikseli w poziomie i w pionie to jego *wymiary*. Rozróżnienie to zilustrowano na rysunku 4.3. Zwróć uwagę na wymiary obrazka: w pikselach w obu przypadkach są jednakowe i wynoszą 800 na 600 pikseli (px), jednak wielkość dokumentu (w centymetrach) jest różna. Zależy ona bowiem od dwóch parametrów: rozdzielczości oraz wymiarów obrazka — dokument jest mniejszy dla większej rozdzielczości (300 ppi) i większy dla mniejszej rozdzielczości (72 ppi). Dane te stanowią informację o możliwościach wykorzystania obrazka.

Rozdzielczość obrazka możemy zatem obliczyć jako stosunek wymiarów obrazka w pikselach do wymiarów obrazka w calach (czasami w centymetrach):

$$\text{rozdzielczość} = \frac{\text{wymiary obrazka w pikselach}}{\text{wymiary obrazka w calach}}$$



Rysunek 4.3. Zależność wielkości obrazka od jego rozdzielczości

Rozdzielczość obrazka najczęściej wyrażamy w jednostkach przeliczanych na 1 cal:

- *ppi* (ang. *pixel per inch*), czyli w liczbie pikseli na cal.

Dla lepszego zrozumienia pojęcia rozdzielczości obrazka posłużymy się następującym przykładem. Masz dwa arkusze papieru w kratkę. Kartki są jednakowej wielkości, na przykład A4, jednak jeden arkusz to papier milimetrowy, drugi zaś to papier w zeszytową kratkę. Kratka, jak piksel, jest niepodzielna i może zawierać tylko jeden kolor. Narysuj kolorowy obrazek, wypełniając kolorem pojedyncze kwadraty (piksele) papieru. Na papierze milimetrowym jest większa rozdzielczość, a powstały obraz ma więcej szczegółów. Na kartce w kratkę jest mniejsza rozdzielczość i mniej szczegółów rysunku.

Gdyby powstałe rysunki zapisać na dysku, to pliki by miały różne rozmiary. Termin *rozmiar* zarezerwowany jest dla zapisanego na dysku pliku i podawany jest w kB, MB, GB.

Dwa obrazki, które pokazano na rysunku 4.3, są wyświetlane na ekranie monitora w takich samych wymiarach. Masz wrażenie, że ustawienie różnych rozdzielczości nie ma żadnego znaczenia — na ekranie widzisz bowiem dwa jednakowej wielkości obrazki. Jest to związane z rozdzielczością monitora, której typowa wartość mieści się w zakresie od 72 ppi do około 96 ppi. Rozdzielczość monitora jest maksymalną *rozdzielczością wyświetlania obrazu*.

W fotografii cyfrowej możesz się spotkać z powszechnie używanym terminem *rozdzielczość zdjęcia*. Pojęcie to właściwie odnosi się do wymiarów fotografii. Na przykład zdjęcie wykonane w wymiarach 3000 × 2000 pikseli ma rozdzielczość 6 megapikseli (*Mpx, Mpix*).

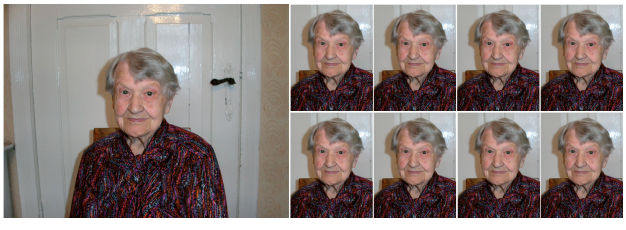
Na rysunku 4.4 zamieszczono dwa zdjęcia wykonane aparatem cyfrowym. Zdjęcia są identyczne — zawierają te same szczegóły. Jednak jeśli powiększymy zaznaczony fragment, mogą wystąpić dwa przypadki: albo uzyskamy widok wydobytych szczegółów, których wcześniej nie było widać, albo zobaczymy powiększone piksele. Fotografie cyfrowe należy więc wykonywać z myślą o ich przeznaczeniu i ewentualnej dalszej obróbce w edytorze graficznym.



Rysunek 4.4. Powiększenia fragmentów fotografii wykonanych w różnej rozdzielczości (zdjęcie lewe -1 Mpx, prawe -10 Mpx ppi)

Obrazek i zdjęcie można wydrukować. I tu pojawia się kolejna rozdzielczość: *rozdzielczość wydruku*, która podawana jest w jednostce *dpi* (ang. *dot per inch*) i wyrażana w kropkach (punktach, plamkach) na cal. Odnosi się do rozdzielczości urządzeń drukujących lub naświetlających i ma wpływ na rozmiar drukowanego obrazka lub fotografii. Jeśli rozdzielczość wydruku obrazka o wymiarach 800 × 600 pikseli zostanie ustalona na 1 dpi, obrazek będzie wydrukowany w rozmiarze 800 × 600 cali, czyli około 20 × 15 metrów. Jeśli rozdzielczość wydruku wyniesie 100 dpi, wielkość wydruku będzie wynosiła 8 × 2 cale, natomiast przy rozdzielczości 200 dpi — 4 × 3 cale. Rozdzielczość wydruku ma decydujący wpływ na jakość wydruku.

Ćwiczenie 4.1. Na płycie CD dołączonej do podręcznika zamieszczono amatorskie zdjęcie portretowe *babcia_Zosia.jpg*, z którego należy przygotować zdjęcia legitymacyjne o wymiarach 3,5×4,5 cm. Zdjęcia należy zamieścić na jednym obrazie, z którego ma być wykonana odbitka fotograficzna o wymiarach 15×10 cm. Plik ze zdjęciami należy tak przygotować, aby można było oddać go do laboratorium fotograficznego w celu wykonania odbitki. Rozdzielczość papieru fotograficznego jest bardzo duża, a powstały obraz ma więcej szczegółów, niż może dostrzec ludzkie oko. Aby odbitka fotograficzna była ostra, ustalmy rozdzielczość obrazu na wartość 300 ppi. Obydwa pliki, źródłowy i wynikowy, pokazano na rysunku 4.5.



Rysunek 4.5. Zdjęcia legitymacyjne i plik źródłowy

Do wykonania polecenia z ćwiczenia 4.1 wykorzystamy edytor obrazów GIMP. Wersję 2.6.10 tego programu zamieszczono na płycie CD dołączonej do podręcznika, w pliku *gimp-2.6.10-i686-setup-1.exe*. Opis wykonania ćwiczenia ze zrzutami ekranowymi znajduje się także na płycie CD, w pliku *zdjecia_legitymacyjne.pdf*.

Ciekawostka

Edytor obrazów GIMP jest oprogramowaniem typu open source. Można go rozprowadzać dalej i (lub) modyfikować na warunkach Powszechnej Licencji Publicznej GNU, wydanej przez Fundację Wolnego Oprogramowania.

W ćwiczeniu 4.1 ograniczyliśmy się jedynie do kadrowania obrazu bez wcześniejszego poprawiania wykonanego zdjęcia. Spróbuj, wykorzystując własną fotografię, przed wykadrowaniem obrazu wykonać korektę zdjęcia. Można do tego celu użyć narzędzi takich jak: klonowanie, wyostrzanie, rozmycie, balans kolorów, łatka. Dowiedz się również, jaka powinna być rozdzielczość przygotowanego obrazu do Twojego laboratorium fotograficznego. Może się okazać, że rozdzielczość 300 ppi to za mało do otrzymania odbitki bardzo dobrej jakości i należy użyć 350 lub nawet 420 ppi.



Rysunek 4.6. Przekształcenie obrazu z efektem 3D (z lewej: oryginalne zdjęcie, z prawej: po transformacji)

Ćwiczenie 4.2. Na płycie CD dołączonej do podręcznika zamieszczono zdjęcie *kroton.jpg*, które należy za pomocą odpowiednich transformacji przedstawić tak, jakby to była fotografia 3D. Przykładową transformację zdjęcia rośliny pokazano na rysunku 4.6.

Szczegółowy opis rozwiązania, wykonanego w edytorze graficznym GIMP, zamieszczono na płycie CD, w pliku *transformacje.pdf*. W ćwiczeniu tym wykorzystano następujące czynności edycyjne: zaznaczanie obrazu, kopiowanie, tworzenie nowej warstwy, zmianę rozmiaru obrazu, pobieranie koloru z obrazu, skalowanie, zniekształcanie i kadrowanie obrazu.

Zmiany rozmiarów obrazu, czyli skalowanie, oraz wszelkie przekształcenia i transformacje (pochylenie, zniekształcenie) pociągają za sobą pewne konsekwencje:

- zmniejszenie lub zwiększenie zapotrzebowania na pamięć,
- wolniejsze przetwarzanie obrazu,
- utratę informacji o szczegółach,
- zmianę tempa przetwarzania obrazu,
- konieczność stosowania antyaliasingu.

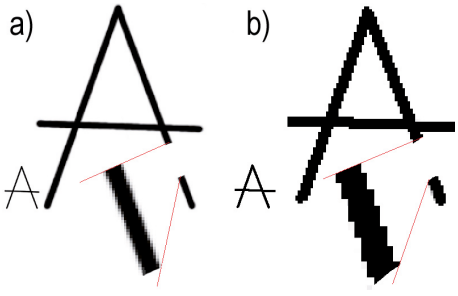
Definicja

Aliasing — stopnie (schodki) widoczne na ukośnych lub krzywych liniach lub krawędziach obiektów spowodowane przez ostry kontrast barw. Możemy usunąć ten efekt, stosując *antyaliasing*.

Definicja

Antyaliasing — technika polegająca na redukcji efektu schodkowania linii spowodowanego aliasingiem.

Obecnie karty graficzne oferują już sprzętowy antyaliasing. Większość programów graficznych nie ma problemu ze „schodkowymi" krzywymi. Dla pikseli wchodzących w skład postrzępionych schodkowych krawędzi wyliczana jest wartość pośrednia pomiędzy sąsiadującymi ze sobą kolorami. Pikselami w tych pośrednich barwach wypełniana jest schodkowa krawędź, przez co powstaje łagodne przejście zamiast ostrej linii. Niestety, obraz poddany antyaliasingowi traci na ostrości. Na rysunku 4.7 został zilustrowany ten problem.

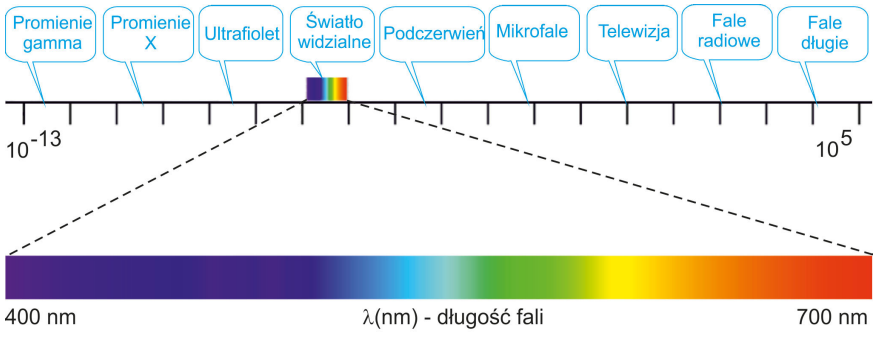


Rysunek 4.7. Powiększona bitmapa: a) z antyaliasingiem, b) bez antyaliasingu

4.1.3. Reprezentacja obrazu w komputerze

Co widzi ludzkie oko?

Wiemy, że źródłem życia jest światło i energia z nim związana. Dzięki światłu możliwe jest również widzenie tego, co nas otacza. Światło widzialne jest częścią promieniowania słonecznego, na którą reaguje siatkówka ludzkiego oka. Promieniowanie to zawiera się w przybliżeniu w zakresie długości fal od 380 nm do 720 nm. Jak widać na rysunku 4.8, światło widzialne stanowi jedynie maleńki wycinek całego zakresu promieniowania słonecznego. Niektóre długości fal z przedziału widzialnego odbierane są jako osobne barwy i tworzą widzialne spektrum światła: czerwone, pomarańczowe, żółte, zielone, niebieskie i fioletowe. Barwy te każdy z nas widział i zna jako kolory tęczy.



Rysunek 4.8. Zakres światła widzialnego wraz z pełnym zakresem promieniowania elektromagnetycznego

Barwa

Czy słów „kolor" i „barwa" można używać wymiennie? W zasadzie tak, w języku polskim bowiem słowa te podawane są jako synonimy. Jednak w poligrafii i malarstwie terminy te są rozróżniane. *Barwa* to wrażenie wzrokowe, natomiast *kolor* (z łac. *color*) jest cechą materii (na przykład farby). Farba o określonym kolorze po nałożeniu na różne podłoża, oświetlona różnymi źródłami światła, może dać efekt widzenia różnej barwy. Wrażenia barwne odbierane przez człowieka są zależne również od takich zmiennych czynników jak: nastrój, choroba, gorączka, spożywane leki, zmęczenie czy brak niektórych witamin i składników mineralnych. Również wiek człowieka ma znaczenie. Im jesteśmy starsi, tym gorzej widzimy barwy niebieskie i zielone. Dlatego żaden pomiar nie jest w stanie oddać subiektywnego odczucia barwy przez ludzkie oko.

Na siatkówce ludzkiego oka znajduje się około 6 mln czopków, które reagują na odpowiednie długości fal z widma światła widzialnego. Czopki dzielą się na trzy grupy, z których jedna reaguje na światło czerwone, druga — na zielone, a trzecia — na niebieskie.

Mózg porównuje natężenie padającego światła, interpretując jego barwę. To zjawisko stało się podstawą konstrukcji urządzeń generujących barwne obrazy, takich jak telewizory czy monitory.

Modele barw

Modele barw powstały z myślą o odwzorowaniu sposobu ludzkiego widzenia.
W tradycyjnym ujęciu barwami podstawowymi, czyli takimi, których nie można otrzymać z mieszania innych, są barwy: czerwona, niebieska i żółta. Aby namalować na płótnie wielobarwny obraz, wystarczy użyć farb w tych trzech kolorach, z nich powstają bowiem barwy złożone. I tak po zmieszaniu farby niebieskiej z żółtą otrzymamy farbę zieloną. A od zastosowanej proporcji będzie zależał jej odcień. Jednak nie zawsze te trzy kolory są barwami podstawowymi. Pojęcie koloru podstawowego zależy bowiem od tego, czy mamy do czynienia ze światłem emitowanym (kineskop, monitor), czy też odbitym (obraz na płótnie, ilustracja w książce, pomalowana ściana).

Brak możliwości jednoznacznego opisu barw w taki sposób, aby było to użyteczne we wszystkich praktycznych zastosowaniach, sprawia, że istnieje wiele różnych modeli barw. Niektóre opracowane są specjalnie dla konkretnych zastosowań. Modele barw zostały podzielone na trzy grupy:

- niezależne od urządzenia: CIE XYZ, CIE La*b*,
- kojarzone ze sprzętem: RGB, CMY, CMYK,
- zorientowane na użytkownika: HSB, HSL, HSV.

Szczegółowo omówimy drugą grupę, czyli modele RGB, CMY i CMYK.

Model CIE XYZ

Litery CIE przed XYZ pochodzą od nazwy Międzynarodowej Komisji Oświetlenia (franc. *Commission Internationale de l'Eclairage*), która w 1931 roku stworzyła definicję przestrzeni barw. CIE XYZ jest specjalną paletą barw skonstruowaną przede wszystkim pod kątem postrzegania barw przez fotoreceptory ludzkiego oka. Aby umożliwić opis w dwóch wymiarach, wprowadzono przestrzeń barw *CIE xyY*, która przelicza składowe barw X, Y, Z na współrzędne x, y, Y, gdzie x i y określają barwę, a Y wskazuje jej jasność.

Model CIE La*b*

Zawiera najszerszą zdefiniowaną matematycznie przestrzeń barw. Jest to najważniejszy model barw grafiki komputerowej, który rozwiązuje problem różnic w odtwarzaniu kolorów wynikających z użycia różnych monitorów i urządzeń drukujących. Został zaprojektowany tak, aby nie zależał od urządzenia, czyli odtwarzał takie same kolory niezależnie od urządzenia wyjściowego.

Kolor Lab tworzony jest ze składnika luminancji albo jasności oraz dwóch składników chromatycznych: składowej *a* zmieniającej się od barwy zielonej do czerwonej oraz składowej *b* zmieniającej się od barwy niebieskiej do żółtej. Przestrzeń barw jest reprezentowana przez kulę, w której trzy osie to: *jasność*, *a* i *b* (*–b* to kolor niebieski, *+b* to kolor żółty, *–a* to kolor zielony, *+a* to kolor czerwony).

Model ten oparty jest na modelu kolorów zaproponowanym przez Międzynarodową Komisję Oświetlenia (CIE), jako międzynarodowy standard miary kolorów.

Model RGB

W modelu, zwanym *modelem RGB*, barwa jest przedstawiana jako trzy liczby (R, G, B). Każda z nich reprezentuje jasność barw składowych *emitowanego światła*, którymi są: czerwona (*Red*), zielona (*Green*) i niebieska (*Blue*).

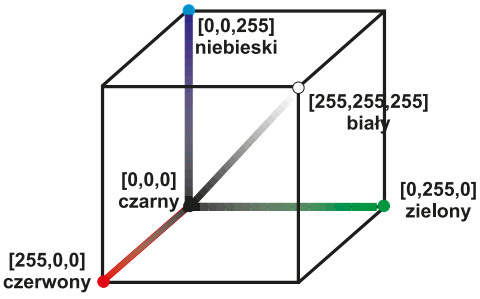
Dowolna *barwa złożona* może powstać przez tzw. *addytywne* mieszanie *barw prostych* (sumowanie składników). Taki sposób tworzenia barw jest stosowany w lampach kineskopowych telewizorów, na ekranach monitorów i laptopów. Monitory komputerowe tworzą różne barwy, emitując czerwone, zielone i niebieskie wiązki światła. Czarny ekran to wynik braku emisji światła, a biały to wynik zsumowania świateł (addytywnego zmieszania) R+G+B z maksymalną wartością.

Najczęściej stosowany jest 24-bitowy zapis kolorów (po 8 bitów na każdą z barw składowych). Zatem każda barwa jest zapisana za pomocą składowych, które przyjmują wartość z zakresu 0 – 255. Wartość 0 wszystkich składowych daje kolor czarny, natomiast wartość 255 wszystkich składników to kolor biały.

Dla pełnego zilustrowania modelu RGB można użyć sześcianu barw pokazanego na rysunku 4.9. Jeżeli na osiach odłożymy barwy RGB, otrzymamy przestrzenny układ barw prostych i złożonych. Patrząc na sześcian, należy zwrócić uwagę, że intensywność barw prostych (RGB) jest zmienna — przyjmują one wartości z zakresu od 0 do 255. Natomiast wzdłuż przekątnej sześcianu biegnie linia szarości — od czerni do bieli.

Z połączenia barw RGB w dowolnych kombinacjach ilościowych można otrzymać szeroki zakres barw pochodnych, na przykład z połączenia barwy zielonej i czerwonej powstaje barwa żółta.

Model RGB jest jednak modelem teoretycznym, a jego odwzorowanie zależy od urządzenia (ang. *device dependent*), co oznacza, że każde z urządzeń może mieć własny zakres barw możliwych do uzyskania.



Rysunek 4.9. Sześcian modelu barw RGB

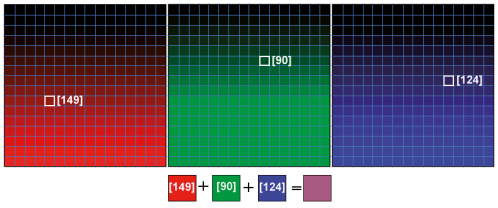
Definicja

Każdy barwny obraz cyfrowy powstały w wyniku skanowania lub fotografowania cyfrowego tworzony jest w modelu RGB.

Barwa każdego piksela obrazu barwnego może być zapisana w trzech kanałach RGB, czyli w kanale czerwonym, zielonym i niebieskim. Każdy kanał reprezentuje więc jedną z trzech barw składowych światła białego. Dowolna barwa wynikowa każdego piksela powstaje przez zmieszanie tych trzech barw — od zera, czyli zerowego udziału składowej barwy, do maksymalnego jej udziału.

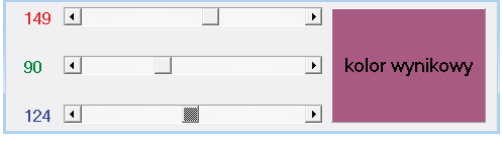
Jeżeli przyjmiemy, że każdy kanał ma 256 odcieni swojej barwy, co jest obecnie najczęściej stosowanym rozwiązaniem, to liczba możliwych kombinacji kolorów wynosi dokładnie 256^3, czyli ponad 16,7 mln możliwych barw. Wśród nich są oczywiście czerń i biel oraz odcienie szarości.

Na rysunku 4.10 przedstawiono tablice możliwych kolorów w kanałach obrazu i sposób tworzenia z nich dowolnej barwy wynikowej.



Rysunek 4.10. Tablice barw podstawowych modelu RGB możliwych do uzyskania w każdym kanale

Ćwiczenie 4.3. Na rysunku 4.11 zobrazowano kanały barw RGB w postaci suwaków. Przesuwanie nimi pozwala natychmiast zobaczyć efekt w postaci barwy wynikowej. Obok pasków wyświetlana jest wartość liczbowa z przedziału od 0 do 255. Tę prostą w obsłudze aplikację zamieszczono na płycie CD — *koloryRGB.exe*. Po uruchomieniu aplikacji zwróć uwagę na skrajne wartości każdego kanału oraz na czerń (wszystkie wartości wyzerowane) i biel (maksymalne wartości RGB). Jeżeli położenie wszystkich suwaków będzie wskazywało te same wartości dla wszystkich kanałów, otrzymamy odcień szarości.



Rysunek 4.11. Mieszanie barw RGB z wykorzystaniem 256 poziomów każdego kanału barwnego

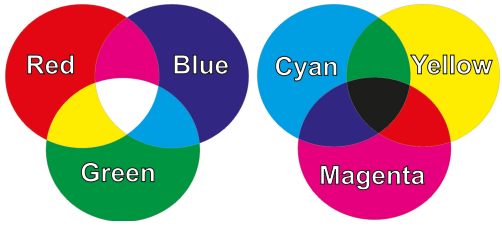
Model CMY

Drugim popularnym modelem reprezentacji barw jest model CMY, w którym barwami podstawowymi są kolory: żółtoniebieski (*Cyan*), bliski purpury fioletoworóżowy, nazywany biskupim (*Magenta*), i żółty (*Yellow*). W modelu CMY zachodzi mieszanie *subtraktywne* kolorów światła odbitego, a efekt widzenia barw powstaje przez pochłanianie niektórych długości fal przez powierzchnię, od której odbija się światło. Efekt ten widzimy, patrząc na wydruk na papierze, na pomalowane powierzchnie lub spoglądając pod światło przez zespół filtrów. Farba drukarska, pokrywając papier, tworzy filtr, a niepochłonięte długości fal świetlnych docierają do oka, wywołując wrażenie określonej barwy. Subtraktywne mieszanie barw wykorzystywane jest w drukarkach.

Model CMYK

Ponieważ w praktyce trudno otrzymać prawdziwie czarny kolor poprzez zmieszanie barw podstawowych CMY, często dodaje się dodatkowo kolor czarny (*blacK*). Mówimy wówczas o modelu *CMYK*. Za pomocą tych czterech barw drukarki mogą generować żądane odcienie kolorów.

Na rysunku 4.12 zostały przedstawione oba najpopularniejsze modele barw: RGB i CMY. Widać na nich wyraźnie, że mieszając barwy modelu RGB, dążymy do bieli, natomiast w modelu CMY — do czerni.



Rysunek 4.12. Mieszanie kolorów w modelach RGB i CMY

Ciekawostka

Tryb RGB jest stosowany jako standard w wyświetlaniu obrazów na monitorach. Stanowi on podstawę przy konwertowaniu obrazka na tryb CMYK (który z kolei ma zastosowanie w drukarkach offsetowych). Aparaty cyfrowe czy też skanery z reguły generują pliki kolorowych obrazów w modelu RGB. Niektóre naświetlarki slajdów i folii barwnych wymagają danych RGB (większość wymaga CMYK).

Modele HSB, HLS i HSV

Modele te są oparte na sposobie postrzegania koloru przez człowieka. Konstruują barwę na podstawie trzech składowych: odcienia (***H**ue*) i nasycenia (***S**aturation*) oraz jasności (***B**rightness*) lub wartości (***V**alue*) albo oświetlenia (***L**ightning*).

Istnieje możliwość przeliczenia barwy z jednej reprezentacji na drugą. Między modelami RGB i CMY (CMYK) można przeliczyć barwy bezstratnie. Teoretycznie istnieje odpowiednik barwy w każdym z tych modeli. W praktyce jednak każde urządzenie pozwala na reprodukcję pewnego zakresu barw wynikającego ze stosowanej technologii. W ten sposób teoretyczne przeliczenie jest przenoszone na praktyczne możliwości techniczne. Trzeba natomiast podkreślić, że różnice w jakości między na przykład obrazem wyświetlanym przez monitor a tym samym obrazem wydrukowanym na drukarce nie wynikają, jak niektórzy sądzą, z przejścia z RGB na CMYK, ale z możliwości technologicznych tych urządzeń. Praktycznie zawsze zakres barw monitora będzie szerszy niż zakres ich odwzorowania przez drukarkę.

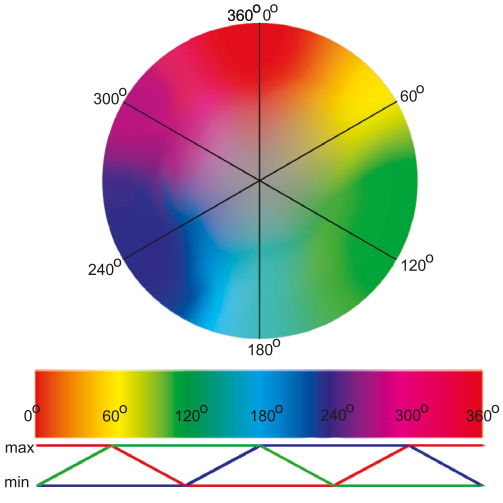
Atrybuty barwy

W teorii koloru wyróżnia się trzy atrybuty barwy, które pozwalają na jej jednoznaczną identyfikację. Są to:

- odcień (H),
- nasycenie (S),
- jaskrawość (B).

Odcień (ang. *Hue*)

W słownikach można znaleźć tłumaczenie angielskiego słowa *Hue* na język polski jako *odcień*, ale i *barwa*. Każda barwa ma określony odcień. Czym więc jest odcień? Jest kątem określającym położenie barwy na kole kolorów, zwanym kołem chromatycznym (*chroma*, z grec. *barwa*), przedstawionym na rysunku 4.13. Popatrzmy na ten rysunek. Umowny początek koła kolorów



Rysunek 4.13. Koło chromatyczne

znajduje się w miejscu odpowiadającym czystej czerwieni, czyli w modelu RGB (255, 0, 0). Każdej barwie odpowiada pewna wartość wyrażana w stopniach w skali od 0° do 360°. Przyjęto, że kolorom odpowiadają następujące przybliżone wartości kątowe: czerwony — 0° (Hue = 0), zielony — 120° (Hue = 120) i niebieski — 240° (Hue = 240).

Pełne koło barw można podzielić na sześć obszarów. Każdy obszar ma pewne charakterystyczne cechy — przedstawiono je w tabeli 4.1. Zamieszczone w niej dane oraz ustawienie barwy w programie graficznym zostały zilustrowane na wykresie widocznym pod kołem barw na rysunku 4.13.

Tabela 4.1. Charakterystyczne cechy barw położonych w określonych obszarach koła chromatycznego

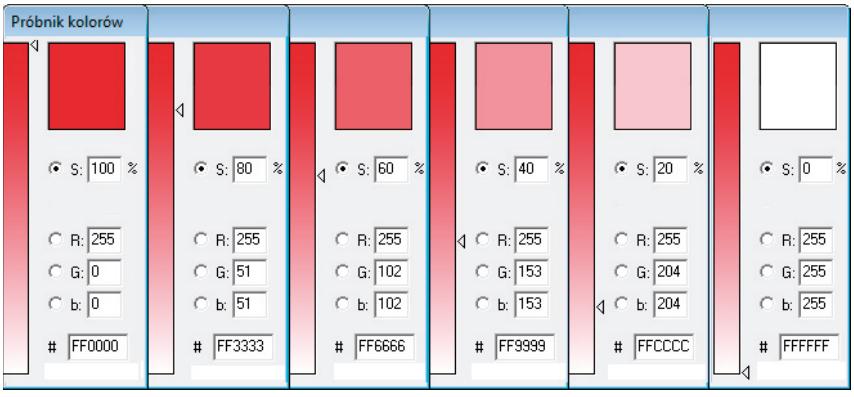
Nr	Obszar	R	G	B
1	od 0° do 60°	maks.	rośnie	min.
2	od 60° do 120°	maleje	maks.	min.
3	od 120° do 180°	min.	maks.	rośnie
4	od 180° do 240°	min.	maleje	maks.
5	od 240° do 300°	rośnie	min.	maks.
6	od 300° do 360°	maks.	min.	maleje

Odcień z zakresu od 0° do 240° opisuje kolory występujące w widmie światła białego. Są to tzw. *barwy proste* (monochromatyczne, widmowe). Kolory z zakresu od 240° do 360° można uzyskać jedynie przez mieszanie w różnych proporcjach światła niebieskiego (lub fioletowego) i zielonego. Barwy te, razem z prostymi, nazywane są *barwami zasadniczymi* (czystymi).

Ćwiczenie 4.4. Wykorzystajmy raz jeszcze aplikację zamieszczoną na płycie CD — *koloryRGB.exe*. Dodaj do tabeli 4.1 jeszcze jedną kolumnę pod nazwą *Barwa*. W oknie aplikacji ustawiaj suwaki zgodnie z wartościami RGB dla poszczególnych obszarów i obserwuj powstałe kolory wynikowe. W przypadku rosnących lub malejących wartości dla *R*, *G* lub *B* zauważ, jak zmieniają się kolory wynikowe. Nazwij obserwowane barwy. Dodaj do tabeli wartości graniczne obszarów: `0°`, `60°`, `120°`, `180°`, `240°`, `300°` i `360°` i uzupełnij kolumny *R*, *G*, *B* oraz *Barwa* na podstawie koła barw oraz ustawień w aplikacji *koloryRGB*. Teraz tabela zawiera wszystkie barwy zasadnicze. Zauważ, że w każdym przypadku przynajmniej jedna składowa przyjmuje wartość minimalną, czyli 0.

Nasycenie (ang. *Saturation*)

Dowolną barwę zasadniczą możemy zmieszać ze światłem białym. Im więcej światła białego, tym bledsza (bardziej wyblakła) staje się barwa. Przy bardzo dużym udziale światła białego przestajemy zauważać odcień i pozostaje jedynie wrażenie bieli. Jeżeli na przykład do czystej czerwieni będziemy dodawać światło białe, to najpierw barwa stanie się czerwonoróżowa, potem różowa, bladoróżowa, a w końcu biała. Na rysunku 4.14 zilustrowano te kolejne, skokowe etapy różnego nasycenia barwy czerwonej. Zwróć uwagę na to, że składowa *R* wynikowego koloru zawsze ma wartość maksymalną, czyli 255, zmieniały się jedynie składowe *B* i *G*, przyjmując jednakowe wartości.



Rysunek 4.14. Czerwień w różnych nasyceniach

Nasycenie barwy może przyjmować wartości z przedziału od 0 do 255 albo od 0% do 100%. Na rysunku 4.14 atrybut S (nasycenie) jest podawany w procentach, ale oczywiście można go podać w wartościach liczbowych.

W dowolnej barwie udział światła białego określony jest jednoznacznie przez wartość najmniejszej składowej. Na przykład barwę (228, 146, 124) możemy przedstawić jako sumę dwóch barw: barwy (124, 124, 124), która reprezentuje udział światła białego, oraz barwy zasadniczej (104, 22, 0).

Dla dowolnej barwy możemy zatem zapisać następujące równanie:

$(R, G, B) = (R-m, G-m, B-m) + (m, m, m)$, *gdzie* $m = \min.(R,G,B)$

Spróbujmy rozpatrzyć nasz przykład barwy RGB (149, 90, 124), zilustrowany na rysunku 4.10.

Minimalną wartością barwy RGB (149, 90, 124) jest składowa zielona, zatem m = 90, więc RGB (149, 90, 124) = RGB (59, 0, 34) + RGB (90, 90, 90). Obrazowo ten przykład pokazano na rysunku 4.15.



Rysunek 4.15. Nasycenie koloru RGB (149, 90, 124)

Pierwszy składnik nazywamy *składnikiem chromatycznym barwy*, drugi jest *składnikiem achromatycznym*.

Wszystkie barwy mające tylko drugi składnik nazywamy *barwami achromatycznymi*. Są to wszystkie odcienie szarości — od czerni do bieli.

Wszystkie barwy, w których można wyróżnić pierwszy składnik, nazywane są *barwami chromatycznymi*.

Definicja

Nasycenie można traktować jako miarę udziału czynnika chromatycznego w barwie.

Barwy achromatyczne (odcienie szarości) mają nasycenie równe zero.

Jaskrawość (ang. *Brightness*) i jasność (ang. *Lightness*)

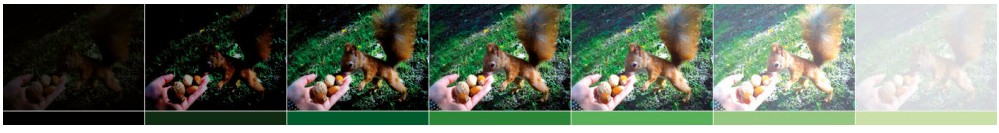
Jaskrawość (*brightness*, *value*) często bywa mylona z *jasnością* (*lightness*). Są to jednak dwie różne informacje o obrazie.

Zerowa jaskrawość odpowiada czerni, a maksymalna jaskrawość — w pełni nasyconym kolorom, co pokazano na rysunku 4.16.



Rysunek 4.16. Jaskrawość obrazu

Aby wywołać zmianę wrażenia barwy, można również zmienić oświetlenie. Odsuwając źródło światła od obserwowanej powierzchni obrazu, nie stwierdzimy zmiany odcienia ani zmiany nasycenia, ale wrażenie coraz słabszego światła, a co za tym idzie — coraz ciemniejszego obrazu. Co się stanie, gdy światło zbyt silnie oświetli obraz? A któż z nas nie widział prześwietlonego zdjęcia? Zerowa jasność zatem także odpowiada czerni, ale maksymalna jasność odpowiada bieli, co zilustrowano na rysunku 4.17. Dla pełniejszej ilustracji obu tych atrybutów użyto tego samego zdjęcia bazowego. Dodatkowo dla pobranego z obrazka koloru zielonego przedstawiono stopniowe zmiany jego jaskrawości i jasności.



Rysunek 4.17. Jasność obrazu

Człowiek lepiej rozróżnia nasycenie i jasność niż kolor widma światła.

Własności barwy — głębia bitowa

Obraz cyfrowy jest liczbowym zapisem występujących w nim barw. Jednym z parametrów tego zapisu jest *głębia bitowa* (głębia kolorów) informująca o liczbie barw możliwych do odwzorowania w obrazie. Jest ona zależna od liczby bitów przeznaczonych do zapisania informacji o barwie każdego piksela obrazu. Jednostką głębi bitowej jest *bpp* (ang. *bits per pixels*, bity na piksel). Bit, jak wiemy, jest elementarną jednostką informacji.

Każdemu pikselowi obrazu można przypisać tyle *poziomów jasności*, ile na przeznaczonej liczbie bitów można utworzyć możliwych kombinacji bitów.

Jeżeli każdemu pikselowi w obrazie cyfrowym przypiszemy jeden bit, to każdy z nich może przyjąć jedną z dwóch możliwych wartości: `0` albo `1`. Możemy zatem przypisać tym stanom dwie barwy, czarną lub białą. A powstały z tych dwóch barw obraz to tzw. obraz *kreskowy*, o *głębi bitowej 1 bpp* lub *1-bitowej*.

Jeżeli każdemu pikselowi w obrazie cyfrowym przypiszemy cztery bity, to mogą one przyjąć 16 (2^4) różnych stanów — kolejno: `0000`, `0001`, `0010`, ..., `1101`, `1110`, `1111`. Każdemu z tych stanów możemy przypisać jedną z 16 barw. A obraz ma głębię bitową 4 bpp.

Jeżeli każdemu pikselowi w obrazie cyfrowym przypiszemy osiem bitów, to mogą one przyjąć 256 (2^8) różnych stanów, od `00000000` do `11111111`. Każdemu z tych stanów możemy przypisać jedną z 256 barw lub jeden z odcieni szarości, od bieli do czerni. Jest to obraz w skali szarości i mówimy, że ma on głębię bitową 8 lub głębię 8-bitową. Obraz w skali szarości jest zapisany w jednym kanale, zwanym *Grayscale*.

Opis obrazu kolorowego jest bardziej złożony. Przypomnijmy, że wynikowa barwa piksela w obrazie cyfrowym składa się z sumy trzech barw składowych: czerwonej, zielonej i niebieskiej, przy czym barwy te mogą mieć różne *poziomy jasności* — od zerowej (brak barwy składowej) do maksymalnej (największy udział barwy składowej). Jeżeli każdej z nich przydzielimy osiem bitów, to każda z tych barw składowych może przyjąć 256 różnych odcieni jasności (2^8). W cyfrowym obrazie kolorowym, złożonym z trzech kanałów: R, G i B, mamy więc przydzielone 24 bity (8×3) do każdego piksela obrazu.

Obraz taki może więc mieć tyle kolorów, ile jest możliwych kombinacji każdego poziomu jasności jednego kanału z każdym poziomem jasności z pozostałych kanałów składowych, czyli $256 \times 256 \times 256 = 2^8 \times 2^8 \times 2^8 = 2^{24}$, co daje około 16,7 mln barw. Ma on więc głębię bitową 24 bpp.

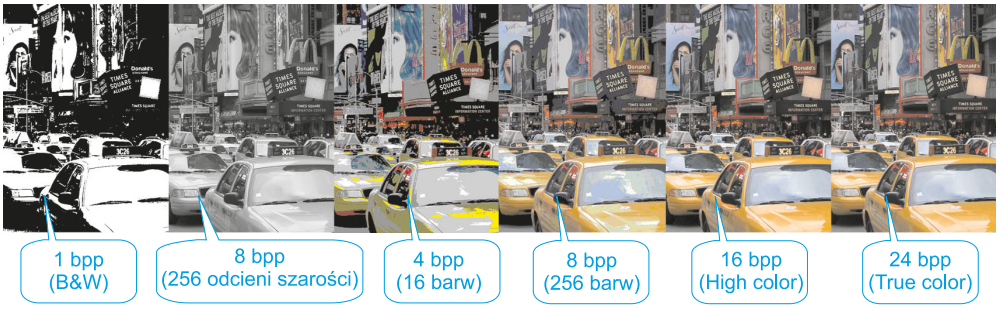
Kolor podajemy następująco: RGB (R = *149*, G = *90*, B = *124*) = RGB (*149, 90, 124*), gdzie składowe jasności poszczególnych barw, podawane w zapisie dziesiętnym, musimy oddzielić przecinkami. Częściej używanym zapisem składowych jasności barwy jest zapis w systemie szesnastkowym, w którym każdy kanał zapisywany jest dwucyfrowo. Zapis barwy w systemie szesnastkowym poprzedza znak #. I tak, barwę RGB (*149, 90, 124*) można zapisać jako: *#955A7C*. Konieczne jest zapisanie dwóch cyfr dla każdej składowej, dzięki czemu nie ma w tym zapisie konieczności używania przecinków jako separatorów składowych koloru, na przykład RGB (*149, 3, 124*) = *#95037C*.

Poszczególne głębie bitowe, ich nazwy oraz liczbę dostępnych barw zestawiono w tabeli 4.2.

Tabela 4.2. Głębie bitowe

Lp.	Nazwa głębi bitowej	Wartość głębi	Liczba dostępnych barw
1	B&W (Black&White)	1	$2^1 = 2$
2	Windows	4	$2^4 = 16$
3	Grayscale (skala szarości)	8	$2^8 = 256$
4	256 color	8	$2^8 = 256$
5	High color	16	$2^{16} \approx 65{,}5$ tys. barw
6	True color	24	$2^{24} \approx 16{,}7$ mln barw

Na rysunku 4.18 przedstawiono — pochodzące z tego samego oryginału — obrazy cyfrowe zapisane z różnymi głębiami bitowymi podanymi w tabeli 4.2. Jak widać, powstałe obrazy mają różną kolorystykę. Utworzone pliki zawierają jednakowej wielkości obrazy, jednak różnią się znacznie rozmiarem. Im mniej bitów użyto do zapisu informacji o kolorze każdego piksela obrazu, tym mniejszy plik powstał.



Rysunek 4.18. Porównanie obrazów zapisanych z różnymi głębiami bitowymi

Ćwiczenie 4.5. W edytorze Paint otwórz jedno ze zdjęć zamieszczonych w folderze *Fotografie* na płycie CD dołączonej do podręcznika. Zapisz obraz w formacie BMP w czterech różnych głębiach bitowych, w kolejności: *Mapa bitowa 24-bitowa*, *Mapa bitowa 256-kolorowa* (8 bitów), *Mapa bitowa 16-kolorowa* (4 bity) oraz *Mapa bitowa monochromatyczna* (1 bit). Obserwuj zmiany obrazu oraz paletę dostępnych kolorów. Odczytaj z paska stanu wymiary zdjęcia (w pikselach) i oblicz rozmiary powstałych plików. Swoje obliczenia porównaj z faktycznymi rozmiarami zapisanych na dysku plików graficznych. Przykładowo plik zawierający kwadratowy obraz o 100-pikselowym boku zapisany w 16 kolorach będzie miał rozmiar 5000 B (100×100×4 b = 40000 b = 5000 B).

4.1.4. Istota obrazu cyfrowego

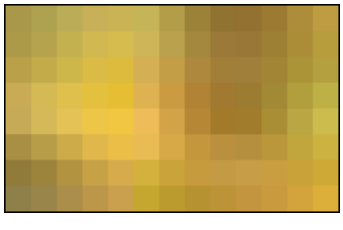
Obrazy cyfrowe, które odwzorowują oryginały, nazywane są *bitmapowymi*, *mapami bitowymi* lub po prostu *bitmapami*. Obrazy cyfrowe to zawsze prostokątne obszary zbudowane z kwadratowych elementów zwanych *pikselami*. Obraz ma dwa wymiary: wysokość i szerokość. Czasami programy graficzne i przeglądarki internetowe wyświetlają rysunki o kształtach nieprostokątnych. Są to obrazy, których fragmenty są przezroczyste.

Definicja

Piksel (ang. *pixel*) stanowi jednostkę wielkości obrazu cyfrowego. Wyraz ten jest utworzony z dwóch angielskich słów: *picture* i *element*.

Całkowita liczba pikseli w obrazie cyfrowym jest iloczynem liczby pikseli leżących wzdłuż obu boków jego prostokątnego obrysu.

Cechą charakterystyczną pojedynczego piksela jest jego *monochromatyczność* (monobarwność, jednobarwność). W efekcie kolorowy obraz cyfrowy cechuje skokowa zmiana barwy przy przechodzeniu od jednego piksela do drugiego. Złudzenie płynnej zmiany kolorów otrzymać można jedynie przy bardzo niewielkich zmianach barwy sąsiadujących ze sobą pikseli. Takie właśnie łagodne przejście pokazano na rysunku 4.19.



Rysunek 4.19. Powiększony fragment zdjęcia z widocznymi jednobarwnymi pikselami

4.1.5. Cyfrowy obraz ze skanera i cyfrowego aparatu fotograficznego

Obraz cyfrowy może powstać między innymi w wyniku skanowania lub fotografowania cyfrowego. Aby otrzymać obraz cyfrowy najlepszej lub optymalnej jakości, trzeba wykorzystać wszystkie możliwości urządzeń, z którymi mamy do czynienia.

Definicja

Oryginały analogowe to obiekty (materialne), z których otrzymujemy obraz cyfrowy.

Tworzenie obrazu w wyniku skanowania lub fotografowania przebiega według analogicznych procesów i tych samych teoretycznych podstaw. Różnica jest tylko w odmiennych urządzeniach realizujących te procesy: skaner do skanowania i cyfrowy aparat fotograficzny do fotografowania. Oba urządzenia z analogowych oryginałów tworzą cyfrowy obraz.

Więcej informacji o skanerach, skanowaniu i oprogramowaniu do optycznego rozpoznawania tekstu zamieszczono na płycie CD dołączonej do podręcznika, w pliku *skaner.pdf*.

4.1.6. Kompresja stratna i bezstratna

Kompresja obrazów jest sposobem na redukcję wielkości plików bitmapowych. Dzięki niej do zachowania danych potrzebujemy mniej miejsca na dysku. Zmniejszając pliki z obrazem, dźwiękiem czy filmem, możemy zastosować *kompresję bezstratną* lub *kompresję stratną*.

Kompresja bezstratna

Kompresja bezstratna (ang. *lossless compression*) to ogólna nazwa metod takiego kodowania informacji do postaci zawierającej zmniejszoną liczbę bitów, aby całą informację dało się z tej postaci odtworzyć, czyli jest to zmniejszenie rozmiaru pliku bez utraty żadnych informacji. Stosowane algorytmy kompresji to: kodowanie Huffmana, RLE, LZW i inne.

Algorytmy kompresji bezstratnej dobrze kompresują „typowe" dane, czyli takie, w których występuje znaczny nadmiar informacji.

RLE (ang. *Run-Length Encoding*)

Metoda kodowania długości ciągów jest dość prosta. Jej działanie polega na opisywaniu ciągów tych samych znaków (bitów, bajtów, symboli itp.) za pomocą zliczenia powtórzeń, określenia długości podciągu, a dokładniej za pomocą par: licznik powtórzeń barwy – barwa.

Na przykład w ciągu 20 pikseli (rysunek 4.20) będącym jedną linią grafiki bitmapowej występuje, w ogromnym uproszczeniu, następujący zapis: NNNCCCCCCCNNNCCCNZCN, czyli 20 znaków z informacją o kolejnych pikselach obrazu.



Rysunek 4.20. Linia grafiki bitmapowej (w dużym powiększeniu)

Trzy niebieskie piksele można zapisać, stosując metodę RLE, jako 3N (ciąg 3 pikseli niebieskiej barwy), dalej C występuje 7 razy — 7C itd. Ostatecznie uzyskujemy ciąg: 3N7C3N3C1N1Z1C1N, składający się z 16 znaków, podczas gdy kodowany ciąg składał się z 20 znaków. W praktyce zapobiega się kodowaniu serii jednoelementowych, a więc powstawaniu ciągów postaci 1N1Z1C1N. Teraz, po niewielkim uproszczeniu toku rozumowania, uzyskuje się ciąg 12-znakowy 3N7C3N3CNZCN. Oto porównanie:

NNNCCCCCCCNNNCCCNZCN — dane oryginalne,

3N7C3N3CNZCN — dane skompresowane metodą RLE.

Kompresja RLE jest stosowana między innymi w formatach zapisu obrazu, takich jak: PCX, BMP, TGA, w faksach, a także w dokumentach PDF.

LZW

Metoda LZW (*Lempel-Ziv-Welch* — nazwiska trzech twórców tej kompresji danych) jest wykorzystywana między innymi w formacie zapisu grafiki GIF. Kodowanie LZW było przez pewien czas algorytmem objętym patentem, co miało wpływ na podjęcie prac nad nowym algorytmem kompresji obrazów, które zaowocowały powstaniem formatu PNG.

LZW jest tzw. algorytmem słownikowym. Metody słownikowe poszukują dokładnych wystąpień danego ciągu znaków.

W metodzie LZW wartościom wejściowym o zmiennej długości, czyli poziomym ciągom pikseli tworzących powtarzający się wzór, przypisywane są kody o stałej długości. Im dłuższe są więc ciągi poziome, tym lepsze efekty kompresji.

Pierwszym etapem algorytmu kompresji LZW jest wstępne wypełnienie słownika alfabetem — wszystkimi symbolami. W naszym przykładzie będą to N, C i Z. Przy kodowaniu trzeba brać pod uwagę to, czy wczytany ciąg znaków znajduje się już w słowniku, czy może pojawił się po raz pierwszy.

Algorytm kompresji (kodowania) LZW można zapisać następująco:

```
Krok 1. Wypełnij słownik alfabetem źródła informacji.
Krok 2. p:=pierwszy symbol wejściowy;
        dopóki są dane na wejściu, wykonuj:
Krok 3. Wczytaj znak z.
Krok 4. Jeżeli ciąg p+z znajduje się w słowniku, przedłuż ciąg p, p:=p+z;
        w przeciwnym razie:
        { wypisz kod dla p, który znajduje się w słowniku;
        dodaj ciąg p+z do słownika;
```

```
        przypisz p:=z
        }
```
Krok 5. `Wypisz kod dla p.`

Przykład 4.1.

Dla ciągu pikseli pokazanego na rysunku 4.20 utwórz algorytm słownikowy LZW.

Szczegółowy (zawierający pełną analizę danych) proces tworzenia słownika i wypisywania wyniku kompresji zamieszczony został w dokumencie *LZW.pdf* na płycie CD dołączonej do podręcznika.

Kodując tak mały obrazek, przeszliśmy przez etap tworzenia słownika. Prawdziwa korzyść z zastosowania metody słownikowej widoczna jest przy kodowaniu obrazu składającego się z większej liczby pikseli. Wówczas, korzystając ze zdefiniowanych w słowniku kombinacji pikseli, używamy kodu, który oznacza niejednokrotnie duże fragmenty obrazu.

Porównując opisane powyżej dwie metody kodowania grafiki bitmapowej: RLE i LZW, możemy dostrzec, że algorytm RLE jest zdecydowanie prostszy. W metodzie RLE na wyjściu wypisywane są pary (licznik — znak), oznaczające liczbę kolejnych sąsiadujących pikseli tego samego koloru. W metodzie LZW wypisywany jest tylko kod oznaczający pozycję (numer) w słowniku. W nim zapisywane są kombinacje barw sąsiadujących pikseli występujących w danym obrazie.

Program kodujący nazywamy *koderem*, natomiast program, który dokonuje powrotnej zamiany, nazywamy *dekoderem*.

Kodowanie Huffmana

Metoda Huffmana (ang. *Huffman coding*) to jedna z powszechnie stosowanych metod bezstratnej kompresji danych. Algorytm Huffmana nie jest najefektywniejszą metodą, dlatego też praktycznie nie używa się go samodzielnie. Często natomiast wykorzystywany jest jako ostatni etap w różnych metodach kompresji, zarówno bezstratnej, jak i stratnej, nie tylko obrazu (JPEG), ale i dźwięku (MP3). Mimo że nie jest doskonały, stosuje się go ze względu na prostotę oraz brak ograniczeń patentowych.

Metoda ta jest oparta na *strategii zachłannej*. Kompresja polega na zastępowaniu znaków kompresowanego pliku krótszymi kodami (a ściślej mówiąc, ciągami bitów), dzięki czemu wynikowy ciąg bitów jest znacznie krótszy od wejściowego.

Dla łatwiejszego zrozumienia posłużymy się w tym przypadku przykładem z tekstem, a nie obrazem. Jak wiadomo, znaki są kodowane na 8 bitach, dzięki czemu można zakodować aż do 256 znaków (kody ASCII od 0 do 255). Bardzo rzadko się jednak zdarza, by w pliku występowały wszystkie znaki ASCII, wobec tego można wykorzystać mniej bitów i tak zaoszczędzić sporo miejsca.

Znaki można kodować na dwa sposoby:
- za pomocą kodów o stałej długości,
- za pomocą kodów o zmiennej długości.

Metoda pierwsza jest bardzo prosta, lecz mniej skuteczna niż druga. Polega ona na przypisaniu znakom liczb porządkowych. Załóżmy, że nasz plik został zapisany za pomocą znaków alfabetu $A = \{a, b, c, d, e\}$. Alfabet ten składa się z 5, a nie z 256 znaków. Nie potrzebujemy więc aż 8 bitów do ich zapisania. Za pomocą 3 bitów można zapisać 8 znaków. Gdyby nasz alfabet składał się z 4 znaków, wówczas wystarczyłyby tylko 2 bity, gdyż $2^2 = 4$.

Załóżmy, że nasz plik ma dokładnie 1 MB, czyli 1024×1024 = 1 048 576 bajtów. Każdy bajt jest kodowany na 8 bitach, czyli nieskompresowany plik zajmuje 8×1 048 576 = 8 388 608 bitów. Sporo. Jak zauważyliśmy jednak, nam wystarczą po 3 bity na znak. Przypiszmy więc każdemu znakowi liczbę porządkową (zapisaną binarnie): $a = 000$, $b = 001$, $c = 010$, $d = 011$, $e = 100$. Zamiast używać 8-bitowych kodów ASCII, używamy 3-bitowych kodów nowego, własnego alfabetu. Sprawdźmy, jakie uzyskaliśmy korzyści: 1 048 576×3 = 3 145 728 bitów, czyli ponad 37% pliku wejściowego. Całkiem dobry wynik. Należy jednak zauważyć, że tak dobry rezultat osiągamy tylko dlatego, że nasz alfabet miał zaledwie 5 znaków. Im więcej liter alfabetu, tym gorszy wynik. Przy alfabecie składającym się ze wszystkich 256 znaków będziemy potrzebowali 8 bitów ($2^8 = 256$), czyli tyle, ile zwykle — w tym przypadku nic nie zaoszczędzimy.

Zwróćmy jeszcze uwagę na sposób zapisu takiego pliku — zamiast słowa „bada" zapisane zostanie zakodowane słowo 001000011000. Ponieważ każdej literze przypisany został trzyznakowy kod, nie ma potrzeby stosowania separatorów. Każde trzy znaki to kolejna litera.

Metoda druga wykorzystuje dla odmiany kody o zmiennej długości. Wykazują one lepsze własności dla znaków występujących częściej. Zauważmy bowiem, że lepiej jest przypisać mniej bitów znakom występującym częściej i więcej bitów znakom występującym rzadziej. Wiąże się z tym jednak pewien problem. Jeśli kody mają różną długość, to nie wiadomo, kiedy kończy się stary kod, a zaczyna nowy. Zamiast stosować dodatkowy znak separacji, należy posługiwać się kodami prefiksowymi.

Definicja

Kod prefiksowy (ang. *prefix-free code* lub *prefix code*), zwany przedrostkowym, to taki, w którym żaden kod znaku nie jest *prefiksem* (przedrostkiem) kodu innego znaku, czyli nie jest częścią początkową innego kodu.

Ćwiczenie 4.6. Kody 101, 01, 010, 10, 1010, 11, 111, 1011, 1110 nie są kodami prefiksowymi, gdyż 10 jest prefiksem 101, 1010 oraz 1011, 101 jest prefiksem 1010 i 1011, a 01 jest prefiksem 010. Zakodowany tekst 101110101011101010 jest nie do odczytania, gdyż nie wiadomo, gdzie kończy się jeden znak, a gdzie zaczyna drugi. Niezbędne tutaj są separatory, które oddzielą podane kody. Można dokonać takiego podziału: 1011|1010|10|1110|10|10. Dopiero teraz zapis ten jest jednoznaczny, ale możliwości było wiele. Dokonaj innego podziału tego szyfru 101110101011101010, tak aby również otrzymać poprawne rozwiązanie.

Ćwiczenie 4.7. Kody: 00, 010, 1101, 100 odpowiadają następującym znakom: *A*-00, *B*-010, *C*-1101 i *D*-100. Kody te są kodami prefiksowymi, żaden kod bowiem nie jest początkiem innego kodu. Odczytując kody prefiksowe z ciągu bitów, wiemy, kiedy kończy się jeden, a zaczyna drugi kod. Dla ciągu bitów 001001101000101001101 analizujemy kolejne cyfry, począwszy od lewej strony. Na początku występuje 00 (*A*). Następnie 100 (*D*). Nie ma najmniejszej wątpliwości, że jest to tekst 00|100|1101|00|010|100|1101, który po rozkodowaniu jest tekstem: *ADCABDC*. Rozszyfruj następujący zaszyfrowany tekst: 01000110100.

Problem przydziału znakom kodów prefiksowych rozwiązał David Huffman. Sposób ten polega na analizie częstości wystąpień poszczególnych znaków i stosowaniu drzew binarnych.

Definicja

Drzewo (ang. *tree*) jest strukturą dynamiczną zbudowaną z elementów zwanych **węzłami** (ang. *node*), z których każdy może mieć jednego **rodzica** (ang. *parent node*) oraz kilku **potomków** (ang. *child node*). Węzeł niemający rodzica nazywamy **korzeniem drzewa** lub **węzłem głównym** (ang. *root node*). Węzeł bez potomków nazywamy **liściem** (ang. *leaf*). Drzewo może mieć wiele liści, lecz tylko jeden korzeń.

Do analizy użyjemy 8 znaków, których wystąpienia zostały zliczone. Każdy znak nazywać będziemy liściem. Wszystkie liście zostały posortowane według częstości wystąpień w kolejności niemalejącej. W każdym liściu znajduje się znak *I – VIII* oraz liczba oznaczająca częstość wystąpień znaku w kompresowanym tekście. W każdym węźle (zielone koło) znajduje się liczba (waga) będąca sumą liczb swoich potomków.

Definicja

Drzewo binarne (ang. *binary tree*) to jeden z rodzajów drzewa (struktury danych), w którym liczba potomków każdego wierzchołka wynosi nie więcej niż dwa. Wyróżnia się wtedy lewego i prawego potomka danego wierzchołka.

Drzewo binarne, w którym liczba potomków każdego wierzchołka wynosi albo zero, albo dwa, nazywane jest **drzewem regularnym**.

Cechą kodów Huffmana jest to, że reprezentujące je drzewo binarne jest regularne, tzn. każdy węzeł ma dwóch potomków lub nie ma żadnego. Początkowo każdy znak (liść) tworzy odrębne poddrzewo, waga korzenia jest równa liczbie wystąpień danego znaku.

Algorytm przebiega następująco:

Krok 1. `Dodaj do posortowanej kolejki wszystkie znaki z alfabetu.`

Krok 2. `Wybierz dwa drzewa o najmniejszej wadze w korzeniu.`

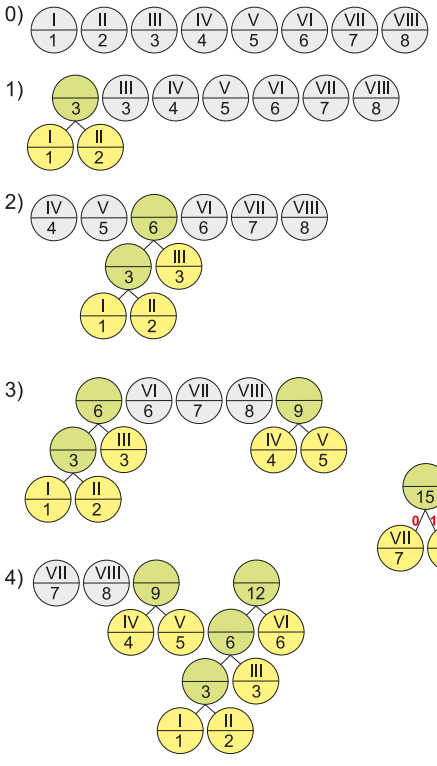
Krok 3. `Scal te drzewa i zsumuj wagi ich korzeni, przypisz tę sumę korzeniowi nowego drzewa.`

Krok 4. `Wstaw nowe drzewo do kolejki, sortując ją.`

Krok 5. `Jeżeli w kolejce jest więcej niż jedno drzewo, wróć do kroku 2.`

Po utworzeniu drzewa należy nadać etykiety krawędziom: lewym `0` i prawym `1`. Następnie można już odczytywać kody znaków.

Na rysunku 4.21 zilustrowano przebieg algorytmu Huffmana dla 8 elementów. Widać również etykiety nadane krawędziom końcowego drzewa: `0` dla lewej krawędzi i `1` dla krawędzi prawej. Te etykiety służą do wyznaczania kodu Huffmana dla każdego znaku. Tworzy go ciąg etykiet krawędzi od korzenia do liścia z danym znakiem.



Rysunek 4.21. Kolejne etapy tworzenia kodu Huffmana

Dla naszego drzewa:

Znak	Kod
I	`11000`
II	`11001`
III	`1101`
IV	`100`
V	`101`
VI	`111`
VII	`00`
VIII	`01`

Jak widać, kody te nie mają równej długości. Ze sposobu tworzenia drzewa wynika fakt, że znaki o większej wartości mają krótsze kody. Znaki, które występują częściej, mają kody krótsze, a te, które występują rzadziej, mają kody dłuższe.

Dekompresja to opisany powyżej rozbiór ciągu bitów pliku skompresowanego na kody Huffmana i odczytanie odpowiednich znaków. Aby odczytać znaki z zakodowanego tekstu, należy podążać po powstałym drzewie binarnym zgodnie ze wskazaniami etykiet (rysunek 4.21). Jak odczytać zakodowany ciąg? Przeanalizujmy to na przykładzie kodu 1111001100001101. Kolejne bity kodu prowadzą nas po drzewie binarnym, począwszy od korzenia znajdującego się na samej górze drzewa. Na każdym węźle (oznaczonym na rysunku 4.21 kolorem zielonym) wędrujemy zgodnie z kolejnymi bitami zapisu kodu: `1` — w prawo, `0` — w lewo. Wędrówka ta zawsze kończy się na liściu (kolor żółty). Kolejne bity to kod następnego znaku, do którego dojdziemy, ponownie począwszy od korzenia drzewa. W podanym przykładzie poszczególne bity prowadzą nas w następujący sposób: `1` — w prawo, `1` — w prawo, `1` — w prawo do znaku *VI*. Kolejne bity oznaczają następny znak: `1` — w prawo, `0` — w lewo, `0` — w lewo do znaku *IV*. Kolejne bity `11000` doprowadzą nas do znaku *I*, `01` — do znaku *VIII*, `101` — do znaku *V*. Zawsze wiadomo, kiedy rozpoczyna się kod następnego znaku. I tak otrzymamy rozkodowane znaki: *VI*, *IV*, *I*, *VIII*, *V*.

Jak widać, jest to metoda trochę trudniejsza niż stosowanie kodów o stałej długości. Zobaczmy, czy ten wysiłek się opłaca. Na płycie CD dołączonej do podręcznika, w pliku *Huffman.pdf*, został przeanalizowany ten sam plik o wielkości 1 MB składający się z ciągu znaków z alfabetu $A = \{a, b, c, d, e\}$.

Policzmy sumę iloczynów długości kodu i liczby wystąpień w pliku, czyli liczbę bitów potrzebną do zapisania skompresowanego ciągu znaków:

3*200 000+1*468 576+3*100 000+3*130 000+3*150 000 = 2 208 576 bitów, co stanowi 26,3% pliku wejściowego. Po skompresowaniu tego samego pliku za pomocą kodów o stałej długości otrzymaliśmy 37,5% pliku pierwotnego.

Kompresja stratna

Kompresja stratna znacznie zmniejsza rozmiar pliku kosztem utraty części informacji, nie dając gwarancji, że odtworzona informacja będzie identyczna z oryginałem. Można więc powiedzieć, że taka kompresja odbywa się kosztem pogorszenia jakości obrazu. Dla niektórych danych algorytm kompresji stratnej może odtworzyć informację w sposób identyczny.

Kompresja stratna jest możliwa ze względu na sposób działania ludzkich zmysłów (wyższa wartość pewnych części danych nad innymi). Algorytmy kompresji stratnej posługują się pewnymi modelami, które mają za zadanie odrzucenie najmniej istotnych danych o dźwięku czy obrazie, pozostawiając dane o wyższej wartości, pozwalające na rozpoznanie informacji przez zmysły. Liczba odrzuconych danych jest określana jako *stopień kompresji*.

Kompresję stratną stosuje się zatem do obrazów, dźwięków i oczywiście filmów. Nie można jej jednak stosować do dowolnego typu danych, na przykład plików wykonywalnych.

4.1.7. Formaty plików bitmapowych

Do zapisywania obrazów cyfrowych wykorzystywane są różne formaty, mające różne własności. Na ogół użytkownik skanera ma możliwość wyboru formatu zapisu, a w prostym amatorskim aparacie cyfrowym zazwyczaj jest dostępny jeden format, wybrany przez producenta aparatu, użytkownik zaś może jedynie wybrać jakość i wielkość zapisywanego obrazu.

BMP

Plik BMP składa się z trzech części: nagłówka (54 bajty), informacji o palecie kolorów i danych obrazowych. Ten plik nieskompresowany charakteryzuje się dużymi rozmiarami i niezmiennością wielkości pliku ze względu na zawartość obrazu. Plik będący białą kartką zajmuje na dysku tyle samo miejsca, co rysunek zamalowany wieloma detalami. Zmniejszenie rozmiaru pliku następuje przez zmianę formatu zapisu związaną ze zmniejszeniem palety barw (głębi bitowej). Pliki BMP poddawane są różnym kompresjom, w wyniku których powstają pliki bitmapowe skompresowane, takie jak: GIF, JPG, PNG, TIFF i inne.

GIF

GIF (ang. *Graphics Interchange Format* — format pliku graficznego z kompresją bezstratną) to format pliku graficznego stworzony w 1987 roku przez firmę CompuServe. Jego cechą charakterystyczną jest paleta barw ograniczona do 256 kolorów, która jest paletą indeksowaną, przypisaną (dołączoną) do obrazu. Format GIF do zapisu używa kompresji bezstratnej LZW.

Format GIF ma możliwość tworzenia w pliku prostych animacji, na przykład rejestrowania kolejnych faz ruchu, które zapisuje się jako jeden plik animowany GIF. Do tego celu doskonale nadają się cyfrowe aparaty fotograficzne mające możliwość wykonywania zdjęć sekwencyjnych, czyli rejestrowania kilku ujęć w określonym czasie.

Format GIF umożliwia również zapis informacji o przezroczystości obrazu.

JPG (DOS) lub JPEG (Unix)

JPEG (ang. *Joint Photographic Expert Group* — połączona grupa ekspertów fotograficznych) to standard kompresji statycznych obrazów bitmapowych, przeznaczony głównie do przetwarzania obrazów naturalnych (zdjęć pejzaży, portretów, zdjęć satelitarnych itp.), charakteryzujących się płynnymi przejściami barw oraz brakiem lub małą liczbą ostrych krawędzi i drobnych detali.

Kodowanie kolorów: True Color RGB pozwala na używanie ponad 16 milionów kolorów.

Kompresja JPEG jest stratna, ponieważ zmniejszenie wielkości pliku jest uzyskiwane przez usunięcie niektórych jego fragmentów, co niestety pociąga za sobą pogorszenie jakości obrazu. Kompresja JPG pozwala na zastosowanie różnych stopni kompresji. Większa wartość procentowa stopnia kompresji (czyli większa kompresja) daje nam

w rezultacie mniejszy rozmiar pliku, ale także gorszą jakość obrazu. Skala kompresji ma zakres od 1 do 100.

Kompresja pliku JPG polega na podzieleniu ilustracji na określone strefy barw układanych w kwadratowe kształty. W pliku zapisywane są raczej informacje o różnicach pomiędzy grupami pikseli niż o samych pikselach. Oddzielane są również informacje o jaskrawości i odcieniach kolorów.

Standard JPEG sam w sobie nie jest formatem plików. Definiuje on jedynie podstawy nazywane formatem wymiany JPEG. Dużą popularność uzyskał format JFIF posługujący się rozszerzeniem *.jpeg* albo *.jpg*, przez co nazywany jest błędnie JPG-iem.

JFIF

Format JFIF (ang. *JPEG File Interchange Format* — format wymiany plików JPEG) — standard zapisu plików wykorzystujący algorytmy ISO JPEG. Jego zalety to mały rozmiar pliku i duża paleta barw (24 bity), natomiast do wad można zaliczyć wyraźnie widoczne pogorszenie jakości obrazu przy dużym stopniu kompresji.

Ciekawostka

Prawdopodobnie znany Ci format JPEG jest w istocie JFIF-em, który używa nazwy JPEG i ma rozszerzenie *.jpg* lub *.jpeg*. Kiedy otworzysz plik z rozszerzeniem *.jpg* w Notatniku, to oprócz niezrozumiałych znaków (krzaczków) na samym początku dokumentu odczytasz informację o formacie pliku. A zatem czy Twoje zdjęcia to JFIF-y?

TIFF

Jest to standardowy format zapisu mapy bitowej. TIFF (ang. *Tagged-Image File Format* — znacznikowy format pliku obrazowego) umożliwia bezstratny zapis informacji o barwie obrazu cyfrowego otrzymanego podczas skanowania lub fotografowania. Często stosowany w aparatach fotograficznych, do zapisu obrazu bardzo dobrej lub najlepszej dla danego aparatu jakości zdjęć.

W formacie TIFF zapisywane są kolejno poszczególne poziomy jasności każdego piksela cyfrowego obrazu.

Format TIFF może zapisywać obrazy w postaci skompresowanej, stosując algorytm kompresji LZW. Obecnie w formacie TIFF można również zapisywać skompresowane obrazy według innych algorytmów kompresji, na przykład JPEG (stratny) lub ZIP (bezstratny), jednak nie każdy edytor grafiki potrafi poprawnie odczytać takie pliki. Kompresja TIFF jest bardziej efektywna dla obrazów z mniejszą różnorodnością detali (pikseli).

PNG

PNG (ang. *Portable Network Graphics* — przenośna grafika sieciowa) to format plików graficznych oraz system bezstratnej kompresji danych graficznych. Został opracowany

jako następca formatu GIF w 1995 roku po ogłoszeniu przez Unisys oraz CompuServe roszczeń patentowych dotyczących kompresji LZW używanej w formatach GIF oraz TIFF. Przewaga formatu PNG nad GIF-em polega na obsłudze stopniowanej przezroczystości, większej niż 8 bitów głębi koloru.

4.1.8. Grafika na stronach WWW

Kolory na stronie WWW

W kodzie źródłowym każdej strony znajdziemy informacje dotyczące ustawień kolorów. Każda strona ma swoją wizualizację i kolorystykę. Jeśli nie ustalimy tego sami, pewne rozwiązania zostaną nam narzucone (na przykład kolory odnośników).

Jak to jest z tymi kolorami? Kiedy można używać nazwy koloru, a kiedy koniecznie należy napisać kod szesnastkowy? W wielu poradnikach podawana jest rozszerzona paleta kolorów nazwanych. Jednak w języku XHTML zostało zdefiniowanych tylko 16 nazw kolorów. Użycie innych nazw jest błędem i nie gwarantuje interpretacji we wszystkich przeglądarkach. W zamian należy używać kodów heksadecymalnych (HEX). W tabeli 4.3 został podany wykaz nazw barw wraz z odpowiadającymi im kodami HEX, których można używać dowolnie, czyli podając ich nazwę lub kod.

Tabela 4.3. Ograniczona do 16 barw paleta kolorów nazwanych

Nazwa	HEX	Kolor	Nazwa	HEX	Kolor
black	#000000		green	#008000	
silver	#C0C0C0		lime	#00FF00	
gray	#808080		olive	#808000	
white	#FFFFFF		yellow	#FFFF00	
maroon	#800000		navy	#000080	
red	#FF0000		blue	#0000FF	
purple	#800080		teal	#008080	
fuchsia	#FF00FF		aqua	#00FFFF	

Definicje kolorów przydają się w takich poleceniach jak na przykład:

```
color: #00ff22;                    //Ustala kolor czcionki za pomocą stylu
background-color:#00ffff;          //Ustala kolor tła za pomocą stylu
<font color="#0000ff">             //Ustala kolor czcionki
```

```
<tr bgcolor="green">          //Określa kolor wiersza w tabeli
<td bgcolor="#cc9966">        //Określa kolor komórki w tabeli
<body bgcolor="white">        //Określa kolor tła strony
```

Elementy graficzne na stronach WWW

Definicja

Obrazek jest samodzielnym plikiem połączonym ze stroną za pomocą znacznika `<img />` lub stanowiącym tło dla strony czy tabeli.

Elementy graficzne zamieszczane na stronach WWW nie są więc ich integralną częścią. W znaczniku `<img />` zawsze podajemy nazwę pliku oraz adres wskazujący miejsce jego przechowywania. Obrazem na stronie WWW jest rysunek, zdjęcie, przycisk, wielobarwne tło, ozdobny napis, baner czy znak graficzny. Obraz może więc pełnić funkcję informacyjną, ozdobną oraz nawigacyjną. Do obrazu można zastosować odnośniki, po których kliknięciu uruchomimy zdefiniowaną akcję, zobaczymy powiększone zdjęcie, inną stronę lub inne miejsce na stronie, pobierzemy plik, wyślemy list elektroniczny…

Obrazek umieszczony na stronie zapisany jest w odpowiednim formacie graficznym. Każdy z używanych formatów ma swoje praktyczne zastosowanie. Na stronach WWW spotkać można trzy podstawowe formaty plików bitmapowych: GIF, JPG, PNG oraz SVG — format grafiki wektorowej, który można oglądać w przeglądarce po zainstalowaniu wtyczki SVG Viewer.

Kiedy użyjemy formatu GIF?

Format GIF powinien być stosowany wszędzie tam, gdzie ważne jest wierne odtworzenie kształtów, a liczba kolorów jest ograniczona. Gdy kolorów w grafice jest więcej, ich liczba zmniejszana jest automatycznie do 256, co może czasem zmienić nieco kolorystykę obrazu.

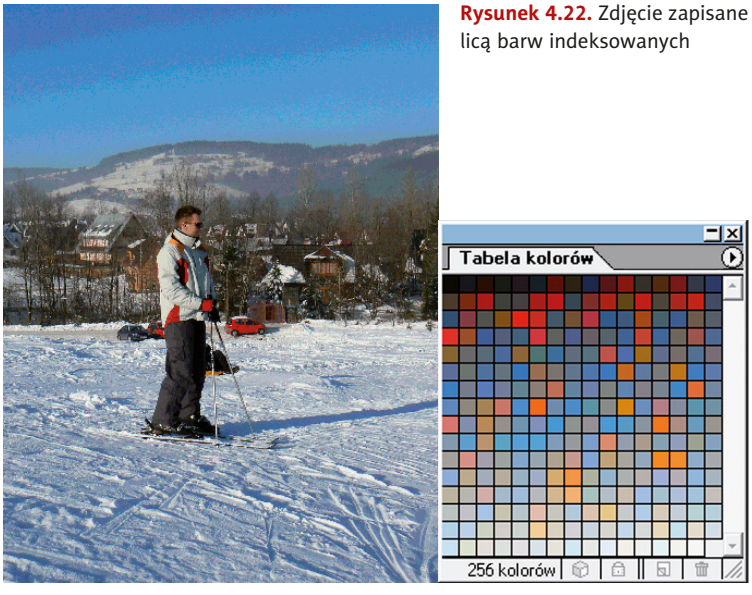
GIF pozwala również na tworzenie animacji i korzystanie z koloru „przezroczystego". Daje to możliwość stworzenia dowolnego kształtu grafiki, w którym zamiast tła obrazka będzie tło strony WWW. W ten sposób powstają na przykład strzałki czy okrągłe przyciski.

To świetny format do zapisu grafik, w których kolory można odróżnić „na oko" (gdy grafika nie zawiera przejść tonalnych, fragmentów zdjęć, dużej liczby kolorów, a także zbyt wielu szczegółów).

Gdy tworzysz animowany baner, nie masz wyjścia — pozostaje tylko GIF. W tym formacie można regulować liczbę kolorów, uzyskując tym samym mniejszy plik. Takie ograniczanie liczby kolorów może jednak prowadzić do zniekształceń barwnych, więc tutaj musisz iść na kompromis i zdecydować, co jest ważniejsze: wierne zachowanie kolorów czy mniejsza objętość pliku.

Na zdjęciu (rysunek 4.22) zapisanym w formacie GIF wystąpiły problemy z kolorami nieba. Plik ze zdjęciem zajmuje mniej miejsca, ale nie jest to bardzo duża oszczędność — zdjęcie to, zapisane w formacie JPG najlepszej jakości, czyli przy najmniejszej kom-

presji, o wymiarach 480×640 pikseli, zajmowało 380 kB. Po zapisaniu go w formacie GIF plik ma wielkość 204 kB, został więc zmniejszony o połowę. Jednak lepszy efekt otrzymamy, gdy pozostaniemy przy formacie JPG i zwiększymy stopień kompresji. Plik będzie mniejszy, stracimy również bezpowrotnie pewne informacje o kolorach, ale zachowamy przejścia tonalne zdjęcia.



Rysunek 4.22. Zdjęcie zapisane w formacie GIF wraz z tablicą barw indeksowanych

Zdjęcie w formacie GIF, pokazane na rysunku 4.22, zostało zapisane w 256 kolorach, które znajdują się w tablicy tzw. indeksowanych barw towarzyszącej obrazowi. Właśnie z tych kolorów jest utworzony ten obraz.

Ciekawostka

Kolory indeksowane — indeksowanie to umieszczenie występujących w obrazie kolorów na palecie (tablicy) kolorów indeksowanych. Każdy piksel obrazu jest indeksem (wskaźnikiem) do koloru w tablicy. Dzięki zastosowaniu indeksów kolorów plik jest mniejszy niż plik z pełnym zestawem kolorów. Indeksowanie wykorzystuje się na przykład w GIF-ach, zestaw kolorów stosowanych w danym obrazku zapisany jest w tzw. *palecie barw*, dołączonej do pliku. Paleta zawiera nie więcej niż 256 kolorów.

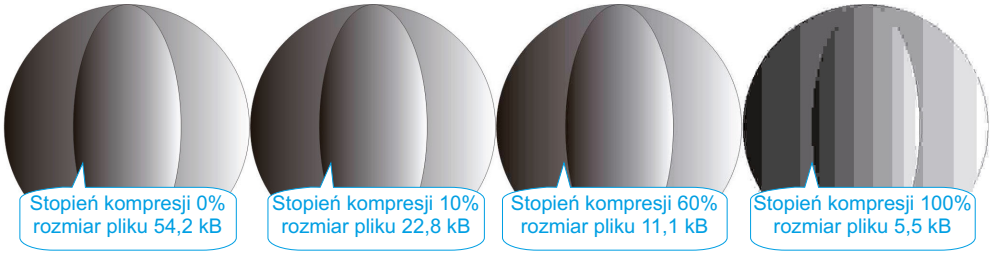
Kiedy użyjemy formatu JPG?

JPG stosowany jest wyłącznie do zdjęć i grafik, które zawierają ponad 256 kolorów i w których nie jest konieczne zachowanie wiernych kolorów i kształtów. W formacie tym kluczową rolę odgrywa stopień kompresji, od którego zależy objętość pliku. Kompresja powoduje, że plik jest zawsze gorszej jakości niż oryginał (chociaż oko może tego w niektórych przypadkach nie dostrzec).

Kompresja oparta na standardzie opublikowanym przez połączoną grupę ekspertów fotograficznych JPEG (ang. *Joint Photographic Experts Group*) jest kompresją stratną, która pozwala na znaczne zmniejszenie objętości pliku przy względnie niewielkich zmianach w jakości obrazu. Im mniejszy stopień kompresji, tym lepsza jakość zdjęcia, ale objętość pliku większa.

Najlepiej kompresuje się zdjęcia pejzaży i portretów, charakteryzujących się płynnymi przejściami barw oraz brakiem lub małą ilością ostrych krawędzi (tzw. kreski) oraz drobnych detali.

Trzeba uważać, aby nie skompresować zdjęcia za bardzo. Spróbuj zapisać pliki w formacie JPG w różnych stopniach kompresji (zawsze z oryginalnego zdjęcia). Obserwuj, jak zmieniają się jakość obrazów i rozmiary plików. Przykład zapisu pliku w różnych stopniach kompresji pokazano na rysunku 4.23.



Rysunek 4.23. Obraz w formacie JPG w kilku stopniach kompresji

Przykład bardzo dobrze oddaje sens stosowania różnych stopni kompresji formatu JPG. A co ważne, na oko nie widać większych różnic między trzema pierwszymi zdjęciami (nie widać różnic na ekranie monitora). Dalsza kompresja powoduje, że obraz jest mocno zmieniony, niemniej wciąż widać, co znajduje się na zdjęciu.

Gdy na stronie WWW jest zamieszczonych wiele zdjęć, kompresja JPG nie tylko pozwoli zaoszczędzić sporo miejsca na dysku, ale również przyspieszy wczytanie strony oraz skróci czas pobierania grafiki.

Kiedy użyjemy formatu PNG?

To najnowszy z trzech formatów grafiki rastrowej stosowanych na stronach WWW. Pozwala zachować szczegóły, korzystając z wydajniejszych metod kompresji bezstratnej.

Format PNG pozwala na tworzenie 256-stopniowej przezroczystości. Umożliwia to „wtapianie" obrazka w tło przez stosowanie realistycznych cieni i gładkich (miękkich) krawędzi obrazów wpasowujących się w tło. To bardzo przydatna opcja w tworzeniu ciekawych efektów na stronach WWW.

PNG nadaje się do zapisywania tych samych rodzajów grafik co GIF, chociaż ze zdjęciami radzi sobie o wiele lepiej od tego formatu. Dobrze zapisane grafiki PNG są prawie zawsze mniejsze od GIF-ów. Na rysunku 4.24 widać znany już obraz, zapisany w formacie PNG, w którym użyto 256 kolorów. Obraz ten zajmuje jedynie 70 kB. Jego ja-

kość jest wystarczająca do zamieszczenia go na stronie WWW. Wielkość pliku wynikowego może być różna, jest to zależne od programu, w którym zapisujemy plik, i od opcji optymalizacji obrazu.

Co zatem wybrać?

Na początku możesz mieć dylemat, który format będzie lepszy. Jeżeli masz wątpliwości, zapisz grafikę w trzech formatach i wybierz ten, który najlepiej spełnia Twoje wymagania. Gdy dojdziesz do pewnej wprawy, niemal od razu będziesz wiedzieć, który format wybrać.

Najważniejsza jest objętość pliku, więc tak dobieraj format i kompresję, aby pliki nie były zbyt duże, a jednocześnie oddawały w miarę wiernie kolory i kształty.



Rysunek 4.24. Zdjęcie zapisane w formacie PNG

Do zdjęć jest tylko jeden dobry format — JPG, tutaj nie musisz się zastanawiać, wystarczy tylko dobrać stopień kompresji. Natomiast w przypadku innych grafik pozostają GIF i PNG (szczególnie gdy chcesz wykorzystać kanał alfa — przezroczystość).

Pamiętaj też, że odbiorca, który nie widzi dwóch grafik — oryginału i wersji skompresowanej — nie zauważy różnicy. Dla niego wersja końcowa będzie dobra, ponieważ nie ma do czego jej odnieść, aby zobaczyć zmianę. Oczywiście, jeżeli kompresja będzie zbyt mocna, nawet osoba niedoświadczona stwierdzi, że zniekształcenia są bardzo widoczne.

Ta sama grafika, używana na stronie kilka razy, pobierana jest tylko raz — a więc strona z nią szybko się wczytuje. Warto też przeglądać strony WWW i podpatrywać najlepszych. Wybór formatu to zwykle kompromis — od Ciebie zależy więc, jak szybko będą pobierane graficzne fragmenty stron WWW i jak będą się prezentować w witrynie.

4.1.9. Grafika wektorowa

Grafika wektorowa to sposób opisu obrazu oparty na formułach matematycznych. W przeciwieństwie do tradycyjnej grafiki bitmapowej (opartej na zapamiętywaniu koloru i położenia pojedynczych pikseli), grafika wektorowa zachowuje informacje o obiektach, liniach prostych i krzywych — włączając w to ich położenie i barwę.

Na początku punktu 4.1.1 zaprezentowano przykład, w którym opis pikseli posłużył do symbolicznego przedstawienia obrazu bitmapowego. Spróbujmy teraz stworzyć podobną definicję grafiki wektorowej i opisać bitmapę z rysunku 4.2.

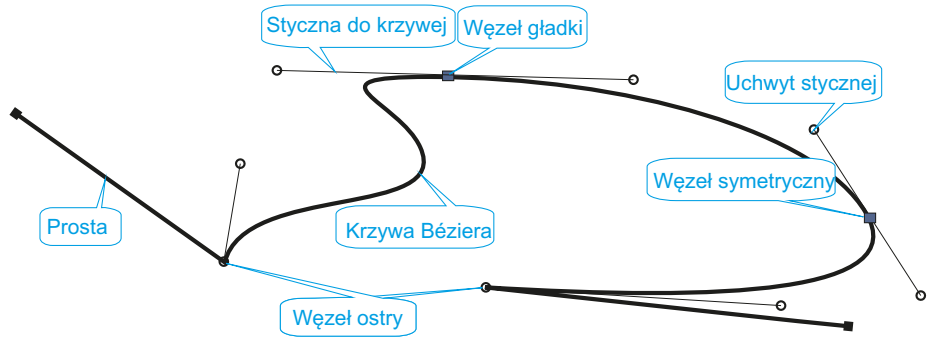
W *zapisie wektorowym* informacja o obrazie mogłaby wyglądać następująco: tło obrazu wraz z informacją o rozmiarach i kolorze, kształty (zwane prymitywami) z podaniem współrzędnych punktów, które opisują dany kształt, kolor prymitywów. Przykładowy obrazek mógłby zostać zapisany w następujący sposób:

```
Tło 10, 9, B;
Prostokąt (1,3) (2,6) N;
Prostokąt (4,1) (10,5) C;
Linia (30,10) (3,9) Ż;
Linia (10,1) (2,9) Z;
Punkt (8,7) N.
```

Zmyślone polecenia — `linia`, `prostokąt`, `punkt` — w symboliczny sposób pokazały ideę tworzenia obrazu za pomocą tzw. *prymitywów*. Każdy z takich *obiektów* można łatwo zdefiniować, podając jedynie kilka parametrów. Jest to grafika wygenerowana komputerowo.

Grafika wektorowa wykorzystywana jest głównie w rysunku technicznym, komputerowym wspomaganiu projektowania — *CAD* (ang. *Computer Aided Design*), w systemach informacji geodezyjnej — *GIS* (ang. *Geographical Information System*) oraz na stronach WWW.

U podstaw grafiki wektorowej leży linia. Może ona być prostym odcinkiem lub zdefiniowaną matematycznie linią krzywą, zwaną *krzywą Béziera*. Przykłady linii pokazano na rysunku 4.25. Zostały tam również opisane podstawowe elementy, zwane *węzłami*. Mogą one być *ostre*, *gładkie* lub *symetryczne*.

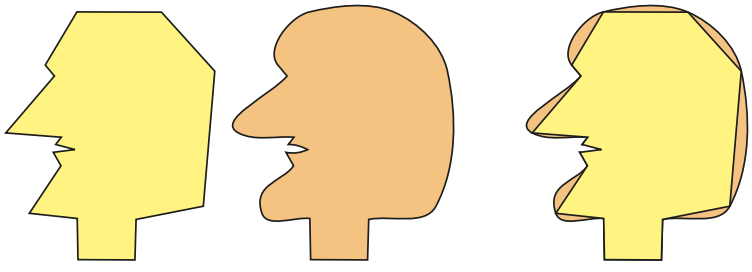


Rysunek 4.25. Linie w grafice wektorowej

Ciekawostka

Pierre Étienne Bézier (1910 – 1999) — francuski inżynier, matematyk, twórca technik krzywych Béziera i powierzchni Béziera, będących współcześnie podstawą grafiki komputerowej i projektowania wspomaganego komputerowo.

Ćwiczenie 4.8. Otwórz program do tworzenia grafiki wektorowej, na przykład zamieszczony na płycie Inkscape. Posługując się *Piórem*, czyli narzędziem do rysowania krzywych Béziera i linii prostych (*B*), narysuj łamaną. Następnie narzędziem *Edycja węzłów* służącym do edycji węzłów i uchwytów sterujących ścieżek (*N*) zaznacz odcinki *proste* i przekształć je w *krzywe*. Zmień niektóre punkty zaczepienia (węzły) z *ostrych* na *gładkie* lub *symetryczne*. Jeżeli początkowo zamieszczonych zostało zbyt mało punktów zaczepienia, dodaj nowe, dwukrotnie klikając pomiędzy istniejącymi węzłami. Chwyciwszy za *uchwyty stycznych* lub bezpośrednio za krzywe, wygnij odpowiednio linię, aby narysować kształt profilu. Wyginaj poszczególne odcinki prostych aż do otrzymania kształtu przypominającego głowę człowieka (rysunek 4.26).



Rysunek 4.26. Linie proste i krzywe w grafice wektorowej

Wszelkie linie mogą być łączone. Obszary całkowicie lub częściowo ograniczone liniami mogą mieć jednolity kolor, wypełnienie gradientowe lub mogą być wypełnione deseniami (czyli mozaiką powtarzających się regularnie wzorów). Można im też przypisać częściową przezroczystość. Możliwy jest również import bitmapy, którą po zaimportowaniu traktować można tak jak inne obiekty grafiki wektorowej.

Grafikę wektorową nazywa się również grafiką obiektową, gdyż obraz w tej grafice składa się ze stosu elementów ułożonych w odpowiedniej kolejności.

Wskazówka

Aby poznać podstawy, a także bardziej zaawansowane techniki tworzenia obrazu wektorowego, zajrzyj do doskonale przygotowanego *Podręcznika programu Inkscape* dostępnego w menu *Pomoc*. W podręczniku większość opisów oparta jest na szczegółowych i czytelnych zrzutach ekranowych. Również w menu *Pomoc* zamieszczono *Poradniki*, które wprowadzają użytkownika w świat grafiki Inkscape'a od działu *Podstawy* po *Porady i sztuczki*.

Ćwiczenie 4.9. Opracuj w grafice wektorowej projekt logo swojej szkoły lub przyszłej firmy. Do narysowania sygnetu posłuż się wieloma obiektami lub jedną łamaną linią. Na rysunku 4.27 pokazano przykładowe logo, którego sygnet narysowano właśnie taką linią. Wnętrze sygnetu wypełniono czarnym kolorem, co szczegółowo ilustruje rysunek.



Rysunek 4.27. Przykładowe logo wraz z widocznymi węzłami sygnetu

Definicja

Logo to graficzna forma przedstawienia symbolu i nazwy firmy. Terminu „logo" używamy dla całego znaku, składającego się z *sygnetu* (symbolu, znaku graficznego) i *logotypu* (tekstowego przedstawienia nazwy). Potocznie terminem „logo" określany jest także sam sygnet. Dodatkowym elementem, coraz chętniej wprowadzanym do logo, bywa hasło firmowe, tzw. *tagline*.

Grafika wektorowa jest grafiką w pełni *skalowalną*, co oznacza, że obrazy wektorowe można powiększać oraz zmieniać ich proporcje bez uszczerbku dla jakości, co pokazano na rysunku 4.28. Nie występuje zjawisko aliasingu ani powiększania pikseli. Dzieje się tak dzięki całkowicie matematycznemu opisowi elementów graficznych. Zmianie ulegną tylko importowane do pliku obiekty bitmapowe.



Rysunek 4.28. Skalowanie grafiki wektorowej

Oprogramowanie

Najpopularniejsze komercyjne programy do tworzenia grafiki wektorowej to: *CorelDRAW* (*www.corel.com*), *Adobe Illustrator* (*www.adobe.com*), *AutoCAD* (*www.autodesk.pl*), *Adobe Flash* (*www.adobe.com*) oraz bezpłatne: Sodipodi (*http://sourceforge.net/projects/sodipodi*), Xara Xtreme (*www.xaraxtreme.org*) i zdobywający coraz większą popularność Inkscape (*www.inkscape.org*) na licencji GNU General Public License (program zamieszczony na płycie CD dołączonej do podręcznika).

Formaty grafiki wektorowej

W odróżnieniu od grafiki rastrowej, istnieje bardzo niewiele uniwersalnych (zewnętrznych) formatów plików przechowujących grafikę wektorową. Praktycznie są to jedynie EPS oraz SVG. Inne formaty zachowywania grafiki wektorowej to formaty wewnętrzne

poszczególnych programów wektorowych. Coraz większą popularność zdobywa promowany przez W3C format SVG oparty na XML-u.

SVG

SVG (ang. *Scalable Vector Graphics* — skalowana grafika wektorowa) to uniwersalny format dwuwymiarowej, statycznej i animowanej grafiki wektorowej stworzony w 1999 roku przez konsorcjum W3C z myślą o zastosowaniu go na stronach WWW.

W SVG, oprócz standardowych obiektów (prostokąty, elipsy, krzywe), można opisywać efekty specjalne, takie jak: filtry, maski przezroczystości i wypełnienia gradientowe. SVG jest domyślnym formatem zapisu obrazu w programie Inkscape.

Aby używać formatu SVG, w niektórych przeglądarkach internetowych potrzebna jest odpowiednia wtyczka (na przykład SVG Viewer firmy Adobe). W niedalekiej przyszłości wszystkie przeglądarki będą obsługiwały format SVG i pozwalały na zamieszczenie tej grafiki w postaci: `<img src="obraz.svg"/>`.

EPS

Format EPS (ang. *Encapsulated PostScript* — format graficzny opisany w języku PostScript — podsumowany PostScript) służy do zapisu grafiki zarówno wektorowej, jak i bitmapowej. Jest obsługiwany przez właściwie wszystkie programy do edycji grafiki. Przez wiele lat format EPS był jedynym uniwersalnym formatem zapisu plików z grafiką wektorową (umożliwiającym również osadzanie w nim bitmap jako obiektów). Praktycznie każdy program stosowany w DTP potrafi takie pliki odczytywać i zapisywać. Każdy bitmapowy plik graficzny może zostać przekonwertowany na ten format.

Wadą używania grafik w postaci EPS jest konieczność wydruku na drukarce postscriptowej albo korzystanie ze specjalnych programów, które przekształcają PostScript do postaci zrozumiałej dla drukarki.

4.1.10. Grafika bitmapowa a wektorowa

Obrazy w grafice wektorowej można łatwo przetwarzać w obrazy bitmapowe, podając jedynie docelową rozdzielczość obrazu. Proces ten w niektórych programach nazywany jest *rasteryzacją*. Z kolei operacja przetworzenia bitmapy w grafikę wektorową (*wektoryzacja*, *trasowanie*) jest niezmiernie trudna i dotyczy tylko prostych elementów graficznych lub tworzenia obrysów wyraźnych, kontrastowych motywów obrazu bitmapowego.

Każda z odmian ma swoje zastosowanie i nieco inne przeznaczenie. Osoba projektująca grafikę komputerową powinna umieć wykorzystać wszystkie możliwości obu rodzajów grafiki i traktować je jako uzupełniające się, a nie rywalizujące. Oba sposoby zapisu informacji o obrazie mają zalety i wady, które zostały zestawione w tabeli 4.4.

Tabela 4.4. Zalety i wady grafiki wektorowej i rastrowej

Zalety grafiki rastrowej	Wady grafiki rastrowej
Można używać wielu ciekawych efektów, na przykład nakładania się obiektów. Można używać wszystkich filtrów. Można używać pędzla, gumki, narzędzi smużenia, rozmycia i wielu innych, dających ciekawe efekty. Można kolorować całe obszary lub obiekty. Mamy dostęp do pojedynczych punktów obrazu. Można szybko wykonać korektę i fotomontaż zdjęć cyfrowych. Mając zmysł artystyczny, można tworzyć obrazy tak jak przy użyciu kredki lub farby. Urządzenia wyjściowe komputera, takie jak drukarka i monitor, współpracują z grafiką rastrową.	Trudno dokonać modyfikacji — obiekty sklejają się natychmiast po ich namalowaniu. Trudno zmienić kolory szczegółów obrazu. Zapotrzebowanie na pamięć dyskową jest duże — im większy obraz, tym obszerniejszy plik. Wielkość obrazka rastrowego nie może zostać zmniejszona bez utraty jego jakości i szczegółów. Występują kłopoty z transformacjami. Występuje zjawisko aliasingu — schodkowania. Do przekształcenia obrazu są wymagane duże moce obliczeniowe. Edycja obrazu jest trudna, czasami łatwiej coś stworzyć od nowa, niż przerobić.
Zalety grafiki wektorowej	**Wady grafiki wektorowej**
Obciążenie pamięci jest niewielkie. Obraz ma wyodrębnione elementy składowe. Łatwo modyfikować poszczególne obiekty. Każdy obiekt ma własny analityczny opis na płaszczyźnie, uwzględniający: kształt, kolor i grubość linii, położenie, kolor wypełnienia oraz kolejność. Można precyzyjnie zdefiniować rozmiar każdego obiektu obrazu. Przy skalowaniu obrazu zachowane zostają jego proporcje i jakość. Istnieje możliwość komponowania obrazka po stworzeniu wszystkich obiektów. Istnieje możliwość wielokrotnego wykorzystania tych samych obiektów.	Trudno wypełnić kolorem obiekty niestandardowe. Występują problemy z zastosowaniem w innych dokumentach. Przed publikacją obraz trzeba zapisać w postaci grafiki rastrowej. Konieczność rasteryzacji obrazu do urządzeń wyjściowych (monitor, drukarka). Opis jest analityczny (kłopoty z niektórymi przekształceniami, na przykład dzieleniem obrazu, cięciem). Przeliczanie narysowanych linii i ich wygładzanie następuje automatycznie (wbrew woli autora).

Tabelę tę potraktujmy jako otwartą, czyli taką, do której możemy dopisywać swoje spostrzeżenia.

4.1.11. Grafika trójwymiarowa — 3D

Definicja

Grafika 3D to jedna z dziedzin grafiki komputerowej, zajmująca się wizualizacją obiektów trójwymiarowych i otaczającego nas świata.

Aby osiągnąć na ekranie monitora realizm generowanych obrazów w trzech wymiarach (czyli mających szerokość, wysokość oraz głębię), powierzchniom obiektów zamieszczonych na scenie nadaje się kolory, wypełnia się je wzorami, określa położenie, oświetla, dodając cienie i odbicia oraz stosując efekty przenikania, załamania czy rozpraszania światła. Bez tych wszystkich zabiegów grafika 3D nie mogłaby istnieć.

Grafika 3D znalazła już wiele praktycznych zastosowań (nie tylko w grach komputerowych).

Medycyna — modelowanie obrazów trudnych lub niemożliwych do zobaczenia (na przykład trójwymiarowy obraz z aparatury medycznej).

Projektowanie — tradycyjne biuro projektów z kreślarzami projektującymi na kalce technicznej zostało zastąpione stanowiskami komputerowymi, na których powstają projekty, makiety, wersje zagospodarowania terenów czy wnętrz. Powstają modele domów, pokoi, mebli, sieci elektrycznych i kanalizacyjnych, samochodów.

Kino — film z efektami trójwymiarowymi pozwolił na stworzenie zjawisk, efektów i obiektów niespotykanych w rzeczywistości, takich jak biegające stworki, świetlne miecze, prehistoryczne zwierzęta i postaci.

Techniki tworzenia grafiki 3D

Istnieje wiele metod tworzenia grafiki trójwymiarowej — od najprostszych grafik wektorowych typu szkieletowego, przez obrazy składające się z wypełnionych jednolitym kolorem ścian, aż po bardziej zaawansowane techniki, dzięki którym powstają obrazy dające wrażenie bardziej realnych niż fotografie.

Modeler

Program umożliwiający tworzenie i modyfikowanie obiektów trójwymiarowych za pomocą zestawu narzędzi, a także zbioru podstawowych figur (prymitywów), takich jak: prostopadłościan, kula, walec czy stożek, które można wykorzystać do budowania sceny. Obiekty 3D są przeważnie budowane z siatek wielokątów lub definiowane za pomocą krzywych parametrycznych. Siatka wielokątów ma tę przewagę, że łatwo można deformować obiekt (skręcać, wyginać), z kolei krzywe parametryczne w naturalny sposób tworzą gładkie powierzchnie.

Radiosity

Metoda wywodząca się z badań nad promieniowaniem cieplnym, niezależna od położenia obserwatora. Wyznacza globalny rozkład natężenia światła, uwzględniając pochłonięcia i odbicia światła, jakie mają miejsce na wszystkich powierzchniach znajdujących się na scenie. Nie uwzględnia natomiast efektów świetlnych zależnych od położenia obserwatora, takich jak rozbłyski na powierzchniach czy załamanie światła.

Ray tracing

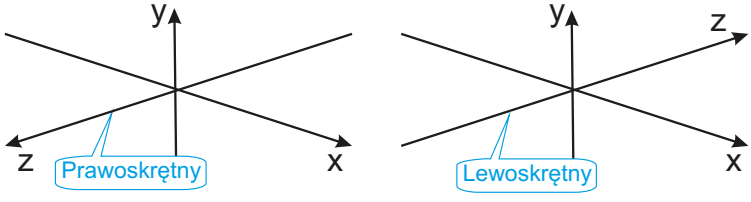
Śledzenie promieni — technika tworzenia fotorealistycznych scen 3D. Rysunki stworzone tą techniką są efektowne i dość łatwe do odróżnienia. Promień światła tworzy obraz. Dzięki programom do ray tracingu powstają również bardzo ładne animacje.

Geometria trójwymiarowa

Praca z grafiką trójwymiarową nie jest możliwa bez geometrii w trzech wymiarach. Podstawową umiejętnością jest określenie i odczytanie położenia punktu w układzie współrzędnych oraz umiejętność wyobrażenia sobie widoku obiektu w różnych położeniach

w przestrzeni. Dowolny punkt w grafice 3D można opisać za pomocą trójwymiarowego układu współrzędnych składającego się z poziomej osi x, pionowej y i prostopadłej do nich osi głębokości z. Środek układu znajduje się na przecięciu wszystkich trzech osi i ma współrzędne (0,0,0).

Ze względu na zwrot osi z możemy się spotkać z układem współrzędnych *lewoskrętnym* lub *prawoskrętnym* — oba typy zilustrowano na rysunku 4.29. Skąd wzięły się takie nazwy? Spróbuj ułożyć trzy palce jednej dłoni w taki sposób, jak są ułożone trzy osie. Zauważysz z pewnością, że uda się w taki sposób ustawić tylko jedną z dłoni: lewą dłonią układ lewoskrętny, prawą prawoskrętny. Najczęściej w grafice 3D wykorzystywany jest układ lewoskrętny, w którym oś z jest skierowana w głąb przestrzeni.



Rysunek 4.29. Układy współrzędnych w grafice 3D

Z czego składa się trójwymiarowa scena?

Scenę trójwymiarową możemy porównać z kosmosem — nieograniczoną przestrzenią, w której panuje ciemność. Wyobraź sobie, że przebywasz w ciemnym pomieszczeniu. Co można w takiej sytuacji zrobić, aby cokolwiek zobaczyć? Z pewnością zapalić przynajmniej jedno źródło światła (*światło*). Następnie otworzyć oczy (*kamerę*) i skierować je w stronę obiektów znajdujących się w pomieszczeniu. Kamera musi mieć zdefiniowane przynajmniej dwa parametry: *położenie*, czyli miejsce, z którego patrzy obserwator, oraz punkt, na który kamera jest skierowana — na który patrzą oczy obserwatora. Kamera obserwuje tylko fragment sceny, ponieważ jest ograniczona *kątem widzenia* (jak ludzkie oczy bądź obiektyw aparatu fotograficznego).

Wirtualne *źródło światła* może być zarówno punktem świetlnym świecącym we wszystkich kierunkach, jak żarówka lub słońce, jak i światłem kierunkowym, takim jak światło latarki lub reflektor.

Obiekty to umieszczone na scenie i opisane odpowiednimi modelami matematycznymi elementy sceny, takie jak: kula, prostopadłościan, walec, stożek czy płaszczyzna, które są określane przez pewne cechy: położenie, rozmiar i pigment wypełnienia powierzchni.

POV-Ray

Jest wiele programów do tworzenia obrazów metodą śledzenia promieni, na przykład Vivid, PolyRay, POV. Skupimy się na jednym z nich, mianowicie POV-Ray, zamieszczonym na płycie CD dołączonej do podręcznika (*povwin362-32bit.msi* lub *povwin362-64bit.msi*). Nazwa tego programu jest skrótem nazwy angielskiej: Persistance of Vision Ray Tracer. Program został stworzony przez zespół programistów (w dokumentacji wymienionych jest aż 17 autorów) i jest jednym z najlepszych i najbardziej uniwersalnych pro-

gramów wykorzystujących technikę śledzenia promieni (ray tracing), na dodatek należy do grupy open source (choć licencja na używanie i rozpowszechnianie programu jest nietypowa i skomplikowana). Te fakty powodują, że jest on popularny, bardzo dobrze przetestowany i ciągle rozwijany. Wersje POV-Ray są dostępne dla większości platform sprzętowych, od Amigi, przez Apple i PC, aż po maszyny uniksowe.

Do zalet programu POV-Ray należy zaliczyć to, że potrafi on w czasie tworzenia obrazu wyświetlać go na ekranie komputera. Może także kontynuować przerwane obliczenia od miejsca, gdzie je zakończył. Jeśli dodać jeszcze, że pliki wejściowe opisujące tworzone obrazy są czystymi plikami ASCII, a jednocześnie pewnym rodzajem programu, to już wiadomo, dlaczego wybór padł właśnie na ten program.

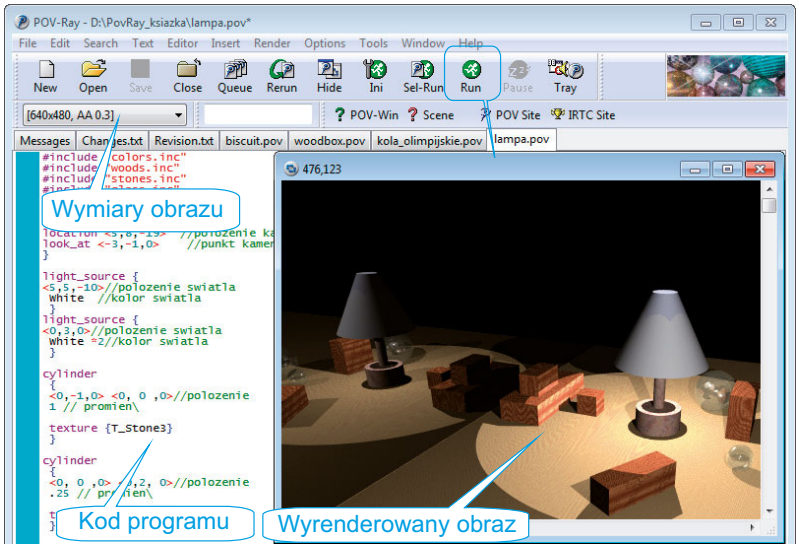
Jak pracuje POV-Ray? Na wejściu dostaje plik tekstowy, który opisuje świat, jaki chcemy ujrzeć, a na wyjściu, po długotrwałych obliczeniach, otrzymujemy plik graficzny zawierający wygenerowany obraz. Plik wejściowy POV-Ray składa się z ciągu instrukcji, które opisują obiekty zawarte w tworzonym świecie, jest to więc pewien rodzaj języka programowania, trochę podobny do języka C (z powodu znacznej różnicy w zakresie realizowanych zadań jest to podobieństwo bardzo powierzchowne).

Program składa się z bloków, z których każdy opisuje jakiś obiekt tworzonego świata. Takim obiektem może być bryła umiejscowiona w przestrzeni, ale także źródło światła oświetlającego scenę, dzięki któremu zobaczymy obraz. W tym języku nie ma poleceń opisujących takie obiekty jak punkt czy prosta, można je przedstawić jedynie jako małą kulkę, sześcianik czy długi, cienki walec. Jest to więc język opisu obiektów fizycznych, a nie matematycznych.

Wszystkie przytoczone poniżej przykłady tworzenia obrazu 3D zostały wykonane w programie POV-Ray. Wykorzystuje on metodę ray tracingu oraz lewoskrętny układ współrzędnych. Wygląd sceny opisuje się językiem poleceń, w którym podawane są informacje o kamerze, oświetleniu oraz obiektach (bryłach) i powierzchniach.

Po uruchomieniu programu obsługa jest stosunkowo prosta, chociaż menu nie jest spolszczone. Rysunek 4.30 przedstawia widok okna programu POV-Ray. Niezbędne polecenia to *New* i *Run*. *Run* **renderuje** obraz i zapisuje na dysku bitmapę, którą oglądamy na pierwszym planie. Program wymaga zapisu pliku z kodem przed uruchomieniem. Utworzony obraz otrzymuje taką nazwę, jaką nosi plik źródłowy. Każde uruchomienie tworzy nowy obraz, który jest nadpisywany na poprzednim. Zatem przykładowy plik źródłowy *kula.pov* po dodaniu kolejnych kul zapisz jako *trzy_kule.pov* i dopiero wtedy renderuj obraz.

W POV-Ray mamy do dyspozycji kilka podstawowych obiektów — są to:
- kula (*sphere*),
- stożek (*cone*),
- walec (*cylinder*),
- prostopadłościan (*box*),
- pierścień (*torus*),
- płaszczyzna (*plane*),



Rysunek 4.30. Widok ekranu w programie POV-Ray

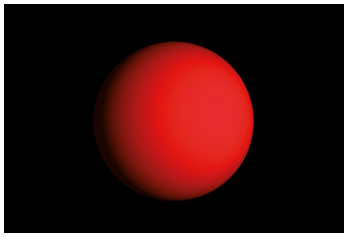
- krzywa obrotowa (*lathe*, *SOR*),
- wielokąt (*polygon*),
- kropla (*blob*).

Wszystkie obiekty muszą mieć określone położenie, wielkość oraz wypełnienie.

Oto przykładowy opis sceny w programie POV-Ray (*kula.pov*):

```
#include "colors.inc" //Biblioteka kolorów
camera { //Tylko jedna kamera (obiekt obowiązkowy)
 location <0, 1, -10> //Położenie obserwatora — kamery <x, y, z>
 look_at <0,0,0> //Punkt, na który skierowany jest obiektyw kamery
}
light_source { //Źródło światła punktowego — musi być przynajmniej jedno
 <5, 2, -10> //Położenie źródła światła
 White //Kolor światła (nazwa koloru, pisana zawsze wielką literą)
}
sphere { //Kula
 <0,0,0> //Położenie środka kuli
 2 //Promień kuli, może być zapisany jako 2 lub 2.0
 pigment {color Red} //Kolor powierzchni — uwaga na kolor!
}
```

Z kodu źródłowego renderujemy obraz przyciskiem *Run* i otrzymujemy plik w formacie BMP, pokazany na rysunku 4.31.



Rysunek 4.31. Kula — obraz wyrenderowany w programie POV-Ray (kula.pov)

Wskazówka

Ponieważ plik obrazu domyślnie jest zapisywany w tej samej lokalizacji, w której został zapisany plik źródłowy, i przyjmuje nazwę pliku źródłowego, przed wydaniem polecenia *Run* musimy zapisać plik z kodem źródłowym na dostępnym nośniku danych.

Dodaj jeszcze dwie kule. W tym celu skopiuj obiekt *sphere* i zmień położenie środków kul oraz ich kolory. Zapisz grafikę jako nowy plik i dopiero wtedy renderuj obraz.

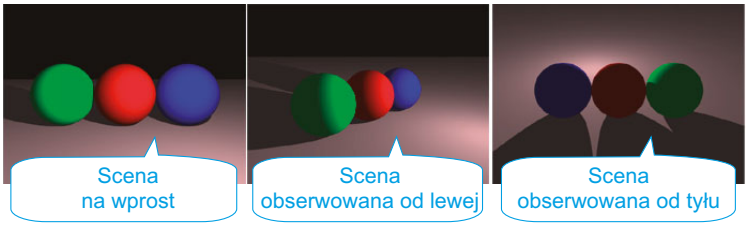
Definicja

Renderowanie (od ang. *rendering* — przedstawianie) to proces przeliczania danej sceny w celu utworzenia pliku wyjściowego w formie obrazu statycznego lub animacji.

Renderowanie jest bardzo czasochłonną operacją, niewymagającą, poza przygotowaniem, żadnej ingerencji ze strony osoby tworzącej obraz. Podczas tych operacji wyliczane są nie tylko obrazy zamieszczonych na scenie obiektów, ale i odbicia światła, jego załamania, cienie, mgła itp.

Kamera i źródła światła mogą zmieniać swoje położenie. Scenę możemy oglądać z dowolnej strony. Kamera może być umieszczona bliżej lub dalej od obserwowanych obiektów, a światło oświetlające scenę może przybierać różne postaci. Każda zmiana niesie ze sobą bardzo ciekawe efekty.

Różne sposoby oświetlania i ich efekty na obrazach pokazano na rysunku 4.32. Obserwator patrzy na scenę z różnych stron: z przodu, z boku i z tyłu — pod światło. Uzyskasz te efekty, zmieniając położenie kamery (*location*).



Rysunek 4.32. Przykłady obserwowania sceny w programie POV-Ray (3kule_z_przodu.pov, 3kule_z_lewej.pov, 3kule_z_tylu.pov)

W tworzeniu sceny bardzo pomocne jest menu *Insert*. Jeśli wybierzesz podmenu *Shapes*, zobaczysz na podglądzie tworzone kształty. Po wstawieniu wybranego obiektu do kodu źródłowego dodawany jest, w postaci komentarza, krótki opis poszczególnych składowych. Musisz jedynie mieć wyobraźnię i znać podstawy geometrii.

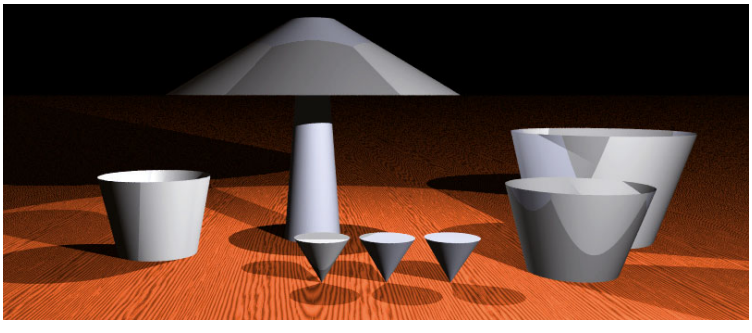
Ćwiczenie 4.10. Wykorzystaj powstały wcześniej plik *kula.pov* i wymień zamieszczoną na scenie kulę na inne obiekty. Początkowo zamieszczaj na scenie pojedyncze obiekty: prostopadłościan (*box*), który wymaga podania położenia w przestrzeni dwóch leżących po przekątnej narożników, walec (*cylinder*) mający dwie podstawy (podajesz położenie ich środków) i promień, stożek (*cone*), którego składnia jest następująca:

```
cone { //Stożek
 <0, 0, 0>, 3 //Środek dolnej podstawy i jej promień
 <0, 5, 0>, 0.5 //Środek górnej podstawy i jej promień (gdy wartość promienia jest równa 0,
                //to otrzymasz klasyczny stożek)
 open //Powstanie stożek pusty w środku (pominięcie tego zapisu da nam zamkniętą bryłę)
 pigment {hexagon Silver, White, Gray} //Powierzchnia bryły pokryta sześciokątną
                                       //mozaiką w trzech kolorach
}
```

Dodaj podłogę, na którą będzie rzucany cień.

```
plane { //Płaszczyzna
 y, 0 //położona prostopadle do osi y i przecinająca ją w punkcie 0
 texture {T_Wood19} //Zdefiniowane drewniane tekstury. Wstaw w nagłówku odwołanie do
                    //biblioteki #include „woods.inc"
}
```

Poeksperymentuj ze współrzędnymi stożka. Zapisuj kolejne pliki i twórz kolejne ciekawe obrazy. Przykładową kompozycję stożków pokazano na rysunku 4.33.



Rysunek 4.33. Różne położenia stożka na scenie (stozki.pov)

Zamieszczone obiekty możemy poddać procesowi przekształcenia — **transformacji**. Zarówno w grafice 3D, jak i w 2D istnieją trzy podstawowe typy przekształceń (transformacji):

Translacja — zmiana położenia obiektu, przesunięcie każdego z wierzchołków o ten sam wektor.

Obrót — obrót obiektu wokół jednej z trzech osi (w zależności od oprogramowania, mierzony jest w radianach lub stopniach).

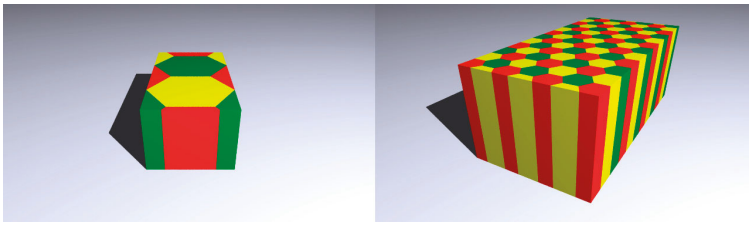
Skalowanie — zmiana rozmiaru obiektu. Może być jednorodne (z zachowaniem proporcji obiektu) i niejednorodne (proporcje się zmieniają — skalowanie wykonane względem każdej osi w różnych skalach), na przykład kula rozciągnięta w jednej osi.

Oto przykład zastosowania transformacji w programie POV-Ray (*obrocony_box.pov*):

```
box { //Prostopadłościan
 <-1, 0, -1> //Lewy tylny dolny wierzchołek prostopadłościanu
 < 1, 1.5, 3> //Prawy przedni górny wierzchołek prostopadłościanu
```

```
    pigment {hexagon Yellow, Green, Red scale <0.2, 0.5, 0.2>}
    // Wypełnienie powierzchni heksagonalnym wzorem z zastosowaniem skalowania wzoru innego w każdej osi
    rotate y*30 //Obrót bryły o kąt 30 stopni względem osi y (można również zapisać to jako obrót
                //o wektor: rotate <0,20,0>
    scale 1.5 //Skalowanie — zmiana wszystkich wymiarów o jednakową liczbę
}
```

Przykładowe obrazy powstałe na skutek transformacji kolorowego pudełka widać na rysunku 4.34 — tu skalowany był prostopadłościan (`scale 1.5`) oraz wzór pokrywający powierzchnię (`scale <0.2, 0.5, 0.2>`). Dokonano tam też obrotu (`rotate y*30`).

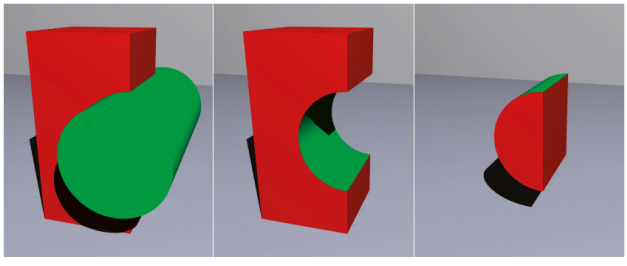


Rysunek 4.34. Przykład zastosowania obrotu i skalowania w programie POV-Ray (nieobrocony_box.pov, obrocony_box.pov)

Konstruktywna geometria brył (CSG) na przykładzie programu POV-Ray

Możliwe jest tworzenie złożonych brył za pomocą operacji teoriomnogościowych na innych bryłach (tzw. *Constructive Solid Geometry* — CSG; konstruktywna geometria brył). Podstawowe działania na zbiorach, znane z lekcji matematyki, to suma, różnica oraz iloczyn dwóch zbiorów. Jeżeli bryłę umieszczoną na scenie potraktujemy jak zbiór punktów, zrozumienie działań na bryłach nie będzie sprawiało kłopotu.

Rysunek 4.35 ilustruje trzy działania (dodawanie, odejmowanie i część wspólna) prostopadłościanu i walca. Opis ich wykonania zamieszczono na płycie CD dołączonej do podręcznika, w pliku *konstruktywna geometria bryl.pdf*.



Rysunek 4.35. Dodawanie, odejmowanie i część wspólna brył

Wraz z grafiką trójwymiarową pojawiło się wielkie zapotrzebowanie na moc obliczeniową, a rozwój gier komputerowych wymusił na producentach zmiany prowadzące do uzyskania płynnego i realistycznego obrazu. Pojawiły się nowe karty grafiki, nowe magistrale (PCI, AGP, PCI-Ekspress), nowe układy elektroniczne przyspieszające obliczenia. Obecnie w kartach grafiki stosowany jest wyspecjalizowany procesor graficzny GPU (ang. *Graphics Processing Unit*), odpowiedzialny za tworzenie i przetwarzanie grafiki trójwymiarowej w komputerze. Głównym jego zadaniem jest przejęcie wykonywanych obliczeń i odciążenie głównego procesora komputera (CPU).

4.1.12. Zadania

Zadanie 4.1. Zapoznaj się z interfejsem programu POV-Ray. Zorganizuj scenę zgodnie z poniższymi poleceniami.

a) Kamerę umieść na wprost sceny i skieruj ją w stronę punktu będącego środkiem układu współrzędnych.

b) Źródło światła punktowego umieść po prawej stronie kamery, ponad nią.

c) Na środku sceny umieść kulę w kolorze czerwonym, tak by jej środek pokrywał się z początkiem układu współrzędnych.

d) Po obu stronach kuli ustaw dwie kule tego samego rozmiaru, ale w innych kolorach, na przykład zielonym i niebieskim.

e) Dodaj poziomą płaszczyznę, tak aby kule leżały na niej.

f) Obserwuj scenę, zmieniając położenie kamery. Spójrz na kule od lewej i prawej strony oraz z góry i z tyłu.

g) Przed środkową kulą umieść czerwone pudełko, które leży na podłodze i nie zasłania kul.

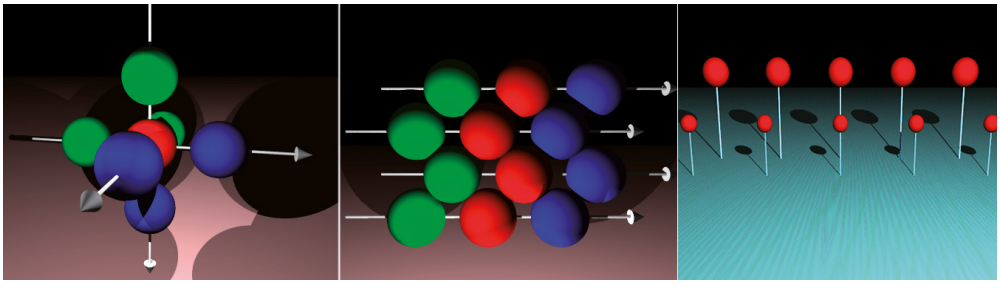
h) Na środkowej kuli połóż walec, na walcu — stożek.

i) Dodaj dwie pionowe ściany.

j) Dopasuj położenie kamery i oświetlenia, aby dobrze zaprezentować wszystkie obiekty na scenie.

k) Plik wynikowy powstały podczas renderowania przekonwertuj z formatu BMP na formaty: PNG, GIF oraz JPG, wykorzystując do tego edytor systemowy Paint lub edytor obrazów GIMP. Za każdym razem do konwersji użyj oryginału BMP.

Zadanie 4.2. Na rysunku 4.36 zamieszczono wyrenderowane obrazy zaprojektowanej sceny. Wykonaj podobne projekty. Kolory wypełnień i proporcje elementów dobierz według własnego uznania.



Rysunek 4.36. Przykładowe obrazy do zadania 4.2

Wskazówka

Jeśli na scenie chcemy zamieścić powtarzające się elementy, położone w różnych miejscach, możemy utworzyć *obiekt*, który później umieścimy w odpowiednich miejscach sceny. Przykład takiego rozwiązania podano w pliku *lizak.pov* zamieszczonym na płycie CD dołączonej do podręcznika.

Zadanie 4.3. Wykonaj projekt wnętrza kolorowego pokoju dziecinnego, w którym umieścisz proste, barwne meble, takie jak: stolik, tapczanik, fotel, dywan, półka na zabawki, lampa stojąca. Zastosuj działania na obiektach: obrót, skalowanie i przesunięcie.

Zadanie 4.4. Wykonaj projekt, który imituje budowlę z drewnianych klocków. Zastosuj działania dodawania, odejmowania i mnożenia brył.

Wskazówka

Definiując obiekt, można pominąć pokrycie jego powierzchni. Odpowiednią teksturę można przypisać przy rysowaniu danego obiektu. Przykład takiego rozwiązania podano w pliku *klocek.pov* zamieszczonym na płycie CD dołączonej do podręcznika.

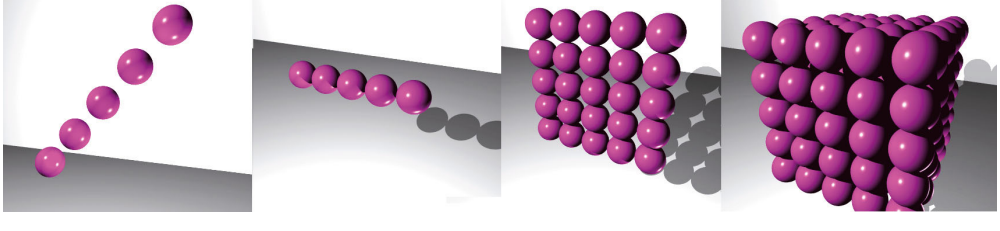
Zadanie 4.5. Wykonaj projekt kartki świątecznej. Umieszczane na scenie obiekty poddaj różnym transformacjom. (Przykładowe rozwiązanie zadania pokazano na rysunku 4.37). Poziomą powierzchnię pokryj wzorem, który będzie imitował śnieg, zgodnie z podanym przykładem zamieszczonym na płycie CD, w pliku *snieg.pov*.



Rysunek 4.37. Kartka świąteczna — obraz wyrenderowany w POV-Ray

Zadanie 4.6. Ustaw na scenie ludzika z kasztanów lub z klocków lego.

Zadanie 4.7. Do stworzenia formacji wielu elementów warto zastosować konstrukcję pętli. W programie POV-Ray jest to możliwe. Na rysunku 4.38 przedstawiono kilka przykładów wyrenderowanych obrazów z zastosowaniem w programie pętli `while`. Przeanalizuj konstrukcję pętli i sposób deklaracji zmiennych, a następnie wykonaj kompozycje zamieszczone na rysunku. Kolorystykę dobierz według własnego



Rysunek 4.38. Przykłady zastosowania pętli w grafice 3D

uznania. Zwróć uwagę na położenie podłogi i ustaw ją tak, aby kule nie znalazły się pod jej poziomem. Oto kod fragmentu programu, w którym deklarujemy obiekty, zmienne i używamy konstrukcji pętli:

```
#declare kula=sphere { //Deklaracja jednego obiektu o nazwie kula
 <0,0,0> //Położenie środka kuli
 .5  //Promień kuli 0.5 lub (skrócony zapis) .5
 pigment {color Magenta}  //Kolor powierzchni
 finish {phong .8}  //Odblask światła na powierzchni
}
#declare i=-2; //Deklaracja zmiennej i o wartości początkowej -2
#while (i < 3) //Dopóki ten warunek jest prawdziwy, wykonuj pętlę while
  object {kula translate x*i translate y*i } //Obiekt kula przesunięty
                                             //o wektor <i,i,0>
 #declare i=i+1; //Zwiększenie i o 1
#end //Koniec pętli while
```

Zadanie 4.8. Obraz ma wymiary 22×5 pikseli i wypełniony jest trzema kolorami: czerwonym (C), niebieskim (N) i zielonym (Z). Przedstaw zapis tego obrazu, stosując metodę kodowania RLE.

C	C	N	Z	Z	Z	N	N	N	C	C	N	N	N	Z	Z	N	C	C	Z	Z	N
N	N	C	C	N	Z	Z	N	N	C	C	C	N	N	Z	Z	N	C	C	C	N	N
C	C	C	N	Z	Z	Z	N	N	N	C	C	N	Z	Z	Z	N	C	C	N	N	Z
N	N	N	C	C	C	Z	Z	C	C	N	Z	C	N	N	Z	Z	Z	C	C	N	N
Z	Z	C	C	Z	Z	C	C	N	C	C	Z	Z	N	N	N	C	C	C	Z	Z	Z

Zadanie 4.9. Wykorzystując słownikową metodę LZW, zakoduj obraz przedstawiony w poprzednim zadaniu. Rozwiązaniem tego zadania jest słownik oraz pięć linii numerów kodów elementów tego słownika.

Zadanie 4.10.

1,1,2,2,3,3,6,4,5,6,2;	Słownik:
4,6,8,2,4,3,14,6;	1.Z, 2.N, 3.C, 4.ZZ, 5.ZN, 6.NN, 7.NC, 8.CC, 9.CN, 10.NNZ, 11.ZZZ,
8,18,19,1,8,23,6;	12.ZNN, 13.NNN, 14.ZZN, 15.NNC, 16.CCN, 17.NZ, 18.ZZC,
21,24,19,25,27,2;	19.CZ, 20.ZZNN, 21.CCZ, 22.ZZCC, 23.CZZ, 24.ZC, 25.CCC,
13,23,30,25,4;	26.CZZN, 27.CCZZ, 28.ZCC, 29.CZC, 30.CCCC, 31.CCZZN,
15,17,12,24,13,4	32.NNNC, 33.CZZC, 34.CCCCC, 35.CCCZ, 36.NNCN, 37.NZZ,
7,10,38,9,42;	38.ZNNZ, 39.ZCN, 40.NNNZ, 41.NCN, 42.NNZZ, 43.ZNNZC,
44,37,10,45,4.	44.CNN, 45.CNNN, 46.NZZN, 47.NNZC, 48.CNNNZ.

Jakiego koloru jest strzałka na rysunku, którego osiem wierszy pikseli zapisano za pomocą podanego powyżej słownika LZW?

Zadanie 4.11. Na płycie CD dołączonej do podręcznika znajduje się plik tekstowy *obrazek.txt*, w którym w pięciu wierszach są podane kolory poszczególnych pikseli obrazu (jak w zadaniu 4.8). Kolory oznaczone zostały literami C, N, Z. Każdy kolejny znak oddzielono od poprzedniego spacją. Napisz program, który zakoduje podany w pliku „obrazek", wykorzystując słownikową metodę LZW. Kod źródłowy programu zapisz pod nazwą *lzw.pas* lub *lzw.cpp*.

Rozwiązaniem tego zadania jest wypisany na ekranie słownik oraz pięć linii numerów wykorzystanych elementów powstałego słownika.

Zadanie 4.12. Obrazek z poprzednich zadań został zmieniony. Dołożono do niego jeszcze dwa kolory: fioletowy (F) i różowy (R). Zatem teraz obraz składa się z pięciu barw: C, F, N, R, Z (w porządku alfabetycznym). Uwzględniając liczbę wystąpień każdego koloru w obrazie, utwórz kod Huffmana, a następnie za pomocą cyfr 0 i 1 przedstaw kody, które przypiszesz poszczególnym kolorom. Zapisz w pięciu wierszach ciągi 0 i 1 utworzonego kodu Huffmana odpowiadające poszczególnym kolorom obrazu.

Aby sprawnie zliczyć wszystkie kolory za pomocą napisanego przez siebie programu lub z wykorzystaniem funkcji arkusza kalkulacyjnego, możesz skorzystać z pliku tekstowego *obraz.txt*, który zawiera oddzielone spacją litery odpowiadające kolorom. Plik ten znajduje się na płycie CD dołączonej do podręcznika.

C	C	N	Z	Z	Z	N	N	N	C	C	N	N	N	Z	Z	N	C	C	Z	Z	N
N	N	C	C	N	Z	Z	N	N	C	C	C	F	F	F	Z	N	C	C	C	N	N
C	C	C	N	Z	Z	Z	N	N	N	C	C	F	F	F	Z	N	C	C	N	N	Z
N	N	R	R	C	C	Z	Z	C	C	N	Z	C	N	N	Z	Z	Z	C	C	N	N
Z	Z	R	R	Z	Z	C	C	N	C	C	Z	Z	N	N	N	C	C	C	Z	Z	Z

4.2. Dźwięk

Tylko niewielki procent informacji z otaczającego świata dociera do nas poprzez narząd słuchu. Słysząc, jak gra fortepian lub skrzypce, możemy powiedzieć, że słyszymy, jak dociera do nas dźwięk. Do powstania dźwięku potrzebne są dwa składniki: źródło i przewodnik dźwięku. Aby można było usłyszeć dźwięk, konieczne jest sprawne ucho słuchającego. Jeśli wyeliminujemy chociaż jeden z tych składników, dźwięku nie usłyszymy.

4.2.1. Teoria dźwięku

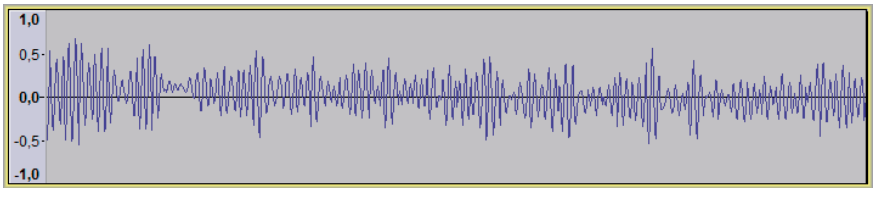
Dźwięk to fala akustyczna rozchodząca się w danym ośrodku sprężystym. Aby można było słyszeć dźwięki, muszą być spełnione określone warunki: musi istnieć źródło dźwięku, które wytwarza falę dźwiękową, a fala dźwiękowa musi mieć możliwość rozchodzenia się. Rozchodzenie się fali dźwiękowej jest możliwe dzięki ośrodkowi (najczęściej jest to powietrze). Prędkość rozchodzenia się dźwięku jest różna w różnych ośrodkach i przykładowo wynosi:

- woda — 1500 m/s,
- powietrze — 340 m/s,
- stal — 5100 m/s,
- szkło — 6000 m/s,
- beton — 3800 m/s,
- próżnia — 0 m/s.

Okazuje się, że w powietrzu dźwięk rozchodzi się stosunkowo wolno, bardzo szybko w metalu i szkle, natomiast w ogóle nie rozchodzi się w próżni, tam panuje cisza.

Prawie wszystkie sygnały występujące w otaczającym nas świecie są *analogowe*. Fala akustyczna dociera do mikrofonu i jest zamieniana na sygnał elektryczny. Sygnał ten wiernie odwzorowuje falę akustyczną, jest więc *sygnałem analogowym*. Na rysunku 4.39 pokazano przykładowy przebieg elektryczny ludzkiego głosu.

Opisując falę dźwiękową, skupimy się na dwóch podstawowych parametrach: *częstotliwości* oraz *natężeniu*.



Rysunek 4.39. Głos człowieka widziany na ekranie oscyloskopu

Człowiek słyszy dźwięki o częstotliwościach w zakresie od 16 Hz do 20 000 Hz. Im większa częstotliwość dźwięku, tym wyższy *ton* dźwięku. Zakres częstotliwości słyszanych przez człowieka można podzielić na trzy podstawowe grupy:

- **tony niskie:** 16 – 300 Hz,
- **tony średnie:** 300 – 3000 Hz,
- **tony wysokie:** 3000 – 20 000 Hz.

Najlepiej słyszymy dźwięki o częstotliwości od 1000 do 5000 Hz, czyli częstotliwości, które są charakterystyczne dla ludzkiego głosu. Zdolność słyszenia dźwięków jest zależna od wielu czynników, wśród których podstawowy to wiek człowieka. Im jesteśmy starsi, tym gorzej słyszymy wysokie częstotliwości. Również słuchanie głośnej muzyki oraz niektóre przebyte choroby i przyjmowane leki powodują pogorszenie słyszalności wysokich częstotliwości, nawet u młodych ludzi. Górna granica słyszalności wysokich częstotliwości może się więc naturalnie obniżyć do 16 kHz, a nawet do 10 kHz.

Drgania akustyczne, których częstotliwość jest tak mała, że nie są słyszalne, nazywamy **infradźwiękami** (niższe niż 20 Hz), zaś dźwięki, których częstotliwość jest tak duża, że również nie są słyszalne, nazywamy **ultradźwiękami** (powyżej 20 000 Hz).

Dźwięk, rozchodząc się w danym ośrodku, niesie ze sobą pewną energię, będącą *natężeniem dźwięku*. Jest to średnia ilość energii przepływającej w czasie 1 s przez jednostkowe pole powierzchni 1 m^2, mierzona w W/m^2. Dla lepszej prezentacji natężenia dźwięku wprowadzono częściej używaną jednostkę *decybel* (*dB*), będącą względną, logarytmiczną jednostką natężenia dźwięku. Choć decybel jest jedną dziesiątą bela (B), to właśnie decybel (dB), a nie bel, jest powszechnie używany.

Ciekawostka

Różnica 3 dB oznacza dwukrotnie większy hałas. Na przykład jeżeli w sklepie widzimy dwie lodówki, które na etykietach mają informacje, że jedna wytwarza hałas 57 dB, a druga 54 dB, to oznacza, że ta druga jest dwa razy cichsza od pierwszej.

Ciekawostka

Alexander Graham Bell (1847 – 1922) — na jego cześć nazwano jednostkę poziomu natężenia dźwięku: bel (B). W Bell Laboratories, już po śmierci Grahama Bella, wynaleziono między innymi system operacyjny Unix, tranzystor, światłowód i laser.

4.2.2. Zapis dźwięku

Dźwięk możemy zapisać metodą analogową lub cyfrową.

Zapis analogowy dźwięku

Dźwięk analogowy jest wytwarzany przez naturalne źródła dźwięku w przyrodzie. *Sygnałami analogowymi* nazywamy takie sygnały elektryczne, które są odpowiednikami określonych wielkości fizycznych dźwięku (częstotliwość, chwilowa amplituda…).

Pierwszego zapisu dźwięku za pomocą urządzenia zwanego *fonografem* dokonał w 1877 roku Amerykanin Thomas Alva Edison. Drgająca membrana, igła żłobiąca spiralny rowek w obracającym się wałku pokrytym metalową folią i wreszcie tuba głosowa, z której wydobywał się dźwięk nagrania — to bardzo skrócony opis tego wielkiego wynalazku.

Później dźwięk analogowy był zapisywany na różnych nośnikach i różnymi metodami:
- Płyta gramofonowa — zapis i odczyt mechaniczny.
- Taśma i kaseta magnetofonowa — zapis i odczyt magnetyczny.
- Taśma filmowa 24 mm — zapis i odczyt optyczny.

Cyfrowy zapis dźwięku

W 1976 roku dwie duże firmy — Philips oraz Sony — opracowały założenia płyty CD. Jej premiera odbyła się w Langenhagen 17 sierpnia 1982 roku — to był początek dźwięku cyfrowego.

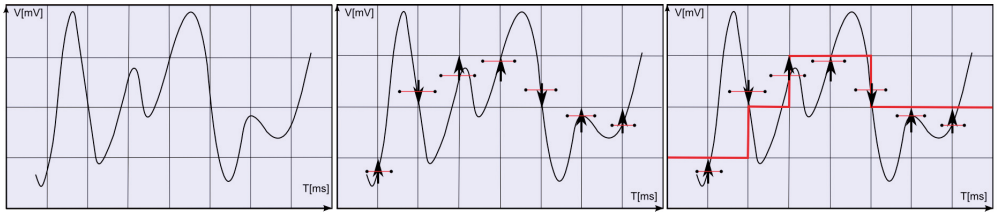
Konwersja sygnału analogowego na cyfrowy

Aby zrozumieć, w jaki sposób działa płyta CD, należy przeanalizować sam proces przetwarzania sygnału analogowego na cyfrowy. Proces ten nazywamy *konwersją AC*. Polega on na pomiarze wartości chwilowej sygnału w równych odstępach czasu. Proces ten to *próbkowanie*. Następnie *chwilowa wartość* próbkowanego sygnału jest zaokrąglana do najbliższej wartości *poziomu kwantyzacji*.

Obecnie w każdym komputerze jest karta dźwiękowa. Do niej podłączone są głośniki oraz mikrofon. Gdy mikrofon jest aktywny, następuje w nim zamiana docierających do niego dźwięków na sygnał analogowy. Ten po dotarciu do karty dźwiękowej jest zamieniany na sygnał cyfrowy. Proces zamiany sygnału analogowego na cyfrowy pomogą zrozumieć ilustracje na rysunku 4.40.

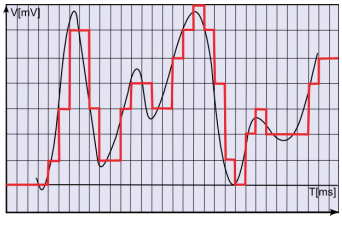
Linie pionowe na wykresie sygnału analogowego symbolizują *próbkowanie*, czyli sprawdzanie wielkości sygnału próbkowanego co pewien określony czas (stała częstotliwość próbkowania).

Linie poziome na wykresie to *poziomy kwantyzacji*. W omawianym przypadku jest ich bardzo mało. Sam proces kwantyzacji polega na zaokrągleniu chwilowej wartości próbkowanego sygnału do najbliższej wartości poziomu kwantyzacji wskazanej strzałkami.



Rysunek 4.40. Sygnał analogowy, poziomy jego kwantyzacji i sygnał cyfrowy uzyskany przy małej częstotliwości próbkowania (linia czerwona)

Sygnał cyfrowy narysowany linią czerwoną na rysunku 4.40, który otrzymaliśmy, stosując niską częstotliwość próbkowania oraz małą liczbę poziomów kwantyzacji, bardzo odbiega od oryginału. Aby uzyskać lepszy efekt, podczas procesu AC (konwersja analogowo-cyfrowa) należy zwiększyć częstotliwość próbkowania oraz liczbę poziomów kwantyzacji. W ten sposób otrzymamy sygnał bardziej zbliżony do oryginalnego brzmienia. Tę różnicę widać na rysunku 4.41. Sygnał cyfrowy jest bardziej zbliżony do oryginału analogowego.



Rysunek 4.41. Sygnał cyfrowy uzyskany przy większej częstotliwości próbkowania i większej liczbie poziomów kwantyzacji

Oba procesy: próbkowanie ze stałą częstotliwością oraz kwantyzacja odgrywają kluczową rolę w procesie zamiany sygnału analogowego na cyfrowy i decydują o jakości wynikowego dźwięku cyfrowego.

Im większa częstotliwość próbkowania sygnału analogowego oraz im więcej wprowadzonych poziomów kwantyzacji, tym dokładniej można opisać sygnał analogowy metodą cyfrową.

Jednak zbyt duża częstotliwość próbkowania oraz zbyt wiele poziomów kwantyzacji może doprowadzić do powstania nadmiernej ilości danych reprezentujących sygnał analogowy, co oczywiście jest niekorzystne ze względu na wielkość pliku wynikowego. Trzeba znaleźć złoty środek, a więc taką wartość częstotliwości próbkowania oraz liczbę poziomów kwantyzacji, aby sygnał cyfrowy zadowalająco opisywał sygnał analogowy, a wielkość pliku wynikowego nie była za duża.

Kryterium Nyquista mówi, że częstotliwość próbkowania sygnału analogowego powinna być minimum dwa razy większa od największej częstotliwości sygnału próbkowanego. Spełnienie tego kryterium daje pewność uniknięcia zniekształceń sygnału cyfrowego.

Jeśli chcemy przekonwertować sygnał dźwiękowy bez zniekształceń, musimy sobie zdawać sprawę, że największą częstotliwością, którą człowiek może usłyszeć, jest częstotliwość dźwięku w granicach 20 kHz. Dlatego minimalna częstotliwość próbkowania sygnału analogowego powinna wynosić 40 kHz (dwa razy większa od 20 kHz). Liczba poziomów kwantyzacji w standardzie CD została ustalona na 16 bitów. Wartość ta pozwala na odwzorowanie dynamiki na poziomie 90 dB. Ważną bowiem cechą wszystkich formatów dźwięku jest ich dynamika.

Dynamika dźwięku to wartość wyrażająca zakres dzielący najcichszą i najgłośniejszą próbkę materiału. Dynamika ludzkiego słuchu wynosi 120 dB, co oznacza, że zakres pomiędzy najcichszym i najgłośniejszym dźwiękiem, jaki możemy usłyszeć, wynosi 120 dB. Dynamika możliwa do uzyskania w przypadku materiału odtwarzanego z płyty CD wynosi zaledwie 90 dB. Dynamika sygnału odtwarzanego z płyty DVD wynosi aż 144 dB, czyli więcej, niż może zarejestrować ludzkie ucho.

Wskazówka

Płyty CD nigdy nie czyścimy, wykonując ruchy okrężne. Takie postępowanie może spowodować powstanie rysy, która uniemożliwi odczyt danych z płyty. Płytę CD czyścimy zawsze od środka do zewnątrz.

Dla standardu CD ustalono częstotliwości próbkowania na **44,1 kHz** oraz **16 bitów** dla procesu kwantyzacji, **stereo**. Pliki wynikowe w tym standardzie mają rozszerzenie *.wav* i są kodowane metodą bezstratną, która powoduje, że pliki zajmują dużo miejsca (około 172 kB/s dla jakości CD). Standardowo na jednej płycie CD można zapisać około 74 minut muzyki stereo.

Ciekawostka

Prędkość obrotu płyty musi gwarantować odczyt tej samej ilości danych w ciągu sekundy (około 172 kB/s). Płyta CD podczas odczytu musi zmieniać swoją prędkość obrotową. Podczas zmiany odsłuchiwania skrajnych utworów na płycie napęd CD musi znacznie zwolnić lub przyspieszyć. Jest to główna przyczyna długiego oczekiwania na utwór. Prędkość liniowa płyty jest stała, zmianie ulega prędkość obrotowa.

Długość ścieżki zapisanej na płycie kompaktowej wynosi około 5 – 6 km, a dla DVD aż 11 km.

Częstotliwość próbkowania (kHz) oraz liczba poziomów kwantyzacji (bity) może być różna w zależności od pożądanej jakości sygnału wynikowego. Stosowane obecnie częstotliwości próbkowania zostały zestawione w tabeli 4.5.

Tabela 4.5. Częstotliwości próbkowania dźwięków

Zastosowanie	Częstotliwość próbkowania
Telekomunikacja	8,0 kHz (mono)
Cyfrowe radio i telewizja	32,0 kHz lub więcej
Płyty CD audio	44,1 kHz/16 bitów (stereo)
Telewizja HD	48,0 kHz, dźwięk przestrzenny 7+1
DVD audio	44,1 kHz, 48 kHz, 88,2 kHz, 96 kHz, 176,4 kHz lub 192 kHz, w zależności od formy zapisu dźwięku (mono, stereo, przestrzenny 5+1)
Blu-ray	Jak w przypadku telewizji HD

Jak widać na tym zestawieniu, częstotliwość próbkowania sygnału jest zależna od jego przeznaczenia, co oczywiście ściśle wiąże się z jakością zapisanego dźwięku. W telekomunikacji, w której głównie mamy do czynienia z przesyłaniem ludzkiego głosu, częstotliwość próbkowania jest mała, a jakość słaba. Dźwięk o najlepszej jakości usłyszymy, odtwarzając płytę DVD lub oglądając telewizję HD.

4.2.3. Formaty plików dźwiękowych

WAVE (ang. *waveform audio format*)

To format plików dźwiękowych stworzony przez Microsoft oraz IBM. Odpowiednikiem WAVE na platformach Apple jest AIFF. Pliki WAVE mają rozszerzenie *.wav*. Według specyfikacji pliki WAVE mogą zawierać zarówno zapis skompresowany, jak i nieskompresowany. W tym sensie WAVE jest więc tzw. kontenerem przeznaczonym do przechowywania sygnału audio, który może być zakodowany na różne sposoby i przez różne programy kodujące (kodeki). W praktyce spotyka się głównie pliki WAVE z zapisem nieskompresowanym zakodowanym algorytmem PCM (ang. *Pulse Code Modulation* — modulacja kodowo-impulsowa), tak samo jak sygnał na płytach CD. Z tego powodu w potocznym rozumieniu bardzo często identyfikuje się pliki WAVE tylko z ich nieskompresowanym wariantem.

Nieskompresowane pliki WAVE mają relatywnie duże rozmiary, w porównaniu zarówno z plikami stosującymi kompresję bezstratną, jak i — tym bardziej — z plikami stosującymi kompresję stratną. Mimo wszystko pliki WAVE cieszą się dużą popularnością. Decyduje o tym ich rozpowszechniona obsługa przez popularne oprogramowanie komputerów, prostota struktury pliku i łatwość jego edycji.

MP3 (MPEG-1/2 audio layer-3)

Format MP3 został stworzony w Instytucie Fraunhofera Zintegrowanych Układów Elektronicznych w 1991 roku. Jest formatem stratnej kompresji dźwięku używającym modelu psychoakustycznego. Początkowo do kompresji MP3 stosowano taką samą liczbę bitów

przypadającą na dany przedział — *Constant Bit Rate* (CBR). Obecnie podczas kodowania częściej korzysta się ze zmiennej przepływności — *Variable Bit Rate* (VBR). Kompresowany fragment strumienia dźwięku poddawany jest analizie i wraz ze zmianami strumienia zmieniają się parametry kompresji podczas kodowania.

Definicja

Bitrate — przepływność, czyli szybkość transmisji, szybkość przesyłu (przepływu) danych przez kanał telekomunikacyjny lub urządzenie — jest kluczowym parametrem plików MP3. Określa liczbę bitów danych użytych do zapisu i transmisji dźwięku przez jednostkę czasu (b/s lub częściej kb/s). To od niego głównie zależy jakość skompresowanego dźwięku. Im wyższa przepływność, tym lepsza jakość dźwięku, ale większy rozmiar pliku. Najczęściej stosowaną wartością przepływności jest 128 kb/s — takie jest domyślne ustawienie wielu programów kodujących. Bitrate odnosi się do plików multimedialnych, muzycznych oraz strumienia danych przesyłanych w sieci, na przykład radia internetowego.

Aby utworzyć plik MP3, potrzebny jest plik źródłowy w formacie WAV. Można go uzyskać przez „zrzucenie" (*ripowanie*) ścieżki audio z płyty CD lub samodzielne nagranie dźwięku w dowolnym edytorze audio.

Jest wiele programów, które przetwarzają utwory z płyt audio CD na formaty WAV i MP3. Znajdziesz w nich informacje na temat metody *bitrate*. Na przykład prosty w obsłudze systemowy *Windows Media Player* wykorzystuje kilka metod zmiennej przepływności, które pozwalają na zmienny bitrate od 128 do 320 kb/s (kbit/s, kbps). Dlatego zanim wybierzesz polecenie *Zgraj dysk CD*, wybierz utwory do zgrania i ustal odpowiedni format w *Ustawieniach zgrywania*. W przypadku wyboru formatu MP3 wybierz odpowiednią jakość audio.

Definicja

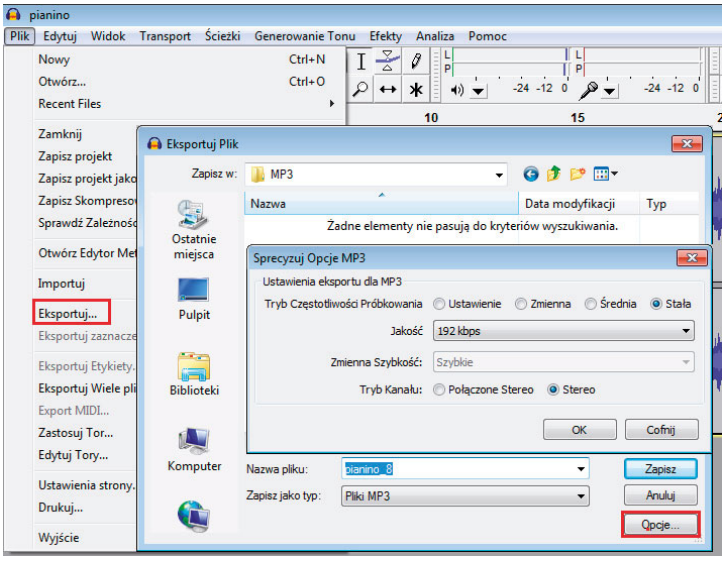
Enkoder to program kompresujący dźwięk do danego formatu.

Enkodery MP3 mają za zadanie zmniejszenie pliku przez obcięcie częstotliwości na przykład spoza granicy słyszalności.

Jakość wynikowego dźwięku zależy w dużym stopniu od algorytmu kompresji. Wykonując ćwiczenie 4.11, zbadamy wpływ parametrów kompresji na wielkość oraz jakość pliku wynikowego w formacie MP3.

Ćwiczenie 4.11. W programie Audacity (program na licencji GNU — *General Public License*), którego plik instalacyjny znajduje się na płycie CD dołączonej do podręcznika (*audacity-win-unicode-1.3.12.exe*), otwórz także znajdujący się na płycie CD plik *pianino.wav*. Wysłuchaj nagrania.

Przekonwertuj plik *pianino.wav* na format MP3 zgodnie z parametrami wynikowych plików MP3, dla których specyfikacja zamieszczona została w tabeli 4.6. Poleceniem z menu *Plik/Eksportuj…* zapisz w nowym folderze o nazwie *MP3* pliki wynikowe *pianino_8.mp3*, *pianino_32.mp3*, *pianino_64.mp3*, *pianino_128.mp3* i *pianino_192.mp3*, wybierając odpowiednie opcje, jak na rysunku 4.42. Podczas eksportowania kolejnych plików pamiętaj, że plikiem źródłowym zawsze musi być oryginalny plik *pianino.wav*. Nie należy kom-



Rysunek 4.42. Ustawienia opcji wynikowych plików w formacie MP3 w programie Audacity

presować pliku już skompresowanego do formatu MP3. Powtórz to ćwiczenie dla pliku z nagraniem ludzkiej mowy — *klamczucha.wav*.

Uzupełnij tabelę 4.6 informacjami dotyczącymi wielkości powstałych plików. Wysłuchaj zapisanych przez siebie plików, zwracając uwagę na jakość nagrania. Wyciągnij wnioski i przedstaw swoją opinię na temat jakości nagrań. Zastanów się, gdzie mogą być wykorzystywane pliki o najgorszej jakości. Zauważ, że ćwiczenie wykonywane było na pliku z nagraniem samego instrumentu i ludzkiej mowy bez instrumentów. Dodatkowo oba nagrania wykonano metodą amatorską — w tle słychać szum i hałas. Nie są to utwory muzyczne nagrane w profesjonalnym studiu nagraniowym, jednak już tutaj dostrzegalne są różnice w jakości wynikowego dźwięku.

Tabela 4.6. Parametry wynikowych plików w formacie MP3 powstałych z kompresji pliku WAV

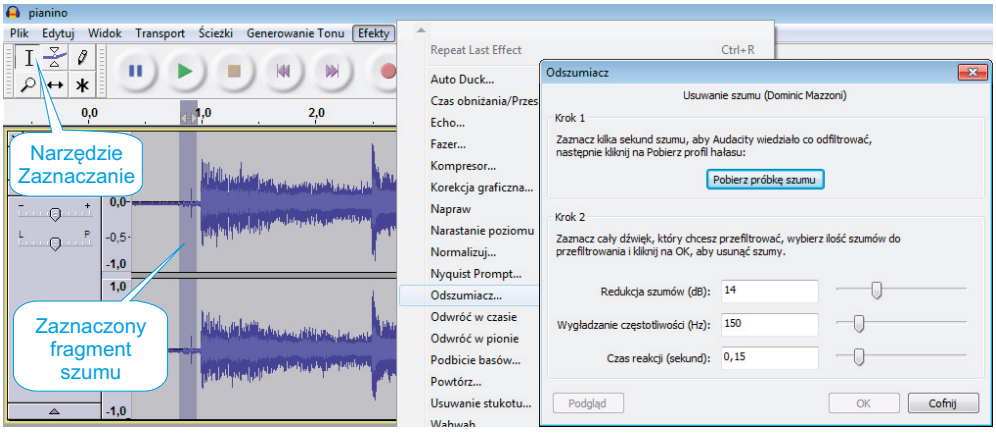
Nazwa pliku źródłowego (wynikowego)	Parametry plików stereo		Wielkość pliku	Uwagi (subiektywne odczucie jakości)
	Częstotliwość próbkowania	Bitrate		
pianino.wav (*klamczucha.wav*)	44,1 kHz	—		
pianino_192.mp3 (*klamczucha_192.mp3*)	44,1 kHz	192 kb/s		
pianino_128.mp3 (*klamczucha_128.mp3*)	44,1 kHz	128 kb/s		
pianino_64.mp3 (*klamczucha_64.mp3*)	44,1 kHz	64 kb/s		
pianino_32.mp3 (*klamczucha_32.mp3*)	44,1 kHz	32 kb/s		
pianino_8.mp3 (*klamczucha_8.mp3*)	22,05 kHz	8 kb/s		

Zauważ, że przy wyborze opcji wynikowego pliku MP3 istniała możliwość wyboru wielu różnych parametrów. Każdy człowiek słyszy inaczej. Są osoby, które mają bardzo wrażliwy słuch, i takie, które słyszą nieco gorzej dźwięki o wysokich częstotliwościach. Można więc przygotować pliki MP3 z zakupionych płyt, dostosowując jakość do własnych wymagań.

Nagrany przy użyciu amatorskiego mikrofonu dźwięk zawiera szum. W programie Audacity podejmiemy próbę jego usunięcia. Jeśli wskażemy szum i go zapamiętamy (zdefiniujemy), to korzystając z odpowiedniego filtru, możemy usunąć z oryginału te częstotliwości, które wchodzą w skład uprzednio zapamiętanej próbki.

Jeśli to możliwe, trzeba odszukać tę część nagrania, w której występuje szum zamiast ciszy, i zaznaczyć ten niewielki fragment nagrania w celu zdefiniowania próbki szumu. Następnie, korzystając z odpowiedniego filtru, należy usunąć zapamiętane częstotliwości szumu.

Ćwiczenie 4.12. Przeprowadź odszumienie znanego z poprzedniego ćwiczenia pliku *pianino.wav*. W tym celu zlokalizuj i zaznacz fragment nagrania — ten, w którym najgłośniej słychać szum zamiast ciszy, jak na rysunku 4.43. Następnie w menu *Efekty* wybierz *Odszumiacz…* i wykonaj krok pierwszy — *Pobierz próbkę szumu*.



Rysunek 4.43. Pierwszy etap usuwania szumu z dźwięku

Następnie zaznacz cały utwór (*Ctrl+A*) i ponownie wybierz z menu *Efekty/Odszumiacz…*. W kroku drugim ustaw suwakiem małą wartość *Redukcja szumów* (na przykład 14 dB) i potwierdź decyzję przyciskiem *OK*. Operację tę można powtarzać wielokrotnie, starając się uzyskać jak najlepszy efekt, jednak zbyt duży poziom odszumienia może spowodować efekt „metalicznego" dźwięku. Należy więc znaleźć najlepsze rozwiązanie, słuchając nagrania po kolejnych etapach odszumiania.

Na tradycyjnej płycie CD najczęściej jest nagranych od kilkunastu do nieco ponad 20 utworów muzycznych, z których każdy trwa od 3 do 5 minut. Nagrywając ulubione utwory również na CD, ale w formacie MP3 lub jeszcze wydajniejszym WMA, zmieścimy na jednej płycie, zależnie od zastosowanego stopnia kompresji, ponad 100 piosenek.

Będą one całkiem dobrej jakości. Jeśli zdecydujemy się na umieszczenie swoich kolekcji na płycie DVD, zdołamy nagrać na jeden krążek zawartość ponad 7 płyt CD. To jest aż 700 – 800 utworów. Stosunkowo niewielka objętość „empetrójek" powoduje, że mogą być one przesyłane przez sieci komputerowe i rozpowszechniane w internecie. W tym tkwi siła formatu MP3.

Najpoważniejszą wadą MP3 są ograniczenia licencyjne. Opatentowanie algorytmów powoduje, że niemożliwe jest tworzenie urządzeń wykorzystujących pliki MP3 bez uprzedniego uiszczenia opłaty licencyjnej. Opłata licencyjna ukryta jest więc w cenie sprzętu.

Algorytm kompresji MP3 można — w wielkim skrócie — przedstawić następująco:

- podział sygnału wejściowego na ramki,
- filtracja sygnału,
- analiza psychoakustyczna,
- normalizacja sygnału,
- kwantyzacja sygnału,
- kodowanie i tworzenie sygnału wyjściowego.

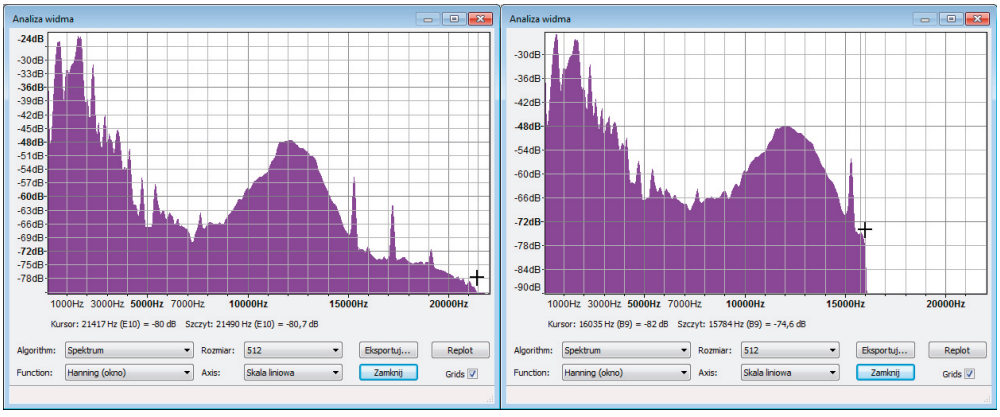
Ćwiczenie 4.13. Przeprowadzimy analizę widma plików dźwiękowych WAV i MP3 z ćwiczenia 4.11. Do wykonania tego ćwiczenia ponownie użyjemy programu Audacity. Otwórz plik *pianino.wav*. Zaznacz fragment dźwięku (na przykład od 3. do 14. sekundy utworu) i z menu *Analiza* wybierz *Wykres spektrum...*. W oknie dialogowym *Analiza widma* zmień sposób wyświetlania widma na skalę liniową. Można odczytać, że analizowany fragment dźwięku *pianino.wav* zawiera częstotliwości w zakresie od 86 Hz do ponad 21 kHz. Plik *klamczucha.wav* ma nieco inne zakresy.

Kolejno otwórz wygenerowane przez siebie pliki MP3 i przeanalizuj ich widma pod kątem występujących częstotliwości. Postaraj się zaznaczać ten sam fragment utworu. Odczytane wartości umieść w tabeli 4.7. Ćwiczenie to możesz wykonać na innym zamieszczonym na płycie CD nagraniu lub na własnym utworze nagranym w rejestratorze dźwięku.

Tabela 4.7. Analiza widma sygnału dźwiękowego plików WAV i MP3

Nazwy plików (źródłowego, wynikowych)	Parametry plików stereo		Pasmo (zakres częstotliwości)	
	Częstotliwość próbkowania	Bitrate	Minimalna wartość	Maksymalna wartość
pianino.wav	44,1 kHz	—	86 Hz	ponad 21 kHz
pianino_192.mp3	44,1 kHz	192 kb/s		
pianino_128.mp3	44,1 kHz	128 kb/s		
pianino_64.mp3	44,1 kHz	64 kb/s		
pianino_32.mp3	44,1 kHz	32 kb/s		
pianino_8.mp3	22,05 kHz	8 kb/s		

Na rysunku 4.44 zestawiono analizę widm plików *pianino.wav* oraz *pianino_128.mp3*. Widać wyraźnie, że widmo pliku MP3 nie zawiera wysokich częstotliwości. Czy słuchając obu utworów, odczuwasz tę różnicę?



Rysunek 4.44. Analiza porównawcza widm sygnału dźwiękowego pianino.wav (po lewej) i pianino_128.mp3 (po prawej) w programie Audacity

Wyciągając wnioski z wykonanych ćwiczeń, udziel odpowiedzi na następujące pytania:

1. Czy jakość dźwięku jest powiązana z brakującymi częstotliwościami w widmie danego pliku MP3?
2. Czy konwersja pliku MP3 na standard WAV poprawi jego jakość?
3. Jaką jakość będzie miał plik WAV utworzony z MP3?

OGG

Jedynym sposobem ominięcia ograniczeń patentowych formatu MP3 jest stworzenie alternatywnych formatów kompresji i udostępnienie ich na zasadach wolnego oprogramowania. Takim algorytmem jest *OGG* — format plików dźwiękowych zbliżony do standardu MP3.

Jest to bezpłatny format *kontenera strumieni danych*, rozwijany przez fundację Xiph.org. Kontener OGG może zawierać wiele *strumieni dźwięków, obrazów i napisów*.

Pliki dźwiękowe mają rozszerzenie *.ogg*.

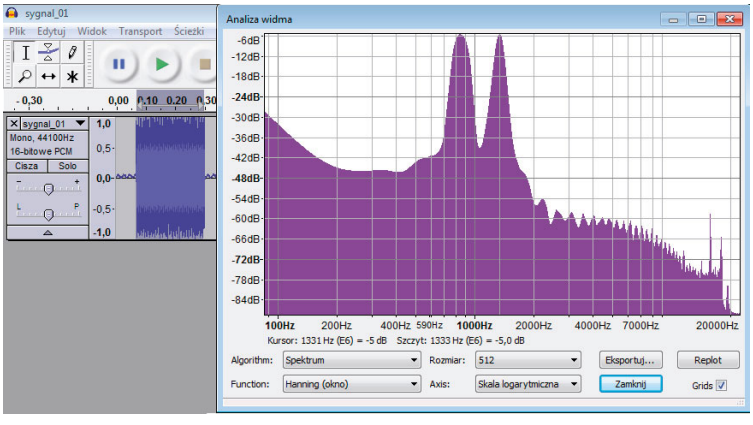
OGG to także rodzina *kodeków*, czyli *algorytmów kompresji-dekompresji*.

Pliki OGG powoli zdobywają zwolenników. Istnieją już strony internetowe, z których można pobrać legalne i pozbawione ograniczeń licencyjnych pliki OGG. Nowy format kompresji dźwięku nie wyprze zapewne „empetrójek", choć jest tego wart — pliki OGG zapewniają nieco lepszą jakość, przy czym są mniejsze od analogicznych plików MP3.

Ćwiczenie 4.14. Do wykonania tego ćwiczenia ponownie wykorzystamy program Audacity. Wyszukaj w internecie informacje na temat sygnalizacji tonowej używanej w urządzeniach telefonicznych, czyli *kodowania DTMF* (ang. *Dual Tone Multi Frequency* — dwutonowe wybieranie wieloczęstotliwościowe). Zwróć uwagę na dwie częstotliwości przyporządkowane odpowiednim przyciskom telefonu. Przykładowo cyfrze 5 przyporządkowano dwie częstotliwości: 770 Hz i 1336 Hz. Otwórz w programie Audacity plik *5.ogg* znajdujący się na płycie CD, w folderze *DTMF_pliki*, i zobacz jego

analizę widma. Tym razem lepiej oglądać wykres w skali logarytmicznej. Gdy zbliżysz wskaźnik myszy do miejsca, w którym jest szczyt, pod wykresem odczytasz dokładną wartość częstotliwości. Zawsze są dwie szczytowe wartości. Odczytaj obie i porównaj z tabelą częstotliwości tonów DTMF. W tym przypadku odczytanymi częstotliwościami są 1334 Hz i 786 Hz, co pozwala jednoznacznie wskazać w tabeli cyfrę 5. Sprawdź jeszcze dwie cyfry z plików *2.ogg* i *8.ogg*.

Otwórz jeden z plików znajdujących się na płycie CD dołączonej do podręcznika, w folderze *Numery_tel*. Możesz wykorzystać pliki w formacie WAV lub OGG. Przeprowadź analizę widma, zaznaczając kolejne fragmenty sygnału wybieranego numeru telefonu. Do ćwiczenia użyto wszystkich możliwych przycisków cyfr, liter i znaków. Na rysunku 4.45 pokazano widok logarytmiczny analizy widma pierwszej wybranej cyfry nagrania *sygnal_01.ogg*. Na wykresie wyraźnie widoczne są dwa szczyty, które pozwalają na rozszyfrowanie wybieranego tonowo numeru telefonu. Zidentyfikuj numer telefonu, analizując kolejne próbki i porównując je z danymi z tabeli kodowania DTMF.



Rysunek 4.45. Analiza widma sygnału tonowego pokazanego w skali logarytmicznej

AAC

Jest to format stratnej kompresji danych dźwiękowych. Format AAC (ang. *Advanced Audio Coding* — zaawansowane kodowanie dźwięku) zaprojektowany został w 1997 roku jako następca MP3. Oferuje lepszą jakość dźwięku przy podobnym rozmiarze danych. Algorytm jest skuteczniejszy i wydajniejszy od MP3, charakteryzuje się również lepszym przenoszeniem częstotliwości — ponad 16 kHz. Parametry AAC to: próbkowanie 8 – 96 kHz oraz możliwość zapisu do 48 kanałów, podczas gdy w standardzie MP3 — tylko do 2 kanałów.

FLAC

FLAC (ang. *Free Lossless Audio Codec*) to kodek do bezstratnej kompresji sygnału audio. Oznacza to, że inaczej niż w przypadku formatów takich jak MP3, WMA czy OGG pliki w formacie FLAC zawierają pełną, oryginalną informację. Jeśli plik audio zostanie skompresowany za pomocą formatu FLAC, to następnie można go zdekompreso-

wać i przywrócić do pierwotnej postaci, tak by każda kolejna próbka zachowała swoją oryginalną wartość. Jakość odtwarzania skompresowanych za pomocą FLAC-a plików dźwiękowych jest wyższa aniżeli jakość plików skompresowanych stratnie, takich jak MP3, WMA czy OGG.

Skuteczność kompresji jest zależna od sygnału poddawanego obróbce. Pliki skompresowane są zwykle około dwa razy mniejsze niż oryginały.

FLAC jest bezpłatny, a jego specyfikacja jest otwarta. We FLAC-u nie są wykorzystywane żadne rozwiązania chronione patentami, nie ma żadnych zabezpieczeń przed kopiowaniem. Istnieją wersje kodeka na wszystkie najpopularniejsze systemy operacyjne, jak: Windows, Linux, OS X czy Solaris.

4.2.4. Zadania

Zadanie 4.13. W programie Audacity wykonaj podstawowe czynności edycyjne na pliku dźwiękowym. Wykorzystaj jeden z plików dźwiękowych zamieszczonych na płycie CD (na przykład *improwizacja4.wav*). Po odszumieniu nagrania wykonaj przycięcie (*Delete*) zaznaczonego fragmentu, a następnie wzmocnienie i wyciszenie fragmentu nagrania (*Narzędzie obwiedni*). Zastosuj różne efekty dźwięku. W programie Audacity spacja uruchamia i zatrzymuje odtwarzanie dźwięku, a przyciski *Home* i *End* przenoszą kursor, odpowiednio: na początek lub na koniec utworu. Nieudane próby zawsze można cofnąć (*Ctrl+Z*). Zapisz powstały po zmianach utwór w kilku formatach: WAV, MP3, OGG i FLAC.

Zadanie 4.14. Wykorzystaj powstały w poprzednim zadaniu utwór jako podkład dla lektora czytającego bajki w plikach *len.wav*, *samochwala.wav* lub *klamczucha.wav*. Zwróć uwagę na oś czasu w tych utworach. Jeśli to konieczne, łagodnie wycisz muzykę po zakończeniu czytania. Powstały plik wyeksportuj do formatu MP3.

Zadanie 4.15. Przygotuj miks muzyczny. Z krótkich fragmentów utworów złóż własną kompozycję. Zadbaj o płynne przejścia między utworami (dodawaj kolejne fragmenty poleceniem *Importuj* lub wklejaj fragmenty w nowych ścieżkach dźwiękowych, które można przesuwać względem siebie, wykorzystując *Narzędzie do przesuwania w czasie*). Pod koniec składanki utworów dodaj nagraną bezpośrednio w programie Audacity własną wypowiedź, na przykład: *Miksował dla was Jan Nowak*. W tym czasie lekko wycisz muzykę. Zapisz powstały projekt jako *mix.aup* oraz wyeksportuj go do pliku *mix.mp3*.

4.3. Wideo

Podstawowym źródłem informacji o otaczającym nas świecie jest obraz. Obraz może być *statyczny* (nieruchomy) lub *dynamiczny* (ruchomy). Czy można zarejestrować ruch? Właściwie nie. Można jedynie zapisać kolejne fazy ruchu. Jak to się więc dzieje, że widzimy obraz ruchomy w kinie lub na monitorze komputera? Za to zjawisko częściowo odpowiedzialna jest siatkówka naszego oka. Zapamiętuje ona na chwilę obraz, który przed chwilą zniknął. Co więcej, jeśli pojawi się kolejny obraz, który będzie się nieco różnił od

poprzedniego (kolejna faza ruchu), to uzyskamy wrażenie płynności ruchu. Za połączenie tych obrazów w całość odpowiedzialne są procesy zachodzące w naszym mózgu.

Kiedy to zjawisko następuje? Obraz ruchomy człowiek widzi, gdy prędkość wyświetlania filmu przekroczy granicę 11 obrazów na sekundę. Jeśli te obrazy (klatki filmu) będą rejestracją kolejnych faz ruchu, to zobaczymy obraz ruchomy. Można więc powiedzieć, że film to nic innego jak szybko pojawiające się kolejne obrazy wyświetlane z klatek taśmy filmowej. Dla potrzeb kina na ekranie wyświetlane są 24 klatki na sekundę, zaś w telewizji analogowej w ciągu sekundy pojawia się 25 pełnych obrazów.

4.3.1. Cyfrowy zapis filmu

Gdy obraz telewizyjny (analogowy) zapiszemy, zamieniając każdą klatkę filmu na zapis cyfrowy w formacie nieskompresowanym RGB24 (RGB 24 bity informacji o każdym pikselu obrazu), to powstanie bardzo duży plik wynikowy.

Format RGB24 charakteryzuje się brakiem kompresji, więc zajmuje bardzo dużo miejsca na dysku. Przy rozmiarze obrazu telewizyjnego 768×576 pikseli i filmie PAL (25 klatek na sekundę) objętość 1 sekundy pliku AVI będzie wynosiła:
768 pikseli×576 pikseli×3 B×25 klatek ≈ 33 MB/s.
Czyli 1 minuta filmu zajmuje około 1980 MB.

Zmniejszenie powstałego w ten sposób pliku wynikowego będzie możliwe jedynie po zastosowaniu odpowiednich kodeków, które pozwolą na skompresowanie powstałego filmu.

Film to nie tylko obraz, ale także dźwięk. Można śmiało powiedzieć, że film to obraz i dźwięk. Czy do kodowania obrazu oraz dźwięku używamy tych samych mechanizmów? I tak, i nie.

Definicja

Kontener multimedialny to pojemnik dla ścieżki wideo i ścieżek audio. Dodatkowo w kontenerze można umieścić napisy, informacje o rozdziałach i tzw. metadanych pozwalających na ich poprawne jednoczesne odtworzenie. Kontener przechowuje kompletny materiał multimedialny w jednym pliku.

Kontenery umożliwiają przechowywanie materiału AV (ang. *Audio/Video*) w wielu formatach. Przykładami kontenerów multimedialnych są: *AVI*, *MOV*, *RealMedia*, *MPEG-2* i *MPEG-4*. Natomiast przykładowe kodeki użyte do kompresji wideo to: *DivX*, *h.264*, *Indeo5*, a do kompresji dźwięku: *ac3*, *OGG* i inne.

Różnorodność metod kompresji obrazu ruchomego jest związana między innymi z przeznaczeniem pliku. Ten sam film nagrany kamerą cyfrową będzie inaczej kompresowany, gdy chcemy go odtworzyć na smartfonie, jeszcze inaczej, gdy umieścimy go na stronie WWW, i całkiem inaczej, gdy chcemy odtworzyć go na uroczystości rodzinnej, prezentując go w najwyższej jakości.

I tu pojawia się pytanie: jak wybrać odpowiedni kodek? Nie jest to wcale takie łatwe, gdyż wymaga pewnej wiedzy związanej z własnościami kodeków. Należy również pamiętać

o tym, że większość kodeków jest zastępowana nowymi wersjami, a co pewien czas pojawiają się całkiem nowe standardy i związane z nimi odpowiednie kodeki. Warto wiedzieć, jakie zastosowanie mają poszczególne kodeki. W tabeli 4.8 zawarto najważniejsze formaty wideo oraz powiązane z nimi kodeki i ich zastosowanie.

Tabela 4.8. Wykaz kodeków wideo i ich przeznaczenie

Nazwa kodeka	Typy plików	Zastosowanie
AVCHD	.mts	cyfrowe kamery AVCHD
DV	.avi, .mov	kamery DV, programy do montażu
DivX	.divx, .avi, .mov, .mkv	filmy w internecie, kopie DVD
DivX HD	.divx, .avi, .mov, .mkv	filmy w internecie, kopie DVD
M-JPEG	.avi, .mov	programy do montażu
MPEG-1	.mpg, .mkv, .dat	filmy w internecie, płyty Video CD, cyfrowe aparaty fotograficzne
MPEG-2	.htr, .ts, .mpg, .mts, .rec, .tod, .trp, .vid, .vob	cyfrowa telewizja, DVD, filmy w internecie, kamery DVD i z dyskiem twardym
MPEG-2 HD	.htr, .ts, .mpg, .mts, .rec, .tod, .trp, .vid, .vob	telewizja HD, płyty Blu-ray, kamery HDV
MPEG-4 AVC (H.264) Level 1.2	.mpg, .m4v, .mp4	filmy w internecie, internetowe wypożyczalnie filmów
MPEG-4 AVC (H.264) Level 4	.mpg, .m4v, .mp4	telewizja HD, płyty Blu-ray
QuickTime	.mov	filmy w internecie
QuickTime HD	.mov	filmy w internecie
Real Video	.rm, .rv	filmy w internecie
Windows Media	.wmv	filmy w internecie
Windows Media HD	.wmv	filmy w internecie

Większość kodeków wchodzi w skład programów do odtwarzania i razem z nimi trafia na dyski twarde komputerów. Jednak zdarza się, że po uruchomieniu odtwarzania filmu nie pojawia się obraz lub dźwięk (albo jedno i drugie). W takim przypadku najprawdopodobniej w urządzeniu lub systemie operacyjnym komputera nie ma odpowiednich kodeków. W celu uniknięcia tego typu problemów stworzono szereg gotowych zestawów kodeków, które można pobrać z internetu i zainstalować jako „paczkę" kodeków. Najprawdopodobniej instalacja jednego z takich pakietów pozwoli uniknąć tego typu kłopotów. Najpopularniejszymi pakietami kodeków są: K-Lite Codec Pack (*http://www.codecguide.com*), Vista Codec Package (*http://shark007.net*) czy XP Codec Pack (*http://www.xpcodecpack.com*).

Jak działają kodeki?

Kodeki wideo wykorzystują techniki podobne do tych, z których korzystają kodeki audio. Dochodzą do tego specjalne metody, które doskonale nadają się do kompresowania obrazów.

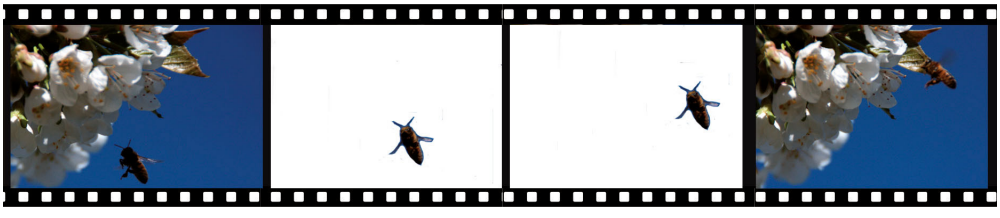
Kodeki wideo pojedynczego obrazu — w tym trybie kodeki zapisują za każdym razem pełną klatkę obrazu, jak na rysunku 4.46. Z tego powodu wynikowe pliki wideo mają bardzo duże rozmiary.



Rysunek 4.46. Film z pełną informacją o obrazie każdej klatki filmu

Kodeki pojedynczego obrazu, na przykład DV, przeliczają każdy obraz na nowo. Łączą piksele w bloki, które następnie pojedynczo kompresują. Oprócz tego kodek redukuje liczbę odcieni koloru w ramach bloku, podobnie jak w formacie JPG w fotografii cyfrowej.

Kodeki ruchomych obrazów, na przykład MPEG, dodatkowo redukują ilość danych, łącząc ze sobą sekwencje z wielu pojedynczych obrazów. Kodek zapisuje w całości tylko obraz pierwszy, a z następnych tylko te jego części, które różnią się od obrazu poprzedniego. Tak więc z nagrania lecącej pszczoły kodek zapisuje tylko ruchy owada, pomijając elementy nieruchome — kwiaty w tle (jak na rysunku 4.47).



Rysunek 4.47. Film z redukcją danych o obrazie kolejnych klatek filmu

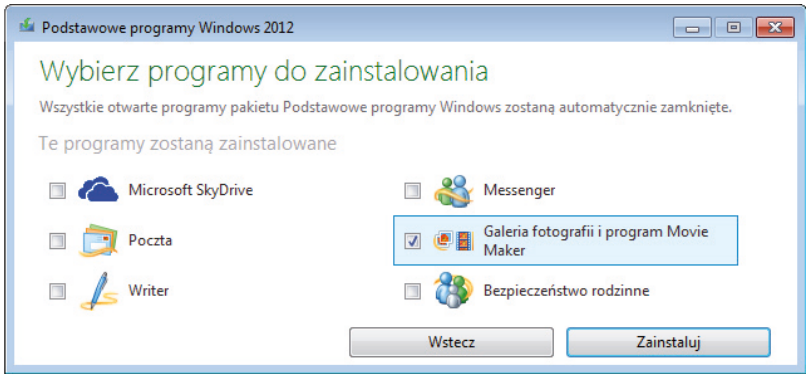
Aby skuteczniej kompresować materiał wideo, kodeki zachowują wszelkie dane o pierwszej klatce sceny oraz informacje, czym od niej różnią się kolejne klatki filmu.

Wymień urządzenia (oprócz komputera), w których stosuje się kodeki audio i wideo.

4.3.2. Montaż filmu i jego publikacja

Do montażu filmu wykorzystamy dodatkowe oprogramowanie firmy Microsoft, które możesz pobrać ze strony WWW: *http://windows.microsoft.com/pl-PL/windows-live/essentials-home*. Z całego oferowanego pakietu podstawowych programów Windows wybierz i zainstaluj *Galerię fotografii i Movie Maker* (rysunek 4.48), które współpracują ze sobą, dzięki czemu można łatwo uporządkować i wybrać zdjęcia oraz wideo do filmu.

O tym, jak korzystać z programu Movie Maker, możesz przeczytać na stronie producenta oprogramowania, który udostępnił obszerny materiał szkoleniowy (*http://explore.live.com/windows-live-movie-maker?os=other*). Z kursu dowiesz się, jak dodać zdjęcia i filmy wideo, edytować tekst, przejścia, efekty, jak opublikować swoje filmy prosto z programu



Rysunek 4.48. Wybór programów do zainstalowania z pakietu podstawowych programów Windows Live

Movie Maker w ulubionych witrynach — na przykład: YouTube, Facebook, Windows Live SkyDrive i innych.

Zmieniaj wszystko, co chcesz, lub pozwól, aby program Movie Maker zrobił to za Ciebie.

Ćwiczenie 4.15. Wykorzystując zamieszczone na płycie CD materiały w folderach: *Filmy*, *Fotografie* i *Pliki dzwiekowe*, zmontuj film pod tytułem: „Jesień w Borach Tucholskich". Wykorzystaj do tego ćwiczenia program *Windows Live Movie Maker*. Ścieżkę dźwiękową zamieszczoną na przykład na końcu filmu możesz zmontować z nagrań dostępnych na płycie CD. Skorzystaj ze wspomnianego wcześniej kursu zamieszczonego na stronie producenta oprogramowania: *http://explore.live.com/windows-live-movie-maker?os=other*.

4.4. Zadania sprawdzające do rozdziału 4.

Zadanie 4.16. Rozwiąż poniższy test. W poszczególnych zadaniach tylko jedna odpowiedź jest prawidłowa.

1. W obrazie z 24-bitową głębią koloru występuje:
   a) 256 kolorów.
   b) 16,7 mln kolorów.
   c) 16 kolorów.
   d) 256 odcieni szarości.
2. Która definicja opisuje grafikę wektorową?
   a) Obrazy tworzone są z regularnie ułożonych pikseli. Mają one różne kolory lub odcienie jasności.
   b) Liczba pikseli przypadająca na jednostkę powierzchni jest wielkością stałą, dlatego przy powiększeniu obrazu występuje efekt powiększenia piksela.
   c) Elementy graficzne rysunku noszą nazwę obiektów. Każdy obiekt stanowi niezależną część obrazu, zdefiniowaną za pomocą takich właściwości jak: kolor, kształt, kontur, wielkość i położenie na rysunku.

d) Elementy obrazu są umieszczone w przestrzeni trójwymiarowej i celem edytora jest przede wszystkim przedstawienie trójwymiarowego świata na dwuwymiarowym obrazie.

3. Kompresja stratna pliku graficznego:
   a) polega na zmniejszaniu wielkości pliku graficznego przy równoczesnej utracie jakości.
   b) polega na zmniejszaniu wielkości pliku graficznego bez utraty jakości.
   c) polega na zwiększeniu wielkości pliku graficznego przy równoczesnej utracie jakości.
   d) polega na zwiększeniu wielkości pliku graficznego bez utraty jakości.
4. W grafice wektorowej wyróżniamy trzy rodzaje węzłów (punktów zaczepienia). Z jakim spośród wymienionych węzłów **nie** spotkamy się w grafice wektorowej?
   a) Z węzłem ostrym.
   b) Z węzłem gładkim.
   c) Z węzłem symetrycznym.
   d) Z węzłem wklęsłym.
5. Który z poniższych opisów charakteryzuje format GIF?
   a) System bezstratnej kompresji plików, maksymalnie 32-bitowy tryb koloru, szczególnie polecany do zdjęć i skanowanych obrazków.
   b) Format kompresji bezstratnej, który wykorzystuje maksymalnie 8-bitowy tryb kolorów, przezroczyste tła i umożliwia zapisywanie animacji.
   c) Format kompresji bezstratnej pozwalający na rysowanie przezroczystego tła, występuje w odmianach 8- oraz 24-bitowej.
   d) Format kompresji stratnej, który wykorzystuje maksymalnie 8-bitowy tryb kolorów, przezroczyste tła i umożliwia zapisywanie animacji.
6. Wskaż podstawowe parametry dźwięku zapisanego na płycie CD:
   a) 196 kHz, 32 bity, stereo.
   b) 44,1 kHz, 16 bitów, stereo.
   c) 22 kHz, 16 bitów, stereo.
   d) 96 kHz, 32 bity, stereo.
7. Poniżej podano różne formaty plików. Wskaż, w której grupie wymieniono plik inny niż dźwiękowy.
   a) WAV, AIFF, PCM.
   b) WAV, MP3, PNG.

c) WMA, MP3, AAC.

d) PCM, APE, OGG.

8. W kolorowej drukarce atramentowej, w której znajdują się cztery pojemniki z tuszem w kolorach zgodnych ze standardem CMYK, skończył się atrament w kolorze Y. Jakich kolorów nie otrzymamy na testowym wydruku? Wskaż odpowiednią grupę.

a) Żółty, niebieski, cyjan.

b) Żółty, niebieski, zielony.

c) Żółty, czerwony, magenta.

d) Żółty, czerwony, zielony.

9. Dany jest obraz o następujących parametrach: 640×480×24 (szerokość×wysokość×głębia bitowa). Do zapisania obrazu (bez kompresji) potrzebna jest pamięć wielkości:

a) 550 kB.

b) 180 kB.

c) 930 kB.

d) 1,5 MB.

10. Kwadrat o współrzędnych wierzchołków (0,0), (1,0), (1,1), (0,1) został poddany operacji skalowania ze współczynnikami Sx = 2, Sy = 3 i przesunięty o wektor [2,3]. Współrzędne figury po tych transformacjach to:

a) (0,0), (1,0), (1,1), (0,1).

b) (2,3), (3,3), (3,4), (2,4).

c) (0,0), (2,0), (2,3), (0,3).

d) (2,3), (4,3), (4,6), (2,6).

11. Jeżeli trzykrotnie zwiększymy rozdzielczość obrazu cyfrowego, czyli na tej samej długości znajdzie się trzy razy więcej pikseli, wtedy liczba pikseli w całym obrazie wzrośnie:

a) trzykrotnie.

b) dziewięciokrotnie.

c) 2^3-krotnie, czyli ośmiokrotnie.

d) 3^3-krotnie, czyli dwudziestosiedmiokrotnie.

12. Jakiego rozmiaru pamięci RAM wymaga obraz RGB o wymiarach 10×15 cali z 24-bitową głębią (format zbliżony do A4) i rozdzielczości 300 ppi?

a) 131,8 kB.

b) 43,9 kB.

c) 12,8 MB.

d) 38,6 MB.

Zadanie 4.17. Odpowiedz na pytania typu prawda – fałsz.

1. W grafice wektorowej można zapisywać obrazy w różnych rozdzielczościach.
2. Bitrate to jedna z metod kodowania obrazu.
3. W modelu RGB odcienie szarości reprezentowane są przez punkty leżące na przekątnej (0,0,0) – (1,1,1).
4. Jasność obrazu to inaczej jaskrawość obrazu — terminów tych można używać wymiennie.
5. Skaner z układem CCD skanuje oryginały zarówno płaskie, jak i nieprzylegające do szyby skanera, a otrzymany obraz przypomina fotografię.
6. Barwę RGB (214,60,120) tworzą dwie składowe: barwa chromatyczna (154,0,60) oraz biel, czyli barwa achromatyczna (60,60,60).
7. Kompresja obrazu metodą RLE jest kompresją stratną.
8. W metodach stratnych kompresja ma wpływ na pogorszenie jakości obrazu.
9. Kompresja LZW jest kompresją słownikową.
10. Głębia bitowa obrazu ma wpływ na liczbę użytych barw oraz wielkość zapisanego pliku.
11. W krzywych Béziera zmiana położenia dowolnego punktu kontrolnego powoduje zmianę wyglądu całej krzywej.
12. Na stronach WWW można zamieszczać obrazy tylko w formatach GIF i JPG.
13. W metodach bezstratnych obraz po dekompresji może różnić się od obrazu pierwotnego.
14. W metodzie Huffmana kod przypisywany barwie zależy od częstotliwości jej występowania.
15. Zapisanie obrazu w formacie BMP z 24-bitową głębią obrazu oznacza, że każdy piksel obrazu wykorzystuje 3 bajty informacji o kolorze.
16. Aliasing to efekt będący skutkiem powiększenia (skalowania) obiektu w grafice wektorowej.
17. Obraz w odcieniach szarości ma 8-bitową głębię kolorów.
18. W modelu RGB przy reprezentacji barwy za pomocą 24 bitów barwa czarna ma współrzędne (255, 255, 255).
19. Obrót obrazu o 90 stopni wymaga zamiany miejscami odpowiednich wierszy i kolumn.
20. Można przekształcić obraz w odcieniach szarości w obraz dwubarwny.
21. Plik w formacie MP3 jest zawsze tej samej jakości.
22. Najlepszym formatem do zapisu dźwięku jest format MP3.

23. Z pliku w formacie MP3 zawsze można otrzymać oryginalny dźwięk w formacie WAV.
24. Film wymaga wyświetlenia minimum 32 klatek na sekundę, w przeciwnym razie zobaczymy tzw. stop-klatki.

Zadanie 4.18. Projekt edukacyjny Przygotuj samodzielnie lub w zespole trzyosobowym 180-sekundowy film edukacyjny. Film powinien być związany z obsługą komputera. Może on dotyczyć: zadań systemu operacyjnego lub bezpieczeństwa danych, sieci komputerowych, budowy komputera, urządzeń wejścia i wyjścia, oprogramowania, algorytmów, baz danych, tworzenia strony WWW, edycji obrazu, dźwięku lub filmu. Wykorzystaj zrzuty ekranowe i fragmenty filmów nagrane na przykład aparatem fotograficznym lub programem do rejestracji aktywności na ekranie monitora (na przykład CamStudio na licencji open source — *http://camstudio.org*). Nagraj ścieżkę dźwiękową z narracją. Przygotuj obrazy na początek i zakończenie filmu. Dodaj napisy tytułowe i końcowe, w których podasz ewentualne źródła informacji czy fotografii. Film przygotuj tak, aby można go było pokazać w dobrej jakości na ekranie monitora. Pamiętaj o dydaktycznym aspekcie tworzonego filmu i prawie autorskim. Przygotuj dokumentację na temat powstania filmu i krótką zapowiedź zachęcającą do jego oglądania.

Rozdział 5.

KIERUNKI ROZWOJU TECHNOLOGII INFORMATYCZNYCH ORAZ ASPEKTY ETYCZNE, PRAWNE I SPOŁECZNE W ZASTOSOWANIACH INFORMATYKI

Przewidywanie przyszłości jest bardzo trudne. Słynna reguła Doliny Krzemowej (ang. *Silicon Valley*) mówi: *nie rób planów na dłużej niż na dwa lata*. W ciągu tych dwóch lat w technologii może zmienić się bardzo wiele. Moc obliczeniowa komputerów zostanie przynajmniej podwojona, pojemność dysków wielokrotnie wzrośnie, przybędzie różnych programów komputerowych, dziesiątki tysięcy nowych użytkowników uzyskają dostęp do internetu. A co nas czeka za 5, 10 czy 30 lat? Żyjemy w bardzo ciekawych czasach dynamicznego rozwoju i szybkich przemian. W jakim kierunku pójdą konstruktorzy, co zostanie odkryte, jak wykorzystane? Jeszcze kilkanaście lat temu nikt nie sądził, że tak dynamicznie będzie rozwijał się internet, a technologie takie jak HDTV czy Bluray w bardzo krótkim czasie staną się powszechne.

5.1. Kierunki rozwoju informatyki i jej zastosowań

Pojawiają się nowe technologie, stare zaś odchodzą w zapomnienie. Płyta winylowa, kaseta magnetofonowa, taśma VHS to technologie i urządzenia, które choć są nadal używane, można spotkać głównie w muzeach techniki. Niedługo dołączy do nich płyta CD, a może i płyta DVD. A rozwój komputerów przyprawia wręcz o zawrót głowy. Komputer dwu-, trzyletni jest już przestarzały. Obecnie jedynie mała grupa specjalistów nadąża za ciągłymi zmianami. Nowe wersje komponentów, coraz szybsze i nowocześniejsze procesory wymuszają ciągłe zmiany konfiguracji komputera. Jesteśmy świadkami i uczestnikami tych przemian. Ciągle otrzymujemy informacje o nowych odkryciach i słyszymy: najszybszy, najdoskonalszy, najlepszy. Jednak czas panowania poszczególnych rozwiązań technologicznych jest coraz krótszy, a zmiany następują coraz szybciej. Wciąż słyszymy o nowszym rozwiązaniu, szybszym od najszybszego, doskonalszym od najdoskonalszego.

5.1.1. Krótkie kalendarium wydarzeń ważnych dla rozwoju komputeryzacji

Każda epoka miała swoje wynalazki. Czasami nawet nazwa epoki wywodzi się właśnie od największego w swoim czasie wynalazku, na przykład *epoka kamienia łupanego*. Za początek niektórych epok umownie przyjmuje się czas wynalezienia pewnej technologii, przykładem jest *epoka brązu*. Można zatem dokonać podziału wynalazków ze względu na czas ich wynalezienia. Poniżej wyszczególniono niektóre z nich, mające wpływ na rozwój komputeryzacji.

- Wynalazki okresu prehistorycznego: narzędzia kamienne, ogień.
- Wynalazki starożytności: pismo, matematyka, kalendarz.
- Wynalazki średniowiecza: proch, armata (choć nie wiem, jaki mają związek z komputerem…).
- Wynalazki wieków XVI – XVIII: luneta, teleskop, silnik parowy, mikroskop.
- Wynalazki wieku XIX: silnik elektryczny i spalinowy, fotokomórka, radio.
- Wynalazki wieku XX: elektryczność, telewizja, komputer, laser, światłowód.
- Wynalazki wieku XXI: …?

Niektóre wynalazki pojawiły się na przełomie wieków, aby w następnym stuleciu rozkwitnąć. Spójrzmy jednak na kilka szczególnych dat, które pokazują dynamikę rozwoju.

1543 — Mikołaj Kopernik zmienia porządek świata, dając początek rewolucji naukowej.

1687 — Isaac Newton rozpoczyna erę rozwoju fizyki.

1888 — Heinrich Rudolf Hertz odkrywa fale elektromagnetyczne, dając podstawy rozwoju radiokomunikacji.

1925 — powstaje dział fizyki: *mechanika kwantowa* (teoria *kwantów*). Teoria praw ruchu obiektów świata mikroskopowego rozpoczyna rewolucję technologiczną, dzięki której powstają kolejno: lampy elektronowe, półprzewodniki, tranzystory, układy scalone, czyli wszystko, co składa się na budowę współczesnych komputerów.

1940 – 1949 — powstają pierwsze komputery.

1947 — wynaleziono tranzystory.

1959 — wynaleziono układy scalone.

1971 — powstają pierwsze mikroprocesory.

1981 — firma IBM tworzy pierwszy komputer osobisty PC.

1982 — pierwsze 8-bitowe komputery domowe (ZX Spectrum) trafiają „pod strzechy".

1984 — Apple Macintosh wprowadza na rynek swój komputer — popularny Mac.

1994 — powstają pierwsze serwery WWW, które dają początek nowym możliwościom (sklepy internetowe, bankowość internetowa, poczta elektroniczna, wymiana danych, informacji).

1995 — premiera systemu Microsoft Windows 95.

1997 — IBM buduje komputer Deep Blue, któremu udaje się dwa razy pokonać szachowego mistrza świata Garriego Kasparowa.

2000 — AMD przedstawia nowy procesor Athlon taktowany zegarem 1,1 GHz. W odpowiedzi na to Intel jeszcze w tym samym miesiącu prezentuje układ 1,5 GHz. Wyścig trwa...

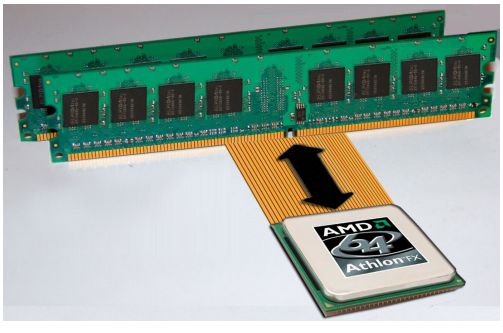
A jak jest dzisiaj? Ogromne pojemności dysków, wielordzeniowe i bardzo wydajne procesory, wielozadaniowe systemy operacyjne, optyczne dyski o dużej pojemności, miniaturowe karty pamięci, fotografia, film, telewizja, dźwięk cyfrowy bardzo dobrej jakości, powszechny dostęp do internetu.

5.1.2. Co przyniesie najbliższa przyszłość?

Oto kilka nowinek, które prawdopodobnie zmienią nie tylko architekturę komputera, ale i sposób zapisu danych na nośnikach zewnętrznych, a może nawet doprowadzą do zmiany technologii z obecnej, opartej na krzemie, na opartą na fotonach.

Technologia bezpośrednich połączeń typu punkt-punkt

Dotychczas komunikacja między procesorem a pozostałymi komponentami komputera odbywała się poprzez magistrale systemowe. Jednak firma AMD zaproponowała inne rozwiązanie, które pozwoliło na bezpośrednią komunikację między procesorem a pamięcią RAM, bez pośrednictwa mostka północnego. Rozwiązanie to oparte jest na koncepcji połączenia punkt-punkt (ang. *point to point*), dzięki czemu uzyskano separowane połączenie procesora i pamięci RAM. Zalety takiego rozwiązania są oczywiste. W przypadku zastosowania szyny systemowej, która jest wykorzystywana do komunikowania się z wieloma urządzeniami, komunikacja CPU z RAM możliwa jest jedynie wtedy, gdy szyna nie jest zajęta. W przypadku rozwiązania *point to point* nie ma takiego problemu. Rysunek 5.1 schematycznie pokazuje rozwiązanie zastosowane w tym procesorze.



Rysunek 5.1. Połączenie punkt-punkt CPU z pamięcią RAM

Połączenie punkt-punkt umożliwia komunikowanie się dwóch urządzeń dzięki dwóm niezależnym łączom jednokierunkowym. Takie połączenie znacznie przyspiesza przesyłanie danych, gdyż nie jest ono współużytkowane przez inne urządzenia. Przejście na takie rozwiązanie i rezygnacja z tradycyjnych szyn systemowych pociąga za sobą całkowicie inną konstrukcję płyty głównej. We współczesnych komputerach można spotkać rozwiązania konstrukcyjne oparte na połączeniach punkt-punkt, znane jako HyperTransport i SATA.

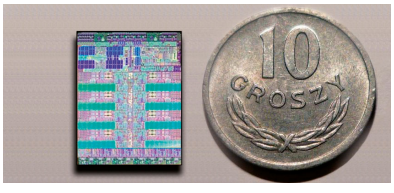
Innym wynalazkiem, w którym zastosowano połączenie punkt-punkt, jest procesor serwerowy Tukwila firmy Intel.

To czterordzeniowy układ, który ma 24 MB pamięci podręcznej. Składa się z **2 miliardów tranzystorów** i jest wyposażony w rozbudowaną technologię wielowątkowości. Szyna systemowa zapewniająca komunikację pomiędzy rdzeniami to QuickPath, wykorzystująca technologię bezpośrednich połączeń typu punkt-punkt. Tukwila ma podwójny, zintegrowany kontroler pamięci. Ten procesor serwerowy ma oferować ponad dwukrotnie wyższą wydajność niż układy poprzedniej generacji.

Terahercowe procesory?

Pod koniec XX wieku walka o wydajność procesorów związana była z częstotliwością zegarów taktujących, które ostatecznie osiągnęły granicę 2 – 3 GHz. Rozpoczęto więc mnożenie rdzeni. Naukowcy twierdzą jednak, że znowu pojawiła się szansa na kilkadziesiąt, kilkaset, a nawet kilka tysięcy gigaherców!

Naukowcy z uniwersytetu w Utah twierdzą, że w ciągu kilku następnych lat zastosujemy technologie optyczne wykorzystujące światło podczerwone, co otworzy nam drogę do komputerów opartych na procesorach taktowanych znacznie szybszymi zegarami niż obecnie. W eksperymencie wykonali „obwody scalone", które zamiast elektryczności wykorzystywały światło. Na rysunku 5.2 pokazano przybliżoną wielkość tego prototypowego obwodu, dla porównania zestawiono go z monetą.

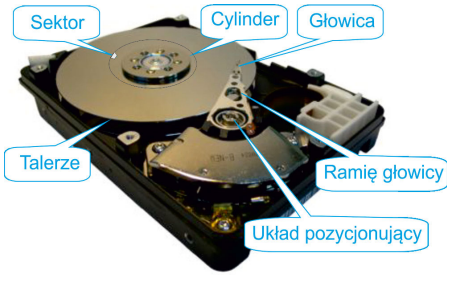


Rysunek 5.2. Pierwowzór procesora optycznego w zestawieniu z monetą

Układ składał się z folii stalowej o grubości **625 mikrometrów**. Wybito w niej kwadratowe rowki o rozmiarze 500×50 mikrometrów. Ułożono je w taki sposób, by uformowały „obwody" — okazało się, że biegnące w ich wnętrzu światło można było kierować w różne strony oraz dzielić i łączyć. I to wszystko na całkiem płaskiej powierzchni, bez konieczności montowania trójwymiarowych rurek ze szkła czy pleksi. Nie brzmi to może imponująco, ale w tym prototypie sygnały modulowano z oszałamiającą częstotliwością 300 miliardów razy na sekundę, co oznacza, że osiągnięto częstotliwość **300 GHz**.

Dyski SSD

Najpopularniejszą pamięcią masową są dyski twarde (HDD), w których zapis i odczyt danych jest magnetyczny. Taki dysk składa się z zamkniętego w obudowie, wirującego talerza (dysku) lub zespołu talerzy, wykonanych najczęściej ze stopów aluminium, o wypolerowanej powierzchni pokrytej nośnikiem magnetycznym (grubości kilku mikrometrów), oraz z głowic elektromagnetycznych umożliwiających zapis i odczyt danych. Na każdą powierzchnię talerza dysku przypada po jednej głowicy odczytu i zapisu. Głowice są umieszczone na elastycznych ramionach i w stanie spoczynku stykają się z talerzem blisko osi, a w czasie pracy unoszą się — ich odległość nad talerzem jest



Rysunek 5.3. Budowa dysku HDD

stabilizowana dzięki sile aerodynamicznej. To najpopularniejsze obecnie rozwiązanie. Na rysunku 5.3 pokazano otwarty dysk HDD z opisem jego budowy.

Dyski SSD (ang. *Solid State Drive*) to urządzenia oparte na pamięci flash, które w pierwszej kolejności zastąpią dyski twarde (HDD) w laptopach oraz urządzeniach przenośnych. Cechą charakterystyczną takich dysków jest bardzo duża prędkość zapisu oraz odczytu. Obecne dyski SSD oferują transfer na poziomie około 200 MB/s. Jest to transfer porównywalny z najlepszymi modelami dysków HDD. Jednak należy się spodziewać, że transfer danych w dyskach SSD w przyszłości będzie znacznie szybszy aniżeli w dyskach HDD. Dyski SSD do magazynowania danych wykorzystują pamięć flash i nie mają żadnych części ruchomych, tak jak to ma miejsce w tradycyjnych HDD. Brak części ruchomych oznacza niższe zużycie energii, niższą temperaturę pracy podczas użytkowania, cichszą pracę oraz odporność na uszkodzenia mechaniczne i wstrząsy. Tak więc wszędzie tam, gdzie występuje duży stopień miniaturyzacji (MP3, telefony komórkowe) oraz narażenie na wstrząsy (laptopy), będą stosowane takie dyski, a ich zalety spowodują, że staną się powszechne. Przykładowy dysk SSD pokazano na rysunku 5.4. Dysk jest wyposażony w interfejs SATA II, może więc śmiało zastąpić tradycyjny dysk HDD.



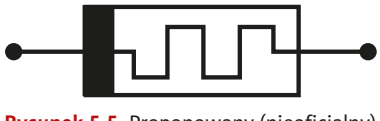
Rysunek 5.4. Dysk SSD

Memrystory

Naukowcom w laboratoriach HP udało się wyprodukować nowy element elektroniczny, który może znacznie przyspieszyć uruchamianie komputerów i stworzyć zupełnie nowe maszyny, które mogłyby naśladować procesy zachodzące w ludzkim mózgu.

Prawie 40 lat temu przewidziano teoretycznie istnienie czwartego, biernego elementu elektronicznego, który nie może być zastąpiony przez dowolną kombinację pozostałych trzech znanych elementów biernych: rezystora, cewki i kondensatora. Jest to *memrystor* (ang. *memristor*; nazwa pochodzi od angielskich słów: *memory* i *resistor*) mający pewną rezystancję, której wielkość zależy od uprzednio przyłożonego do niego napięcia. Po odłączeniu napięcia wartość rezystancji zostaje zapamiętana i nie ulega zmianie. Ponowne podłączenie innego poziomu napięcia zmienia wartość rezystancji memrystora, która nie zmienia się aż do ponownego przyłożenia napięcia o innej wartości.

Jak działa memrystor?
Zbudowany jest z dwóch warstw tlenku tytanu, podłączonych do przewodów elektrycznych. Gdy do jednej z warstw przyłożone zostanie napięcie, zmienia się rezystancja drugiej warstwy. Zmiany rezystancji są różne w zależności od wartości przyłożonego napięcia. Można je rejestrować jako dane. Po odłączeniu zasilania memrystor pamięta więc informacje. Proponowany, choć jeszcze nieoficjalny symbol memrystora pokazano na rysunku 5.5.



Rysunek 5.5. Proponowany (nieoficjalny) symbol memrystora

Jak można go wykorzystać?
Na bazie memrystorów mogłyby powstać nowe rodzaje pamięci RAM. Dziś stosuje się najczęściej pamięć typu DRAM, której głównym problemem jest ulotność. Informacje, które są przechowywane w kondensatorach, giną, gdy tylko odetnie się źródło zasilania. Dlatego kiedy włączamy ponownie komputer, trzeba długo czekać, zanim będzie on gotowy do pracy. Wyobraź sobie komputer, który po wyłączeniu zasilania pamięta wszystko, a po ponownym włączeniu natychmiast jest gotowy do działania.

Oczywiście, technologia ta jest we wczesnej fazie rozwoju, jednak pierwsze kroki zostały już poczynione. Wykorzystanie własności memrystorów z pewnością zrewolucjonizuje rynek sprzętu komputerowego.

USB 3.0

Uniwersalna magistrala szeregowa USB 3.0 (ang. *Universal Serial Bus*) jest nowym rodzajem portu komunikacyjnego, który zapewnia:

- zachowanie kompatybilności z USB 2.0,
- zwiększenie szybkości transmisji danych,
- zwiększenie wydajności prądowej samego portu.

Urządzenia pracujące w tym standardzie mogą być łączone ze sobą i tworzyć sieć o topologii drzewa. Możliwe jest podłączenie aż 127 urządzeń, jednak ze względu na pobór mocy i ograniczenia z tym związane (szczególnie w komputerach przenośnych) w praktyce liczbę podłączanych urządzeń trzeba będzie ograniczyć.

Specyfikacja standardu przewiduje dziesięciokrotne zwiększenie transferu danych — z poziomu 480 Mb/s (60 MB/s) przy USB 2.0 do 4,8 Gb/s (600 MB/s) przy USB 3.0. Tak duże zwiększenie prędkości transmisji danych będzie dotyczyło jedynie urządzeń pracujących w tym standardzie. Gdy do portu USB 3.0 podłączymy starsze urządzenie (USB 2.0), prędkość zostanie zmniejszona do maksymalnej prędkości przewidzianej w standardzie USB 2.0. Należy o tym pamiętać, szczególnie gdy łączymy urządzenia szeregowo. Wtedy wszystkie urządzenia będą działały w takim tempie jak najwolniejsze ogniwo łańcucha.

Aby osiągnąć maksymalną prędkość 600 MB/s, należy urządzenie z portem USB 3.0 podłączyć specjalnym kablem USB 3.0 do kontrolera USB 3.0. Kabel nie może być dłuższy niż 4,5 m. Kabel ten rozpoznasz po niebieskiej części wewnętrznej we wtyczkach (rysunek 5.6).



Rysunek 5.6. Gniazda i wtyczki USB 3.0

Zwiększono również wydajność prądową kontrolera USB 3.0 z 500 mA do 900 mA na każde urządzenie podłączone do portu USB 3.0. Pozwoliło to na wyeliminowanie dodatkowych zasilaczy choćby do takich urządzeń jak zewnętrzne dyski twarde.

Cloud computing — fantazja czy przyszłość?

Przetwarzanie w chmurze (ang. *cloud computing*) to architektura usług, w których przetwarzanie danych i obliczenia nie odbywają się na komputerze użytkownika, a są realizowane przez wiele rozproszonych serwerów pracujących w sieci internetowej. Usługi te dostępne są w internecie i realizowane w formie odpłatnej lub darmowej. Opłata za usługę jest opłatą za możliwość korzystania na przykład z arkusza kalkulacyjnego bez posiadania na niego licencji. Wykorzystuje się tu rozliczenia oparte na modelu „płać za zużycie" (ang. *pay-per-use*). W zamian otrzymujemy do dyspozycji zawsze najnowsze oprogramowanie, bez konieczności kupowania licencji na daną aplikację i jej aktualizację.

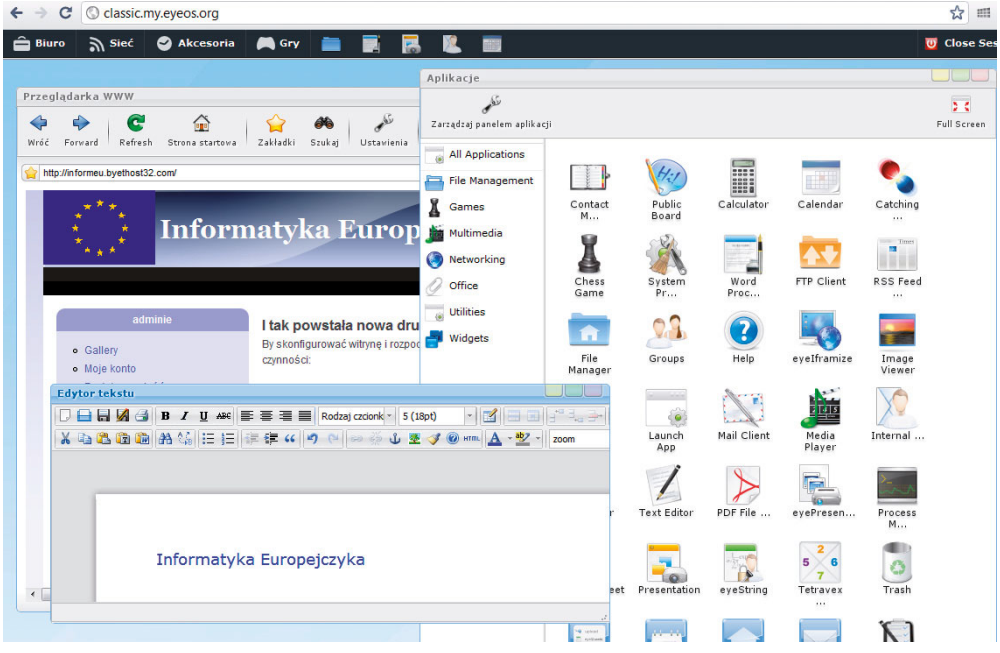
Obecnie w sieci można spotkać kilka ciekawych propozycji usług opartych na przetwarzaniu w chmurze.

Godną polecenia jest nieodpłatna usługa udostępniająca wirtualny system operacyjny w przeglądarce internetowej, której widok pokazano na rysunku 5.7. Jest ona dostępna po rejestracji pod adresem *http://icloud.com/pl*. Do dyspozycji są aplikacje takie jak: kalkulator, iconsola, kalendarz, kontakty, e-maile, pliki, dokumenty, zdjęcia i inne. Istnieje również opcja udostępniająca więcej możliwości. Jednak uaktywnia się ona dopiero po opłaceniu odpowiedniego abonamentu.



Rysunek 5.7. Ekran icloud — przetwarzanie w chmurze

Kolejnym ciekawym przykładem serwisu udostępniającego usługi oparte na przetwarzaniu w chmurze jest system operacyjny *eyeOS* dostępny pod adresem *http://www.eyeos.org/*. Po rejestracji na serwerze istnieje możliwość bezpłatnej pracy grupowej na jednym pulpicie serwera. Otwarty plik tekstowy oraz okno z wieloma dostępnymi aplikacjami działającymi w chmurze pokazano na rysunku 5.8.



Rysunek 5.8. Pulpit eyeOS z otwartymi aplikacjami

Istnieje również możliwość instalacji własnej wersji eyeOS. Możesz to uczynić, korzystając na przykład z serwera XAMPP (z konieczną obsługą języka PHP i serwerem bazodanowym MySQL). Sam proces instalacyjny nie powinien przysporzyć większych problemów.

Firma Microsoft stworzyła komercyjną platformę *Azure Services Platform*. Jest to platforma usług (ang. *Platform as a Service*) chmury obliczeniowej. Powodem stworzenia Azure Services Platform jest próba przejścia na nowy model dystrybucji oprogramowania, a także chęć dorównania innym firmom udostępniającym podobne usługi. Głównym celem jest oferowanie oprogramowania, które nie musi być instalowane na komputerze użytkownika, nie wykorzystuje mocy obliczeniowej maszyny lokalnej i jest zawsze dostępne. Windows Azure to praktycznie nieograniczone środowisko robocze w strukturze cloud computing.

E-learning — nauczanie na odległość

E-learning to nauczanie z wykorzystaniem sieci komputerowych i internetu. Jest to doskonałe uzupełnienie tradycyjnego nauczania (tzw. **nauczanie hybrydowe**). E-learning umożliwia nam samodzielne wybranie preferowanego poziomu nauczania i tempa pracy. Istnieje również możliwość sprawdzania nabytych umiejętności poprzez quizy, zadania czy formularze. **Każdy sam decyduje o tym, kiedy, gdzie i w jakim tempie się uczy.** Istnieje możliwość współpracy z prowadzącym szkolenie i wymiany doświadczeń z innymi uczestnikami kursu na czatach i forach dyskusyjnych udostępnionych na platformie.

Uczestnicy kursu mogą przesyłać rozwiązywane zadania ćwiczeniowe do sprawdzenia i oceny przez wykładowcę. Testy i kody źródłowe napisanych programów sprawdzane są automatycznie, co pozwala na natychmiastowe ocenianie wykonanych prac.

Przygotowanie kursów jest czasochłonne i wymaga użycia wielu aplikacji przez nauczyciela. Jednak dobrze przygotowany kurs e-learningowy może stanowić uzupełnienie wiedzy zarówno dla ucznia mającego trudności w nauce, jak i bardzo uzdolnionego.

Istnieje wiele platform e-learningowych LMS (ang. *Learning Management System*). Do najpopularniejszych można zaliczyć: bezpłatne Moodle (*http://moodle.org*), Olat (*http://www.olat.org*) oraz komercyjny Fronter (*http://com.fronter.info/*).

Materiały edukacyjne można przygotowywać w formatach *IMS* lub *SCORM*. Zastosowanie tych formatów gwarantuje możliwość przenoszenia materiałów edukacyjnych między platformami LMS.

Godne polecenia adresy URL darmowych kursów na platformach zdalnego nauczania:

Kurs matematyki: *http://www.moodle.pg.gda.pl/course/view.php?id=59*.

Kursy w języku angielskim: *http://moodlecommons.org/*.

Kurs o bezpieczeństwie w sieci: *https://kursy.dzieckowsieci.pl/*.

5.1.3. Zadania

Zadanie 5.1. Wyszukaj informacje na temat czasu przechowywania danych na poszczególnych nośnikach: FDD, HDD, CD, ZIP, DVD, HDS, Blu-ray.

Zadanie 5.2. Porównaj procesory firmy Intel i AMD pod kątem zastosowania do gier komputerowych.

Zadanie 5.3. W jaki sposób odczytywana jest informacja z płyty Blu-ray?

Zadanie 5.4. Wyszukaj informacje na temat HDMI.

Zadanie 5.5. Oto treść doniesienia prasowego:

Intel poinformował, że udało mu się wyprodukować pierwszy testowy chip składający się z osiemdziesięciu rdzeni. Układ jest w fazie testów, ale, jak twierdzi producent, niewykluczone, że za kilka lat takie superprocesory pojawią się w naszych domach.

Dopisz do rozpoczętego forum swoje uwagi, wątpliwości i komentarze:
- Ciekawi mnie, co z poborem prądu?
- Jakie oprogramowanie wykorzysta możliwości takiego układu?
- .....
- .....

Zadanie 5.6. Wyszukaj informacje na temat wyświetlaczy AMOLED. Podaj przykłady różnych zastosowań (te już zaproponowane i — ewentualnie — własne pomysły).

5.2. Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki

5.2.1. Prawo autorskie

Podstawowym dokumentem regulującym w Polsce prawa dotyczące ochrony własności intelektualnej jest ustawa z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (UOPA).

Przedmiotem prawa autorskiego jest utrwalony utwór rozumiany jako dobro niematerialne. Aby dany wytwór był utworem w rozumieniu UOPA, musi być rezultatem pracy człowieka, musi być oryginalny oraz musi mieć indywidualny charakter. Zatem utworem może być na przykład: projekt graficzny witryny internetowej, program komputerowy, dzieło plastyczne, fotografia, wiersz itp. Prawo autorskie nie zajmuje się przedmiotem, na którym utrwalono utwór (na przykład płytą CD, dyskiem twardym, papierem).

Ustawa chroni wszystkie utwory o indywidualnym charakterze, bez względu na ich wartość artystyczną czy użytkową, i nie wymaga ani podejmowania specjalnych działań w celu ich rejestracji, ani umieszczania specjalnego znaku ©.

5.2.2. Rodzaje licencji programów komputerowych

W odniesieniu do programów komputerowych, oprócz przestrzegania przepisów prawa autorskiego, musimy postępować zgodnie z warunkami licencji udzielonymi na produkt. Nie wolno korzystać z programu, na który nie mamy licencji.

Definicja

Licencja na oprogramowanie to umowa na korzystanie z utworu, jakim jest aplikacja komputerowa. Umowa jest zawierana pomiędzy podmiotem, któremu przysługują majątkowe prawa autorskie do utworu, a osobą, która zamierza z danej aplikacji korzystać.

Licencja użytkownika to zbiór reguł, według których wolno korzystać z programu komputerowego. Zasady te są na ogół bardzo restrykcyjne. Wprowadzają ograniczenia, takie jak liczba komputerów, na których można zainstalować dany program, czy liczba użytkowników korzystających z programu jednocześnie.

Poniżej przedstawiono kilka przykładów popularnych typów licencji.

Freeware — programy mogą być używane bezpłatnie i bez żadnych ograniczeń. Prawa autorskie pozostają jednak w mocy. Dlatego nie można na przykład wprowadzać żadnych zmian w tych programach. Osoby trzecie nie mogą czerpać korzyści materialnych z dystrybucji tych programów. W podręczniku korzystaliśmy z oprogramowania na licencji freeware, na przykład z edytora języka XHTML kED.

Adware — jest to oprogramowanie rozpowszechniane za darmo, ale zawierające funkcję wyświetlającą reklamy. Przykładem tego typu oprogramowania jest komunikator Gadu-Gadu. Można zakupić wersję takiego programu bez reklam.

Shareware — program udostępniany bezpłatnie do testowania, dzięki czemu przed podjęciem decyzji o jego zakupie można sprawdzić, czy spełnia on nasze oczekiwania. Programy tego typu zwykle mają ograniczenia, jak limitowany czas testowania — najczęściej 30 dni. W chwili uruchomienia lub w trakcie pracy wyświetlają komunikaty przypominające o konieczności uiszczenia opłaty rejestracyjnej. Przykładem takiego programu jest Total Commander.

GNU GPL (ang. *General Public Licence*) — jedna z licencji tzw. wolnodostępowego oprogramowania. Użytkownik może uruchomić takie oprogramowanie w dowolnym celu, analizować jego działanie i dostosować je do swoich potrzeb, jak również kopiować oraz udoskonalać i publikować własne poprawki programów i kodu źródłowego.

Domena publiczna, czyli *public domain* — obejmuje oprogramowanie (a także inne utwory), do którego wygasły prawa autorskie i dzięki temu jest dostępne dla wszystkich. Można powiedzieć, że domena publiczna jest zbiorem informacji i wiedzy stanowiącym dorobek kulturalny i intelektualny wszystkich ludzi.

Wersja beta to prototypowe oprogramowanie, które jest testowane w celu wykrycia błędów i niedociągnięć. Ma również przybliżyć użytkownikowi prezentowany program i nakłonić go do kupna pełnej wersji.

Na szczególną uwagę zasługują tzw. *wolne licencje*.

5.2.3. Wolne licencje

Czym są wolne licencje? **Udostępnienie twórczości** na wolnych licencjach oznacza, że każdy ma prawo ją wykorzystywać, poprawiać, dostosowywać, powielać, rozprowadzać za darmo lub za opłatą oraz upowszechniać swoje udoskonalenia tak, aby mogła z nich korzystać szeroka grupa odbiorców.

Przygotowując się do lekcji, pisząc referat lub scenariusz, pracując nad prezentacją, każdy z nas może skorzystać z zasobów dostępnych w internecie na otwartych zasadach. Możemy na przykład:

- wykorzystać tłumaczenie artykułu — w całości lub we fragmencie — do pracy z uczniami;
- wykorzystać gotowy konspekt lekcji i dostosować go do potrzeb własnej lekcji;
- wykorzystać zdjęcia do prezentacji, referatu, ćwiczeń czy podręcznika;
- wykorzystać nagranie muzyczne jako podkład do animacji, filmu, prezentacji.

Wolne licencje to:

- **GNU GPL** (ang. *General Public License*) opisana krótko powyżej.
- **Domena publiczna** (ang. *public domain*). Jest to twórczość, z której można korzystać bez ograniczeń, ponieważ prawa majątkowe do niej wygasły lub twórczość ta nigdy nie była (lub nie jest) przedmiotem prawa autorskiego.

- **Licencje Creative Commons** (*CC*), które wykorzystują prawo autorskie i nie wymagają całkowitego zrzeczenia się praw autorskich.

Autor ma do wyboru cztery różne warunki udostępnienia swojego utworu. Podstawowe warunki licencji *CC* to:



Uznanie autorstwa. Wolno kopiować, rozprowadzać, przedstawiać i wykonywać objęty prawem autorskim utwór oraz opracowane na jego podstawie utwory zależne, pod warunkiem że zostanie przywołane nazwisko autora oryginału. Oznaczana również symbolem **BY:** w kółku.



Użycie niekomercyjne. Wolno kopiować, rozprowadzać, przedstawiać i wykonywać objęty prawem autorskim utwór oraz opracowane na jego podstawie utwory zależne jedynie do celów niekomercyjnych.



Na tych samych warunkach. Wolno rozprowadzać utwory zależne jedynie na licencji identycznej z tą, na jakiej udostępniono utwór oryginalny. Oznaczana również symbolem **SA**.



Bez utworów zależnych. Wolno kopiować, rozprowadzać, przedstawiać i wykonywać utwór jedynie w jego oryginalnej postaci — tworzenie utworów zależnych nie jest dozwolone.

Należy więc zwrócić uwagę na to, jaką licencję CC wybrał autor utworu, który chcemy wykorzystać.

Możemy spotkać się z sześcioma różnymi oznaczeniami wolnych licencji CC przedstawionymi na rysunku 5.9.



Rysunek 5.9. Oznaczenia wolnych licencji CC

Na polskiej stronie Creative Commons, *http://creativecommons.pl/*, można udostępnić swoją twórczość na jasnych, przyjaznych zasadach — na licencjach CC. Można tam również znaleźć muzykę, fotografię i filmy udostępnione na tej licencji.

5.2.4. Ustawa o prawie autorskim a szkolny referat

Opracowania, referaty, prezentacje szkolne przyjmują różne formy. Mogą być słowne, graficzne oraz te najczęściej stosowane — multimedialne. Do przygotowania prezentacji uczniowie wykorzystują między innymi: utwory literackie, plastyczne, fotograficzne, publicystyczne, naukowe oraz tzw. dobra pokrewne, którymi są utrwalone artystyczne wykonania utworów przez wokalistów, instrumentalistów i aktorów. Zdarza się również wykorzystywanie nagrań programów radiowych i telewizyjnych.

Zgodnie z ustawą z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych wszystkie te utwory i dobra pokrewne są chronione prawami wyłącznymi. Oznacza to, że na korzystanie z nich trzeba mieć zgodę właścicieli praw lub wskazać właściwy przepis prawa pozwalający na ich użycie. Artykuł 16. mówi, że twórcom przysługują w ra-

mach tych wyłącznych praw autorskie prawa osobiste, które są niezbywalne i niegasnące w czasie. Natomiast artykuł 17. mówi, że przysługują im autorskie prawa majątkowe, które wygasają po 70 latach od śmierci autora. A zatem bez zgody autora lub jego spadkobiercy utwór wolno wykorzystać dopiero 70 lat po jego śmierci (artykuł 36). Natomiast w przypadku dóbr pokrewnych — 50 lat od ich publikacji.

W niektórych przypadkach prawo zezwala na korzystanie z utworów i dóbr pokrewnych bez konieczności uzyskiwania zezwolenia i zapłaty wynagrodzenia. Według artykułu 27 do takich przypadków należy prezentacja szkolna lub referat, gdyż utwory i dobra pokrewne są wykorzystywane w celach dydaktycznych w ramach instytucji oświatowej. Sama prezentacja szkolna powinna być utworem przygotowanym samodzielnie i stanowić o indywidualnym podejściu do tematu. Nie powinna być dosłowną kopią materiałów źródłowych, a jedynie na nich się opierać i zawierać własne przemyślenia oraz ewentualne analizy, musi być bowiem przestrzegane również prawo cytatu (art. 29 ust. 1). Prawo cytatu zezwala na wykorzystanie urywków cudzych utworów lub drobnych utworów w całości w innym utworze, między innymi w celu wyjaśniania i analizy. Aby nie popełnić kolejnego przestępstwa, tzw. plagiatu (art. 115), należy zadbać o odpowiednią informację o tym, kto jest autorem wykorzystywanego dzieła, jaki jest jego tytuł i skąd został zaczerpnięty dany urywek lub cały drobny utwór, na przykład wiersz. Można korzystać z utworów w granicach dozwolonego użytku pod warunkiem wymienienia twórcy i źródła (art. 34). Informacja może być podana pisemnie (w postaci komentarza, przypisu), jak i słownie, na przykład podczas prezentacji lub po jej zakończeniu. Za niewłaściwą uznaje się prezentację (czy referat) składającą się wyłącznie z urywków cudzych utworów bez żadnego własnego wkładu intelektualnego.

Kiedy mamy do czynienia z celem dydaktycznym w odniesieniu do prac uczniowskich? Gdy prezentacja lub referat jest opracowana jako zadanie domowe i zaprezentowana tylko na lekcji w celu otrzymania oceny lub przekazania wiedzy innym uczniom. Jednak jeżeli to zadanie zostanie opublikowane w internecie, to aby nie naruszyć warunków zawartych w artykułach 27 i 34 trzeba bezwzględnie podać źródło wykorzystanych w pracy utworów oraz zamieścić informację o tym, że prezentację umieszcza się czasowo, by wykazać się umiejętnością wykonania takiego działania. Jednak po uzyskaniu oceny należy publikację usunąć.

5.2.5. Zadania

Zadanie 5.7. Na rysunku 5.9 zamieszczono symbole oznaczające sześć różnych licencji CC. Rozszyfruj je. Pomocne w rozwiązaniu tego zadania mogą być informacje umieszczone na stronie *http://creativecommons.org/about/licenses/*.

Zadanie 5.8. Sprawdź umowę licencyjną edytora tekstu Microsoft Office Word 2007 (w otwartym oknie edytora kliknij *Przycisk pakietu Office* i wybierz *Opcje programu Word*. W oknie dialogowym wybierz *Zasoby*, a w nich *Microsoft Office Word 2007*. W tej sekcji naciśnij przycisk *Informacje*). Odszukaj również licencje innych aplikacji zainstalowanych na Twoim stanowisku pracy.

Zadanie 5.9. W witrynach *http://prawo.interia.pl, http://prawo.ngo.pl* lub w innym miejscu w sieci odszukaj ustawę dotyczącą praw autorskich. Znajdź w niej artykuły, na które powołuje się podrozdział 5.2.4 podręcznika. Na podstawie rozdziału 3. tej ustawy podaj przykłady korzystania z utworów w ramach dozwolonego użytku prywatnego.

Zadanie 5.10. Zastanów się i odpowiedz na pytanie: kiedy należy zapytać sfotografowaną osobę o zgodę na opublikowanie jej wizerunku? Podaj stosowne przykłady.

Zadanie 5.11. Odszukaj w zasobach internetu znaczenie innych licencji, na przykład *trial*. Jaka jest różnica pomiędzy licencjami *trial* a *beta*?

Bibliografia

Na płycie CD wykorzystano utwory:

1. Brzechwa J., *Kłamczucha* [w:] tenże *WIELKA KSIĘGA POLSKIEJ POEZJI dla dzieci*, Papilon, Poznań 2008
2. Brzechwa J., *Leń* [w:] tenże *WIELKA KSIĘGA POLSKIEJ POEZJI dla dzieci*, Papilon, Poznań 2008
3. Brzechwa J., *Samochwała* [w:] tenże *WIELKA KSIĘGA POLSKIEJ POEZJI dla dzieci*, Papilon, Poznań 2008
4. Konopnicka M., *Dym*
5. Leśmian B., *Zachód*

Wydawnictwo Helion dołożyło wszelkich starań, by dotrzeć do właścicieli praw autorskich wszystkich tekstów literackich lub ich fragmentów, fotografii i zrzutów ekranowych zamieszczonych w podręczniku. Wydawnictwo Helion informuje, iż wyżej wymienione elementy zostały wykorzystane wyłącznie w celach edukacyjnych, na podstawie art. 29 ustawy o prawie autorskim i prawach pokrewnych z dnia 4 lutego 1994 r. (tekst jednolity z dnia 17 maja 2006 r. Dz.U. Nr 90, poz. 631 z późniejszymi zmianami).

CD-ROM zawiera:

W folderze *Rozdzial 1*:

- folder *Dane do cwiczen i zadan*: pliki z danymi do ćwiczeń i zadań opisanych w podręczniku;
- folder *Zadania maturalne*: treści wybranych zadań maturalnych z egzaminów i informatora maturalnego;
- folder *Zadania maturalne\Dane*: pliki z danymi do zadań maturalnych;
- plik *r01 OpenOffice.pdf*: ilustracje, dzięki którym można wykonać zawarte w podręczniku ćwiczenia i zadania w pakiecie biurowym Open.Office.org;
- folder *Excel algorytmy – ciekawe rozwiazania:* pliki z przykładowymi rozwiązaniami wybranych algorytmów z wykorzystaniem arkusza kalkulacyjnego.

W folderze *Rozdzial 2*:

- folder *Dane do cwiczen i zadan*: pliki z danymi do ćwiczeń i zadań opisanych w podręczniku;
- folder *Zadania maturalne*: treści wybranych zadań maturalnych z egzaminów i informatora maturalnego;
- folder *Zadania maturalne\Dane*: pliki z danymi do zadań maturalnych;
- plik *r02 OpenOffice.pdf*: instrukcje, dzięki którym można wykonać zawarte w podręczniku ćwiczenia i zadania w pakiecie biurowym Open.Office.org.

W folderze *Rozdzial 3*:

- pliki zawierające rozszerzenie treści zawartych w podręczniku: *systemy plikow.pdf, systemy operacyjne.pdf, swiatłowody.pdf, specyfikacje sieci komputerowej.pdf, obsluga i konfiguracja sieci w Windows.pdf, orange book.pdf*;
- folder *Cwiczenia*: opis i niezbędne oprogramowanie do wykonania ćwiczeń 3.1, 3.2, 3.6 i 3.16;
- folder *Hosting*: pliki instalacyjne wraz z dodatkowymi modułami do instalacji CMS na zdalnym serwerze;
- plik *hosting CMS.pdf*: opis publikacji CMS — serwis Drupal i Gallery2 na serwerze *http://byethost.com*;
- plik *publikowanie strony WWW.pdf*: opis publikacji strony WWW na serwerze *onet.pl*;
- folder *Literatura*: przykładowa witryna z zastosowaniem zewnętrznych kaskadowych arkuszy stylów;
- folder *PHP*: pliki źródłowe programów opisanych w podrozdziale 3.5.8, *Czym jest PHP?*.

W folderze *Rozdzial 4*:

- folder *Dane do cwiczen i zadan*: pliki z danymi do zadań opisanych w podręczniku;
- plik *zdjecia_legitymacyjne.pdf*: opis ćwiczenia 4.1;
- plik *transformacje.pdf*: opis ćwiczenia 4.2;
- pliki zawierające rozszerzenie treści zawartych w podręczniku *skaner.pdf, LZW.pdf, Huffman.pdf, konstruktywna geometria bryl.pdf*;
- folder *Kody programow*: pliki źródłowe programów opisanych w podręczniku — grafika 3D: **.pov;*
- folder *Filmy*: pliki z nagraniami filmów do ćwiczeń i zadań zawartych w podręczniku;
- folder *Fotografie*: zdjęcia do ćwiczeń i zadań zawartych w podręczniku;
- folder *Pliki dzwiekowe*: pliki z nagraniami do ćwiczeń i zadań zawartych w podręczniku;
- folder *Programy*: wykorzystywane w podręczniku oprogramowanie.

Skorowidz

A

Access, 38, 43
adres
  bezklasowy, 168, 170
  broadcastowy, 156
  domyślny, 168
  fizyczny MAC, 155
  IP, 156, 163, 164, 180, 189
  IPv4, 164, 168
  IPv6, 166
  multicast, 157
  niepubliczny, 168
  nierutowalny, *Patrz:* adres niepubliczny
  prywatny, *Patrz:* adres niepubliczny
  publiczny, 167
  rozgłoszeniowy, 167, 170
  sieci, 167, 169
  specjalny, 167
  URL, 193
adware, 189, 301
aliasing, 229
anomalia, 44, 45
antyaliasing, 229
architektura klient-serwer, 182
arkusz danych, 65
arkusz kalkulacyjny, 9, 71
  adres, 10
  format komórki, 10
  formuła, 10
  tabela, 10
atak
  DDoS, 190
  DoS, 190
  poprzez słownik, 184
atrybut, 11
autofiltr niestandardowy, 18
autofiltrowanie, 14
autoformularz, 103

B

barwa, 230, 234, 236
  achromatyczna, 236
  prosta, 232
baza danych, 11, 35, 37, 43
  dekodowanie, 114
  hierarchiczna, 37
  jednotabelowa,
    Patrz: baza danych kartotekowa
  kartotekowa, 9, 36
  kodowanie, 114
  multimedialna, 37
  obiektowa, 37
  obiektowo-relacyjna, 37
  plik, 43
  relacyjna, 35, 37
  sieciowa, 37
  typy, 36
  złożona, 37
Bell Alexander Graham, 271
Bézier Pierre Étienne, 254
bezklasowy routing międzydomenowy,
  Patrz: CIDR
bezpieczeństwo danych,
  Patrz: dane ochrona i bezpieczeństwo
BIOS, 122, 123
bitrate, 275

C

CIDR, 167, 169
cloud computing,
  Patrz: przetwarzanie w chmurze
CMS, *Patrz:* system zarządzania treścią
cookies, 178

COPS, 184
CPU, *Patrz:* procesor
CSG, *Patrz:* konstruktywna geometria brył
CSS, *Patrz:* kaskadowy arkusz stylów
CSV, 14
częstotliwość bitowa, *Patrz:* bitrate

D

dane
  autonumerowanie, 49
  data/godzina, 49
  hiperłącze, 50
  liczba, 48, 51
  nota, 48
  obiekt OLE, 49, 64
  ochrona i bezpieczeństwo, 112, 113, 183, 186
  tak/nie, 49
  tekst, 48, 51
  typy, 47, 57
  waluta, 49
  załącznik, 50
Database Management System, *Patrz:* system zarządzania bazą danych
datagram, 161, 162, 163
DB2, 38
decybel, 271
dekodowanie, *Patrz:* baza danych dekodowanie
diagram obiektowo-związkowy, 41
dokument HTML, 193
domena publiczna, 301
DOS, *Patrz:* system operacyjny DOS
Drupal, 208
drzewo, 244
  binarne, 244
  korzeń, 244
dynamika dźwięku, 273
dysk twardy
  HDD, 294
  SSD, 294
dziedziczenie, 205
dźwięk, 269, 271
  analogowy, 271
  cyfrowy, 271

E

eksport, 70, 73
e-learning, 298
enkoder, 275
Ethernet, 152
exploit, 188

F

filtr zaawansowany, 20
filtrowanie, 15, 65, 66
  według formularza, 68
  z wykluczeniem wyboru, 16
firewall, *Patrz:* zapora sieciowa
folder, 127
formant, 105, 108, 190
format
  AAC, 280
  AIFF, 274
  AVI, 282
  bajtowo-dziesiętny, 164
  BMP, 247
  EPS, 257
  FLAC, 280
  GIF, 241, 247, 250
  JFIF, 248
  JPG (JPEG), 247, 251
  MOV, 282
  MP3 (MPEG), 274, 275, 276
  MPEG, 282
  OGG, 279
  PNG, 241, 248, 252,
  RealMedia, 282
  RGB24, 282
  SVG, 199, 257
  TIFF, 248
  WAVE, 274
formularz, 12, 14, 43, 103
  dzielony, 104
freeware, 300

G

GIMP, 228
głębia bitowa, 237
GNU GPL, 301
GPU, 265
graficzny interfejs użytkownika,
  Patrz: GUI
grafika, 225
  2D, 225
  3D, 225, 258
  bitmapowa, *Patrz:* grafika rastrowa
  komputerowa, 225
  płaska, *Patrz:* grafika 2D
  przestrzenna, *Patrz:* grafika 3D
  rastrowa, 226, 257
  wektorowa, 253, 256, 257
GUI, 127

H

haker, 183
hasło, 113, 183, 184
hiperlink, 193
hipertekst, 192
host, 164, 169
HSPA, 152

I

identyfikacja danych, 41
IEEE, 152, 157
import, 70, 71
indeksowanie, 54
infradźwięki, 270
inżynieria społeczna, 188
IPtrace, 184
ISO, 157, 158

J

jaskrawość, 234, 237
jasność, 237
jednobarwność,
  Patrz: monochromatyczność
jednostka centralna, 117
język
  HTML, 193, 194
  PHP, 210
  skryptowy, 210
  XHTML, 194, 195
  XML, 194
język zapytań,
  QBE, 74
  SQL, 74, 95, 101

K

kabel
  ekranowany STP, 145, 152
  nieekranowany UTP, 145
kamera, 260, 263
karta rozszerzeń, 121
kaskadowość, 205
kaskadowy arkusz stylów, 202
katalog, 127
klaster, 128
klient, 193
klucz
  główny, 41, 49, 55, 56
  obcy, 42, 56
  podstawowy, *Patrz:* klucz główny
  własny, *Patrz:* klucz główny
kod
  Huffmana, 244
  prefiksowy, 243
  szesnastkowy, 249
kodek, 274, 279-283
kodowanie, *Patrz:* baza danych kodowanie
kolor, 230, 249
  indeksowany, 251
koło chromatyczne, 234
komórka, 161
kompatybilność, 117
kompresja
  bezstratna, 240, 274
  stratna, 240, 246, 251, 274, 278
komunikat, 161
  o błędzie, 53
  sieciowy, 167

koncentrator, 145, 154
konstruktywna geometria brył, 265
kontener multimedialny, 282
konwersja, 272
koń trojański, 184, 188, 189
kreator odnośników, 50
krotka, 44
kwantyzacja, 272
kwerenda, 43, 68, 74
  aktualizująca, 88
  dołączająca, 88, 90
  funkcjonalna, 88, 90
  krzyżowa, 88, 92, 100
  oparta na jednej tabeli, 95
  parametryczna, 86, 99
  tworząca tabelę, 88, 89, 100
  usuwająca, 88, 91
  wybierająca, 75, 88, 97

L

LanWatch, 184
licencja, 300, 302
link, 201
Linux, *Patrz:* system operacyjny Linux
logo, 256
LTE, 153
LZW, 241

M

magistrala, 121, 293
  AGP, 121, 265
  PCI, 121, 265
  PCI Express, 121, 265
  QuickPath, 294
  USB 3.0, 296
  wejścia-wyjścia, 121
makro, 43, 110
makropolecenie, *Patrz:* makro
malware, 187
maska sieci, 168
maska wprowadzania, 51, 52
maszyna wirtualna, 131
memrystor, 295
Microsoft SQL Server, 38
model
  barw, 231
  CIE La*b*, 231
  CIE XYZ, 231
  CMYK, 233
  HLS, 234
  HSB, 234
  HSV, 234
  OSI, 158,162, 177, 182
  RGB, 232, 234
modeler, 259
moduł, 43
monochromatyczność, 237
most, 155
mostek
  południowy, 121, 122
  północny, 121, 122, 293
MySQL, 38, 208

N

nadmiarowość danych, 38
nasycenie, 234, 235, 236
NAT, 167
notacja dziesiętna z kropkami, 164
null, 48
numer sekwencyjny, 161

O

obrazek, 250
obrót, 264
ochrona danych,
  Patrz: dane ochrona i bezpieczeństwo
odcień, 234
odnośnik hipertekstowy, 201
odsyłacz, 201
okablowanie strukturalne, 152
oprogramowanie złośliwe, *Patrz:* malware
Oracle Universal Server, 38
oryginał analogowy, 240
osadzanie obiektu, 64

P

pakiet, 161
pamięć
  DRAM, 296
  flash, 295
  operacyjna, 118
  RAM, 118, 293
phishing, 191
piksel, 226, 239
plik, 43, 126
płyta główna, 120, 121,
podformularz, 105
podsieć, 174
pole, 11, 57, 64, 68, 99
  kluczowe, 46, 54, 55, 56
  rozmiar, 51
połączenie punkt-punkt, 293
port, 177
PostgreSQL, 38
potomek, 244
prawo autorskie, 300, 302
procesor, 118, 119, 121, 293
procesor graficzny, *Patrz:* GPU
program antywirusowy, 187
protokół, 160
  bezstanowy, 178
  DHCP, 180
  DNS, 180
  FTP, 177, 179
  HTTP, 177, 178, 185, 193
  HTTPS, 178, 185, 188
  IMAP, 179
  IP, 162, 164, 181
  IPv6, 166, 175, 176
  klient-serwer, 179
  nierutowalny, 177
  PGP, 186
  POP3, 177, 179, 188
  rutowalny, 177
  scentralizowany, 186
  SFTP, 179
  SMTP, 177, 179, 188
  SSH, 181
  SSL, 178, 185
  TCP, 162, 164
  TCP/IP, 162, 164, 177, 183
  telnet, 181, 184
próbkowanie, 272, 273
przeglądarka internetowa, 193
przełącznik, 145, 155, 189
przepływność, 152, *Patrz też:* bitrate
przepustowość, 152
przetwarzanie w chmurze, 297
przezroczystość, 247, 250, 253
punkt dostępowy Wi-Fi, 186

R

radiosity, 259
ramka, 158, 159, 161, 162
raport, 27, 43, 108
ray tracing, 259
redundancja, 44, 45
reguła sprawdzania poprawności, 53
rekord, 11, 44
relacja, 57
  jeden-do-jednego, 57, 58
  jeden-do-wielu, 57, 60
  wiele-do-wielu, 57, 61
renderowanie, 263
RLE, 240
robak, 188
rodzic, 244
rootkit, 189
router, 156, 163, 186
rozdzielczość, 226

S

segment, 161, 162
sektor, 128
serwer WWW, 182, 193, 206,
shareware, 301
sieć bezprzewodowa, 186
sieć komputerowa, 143, 144
  bezawaryjność, 182
  instalacja, 182
  konfiguracja, 182

LAN, 144, 155, 156
MAN, 144
WAN, 144
skalowanie, 264
skaner, 188
sniffing, 188
Snoop, 184
sortowanie danych, 23, 54, 65
spoofing, 189
spójnik
alternatywy, 18
koniunkcji, 18
sprzężenie, 57, 58
spyware, 189
SQLite, 38
strona, 43
strumień danych, 161
sumy
częściowe, 25, 29
pośrednie, *Patrz:* sumy częściowe
Sybase SQL Server, 38
sygnet, 256
system
plików, 127, 128
zarządzania bazą danych, 35, 37
zarządzania treścią, 207
system operacyjny, 124, 126, 129
czasu rzeczywistego, 130
dla jednego użytkownika, 130
DOS, 125, 133, 137
jądro, 125
jednozadaniowy, 129
Linux, 136
powłoka, 125
rozproszony, 130
równoległy, 130
wielodostępny, 130
wielozadaniowy, 129
szum, 277
szybkie filtrowanie, 66
szybkie sortowanie, 65
szyfrowanie, 178, 181, 183, 185, 186, 188, 189,
Patrz też: baza danych kodowanie
szyna systemowa, *Patrz:* magistrala

Ś

ścieżki przeszukiwań, 132
światłowód, 152

T

tabela, 43, 44, 45, 63, 70
kryteriów, 20, 21
połączona, 70
przestawna, 27
załączanie, 70, 72
token, 148
Token-Ring, *Patrz:* topologia pierścienia
tony
niskie, 270
średnie, 270
wysokie, 270
topologia
drzewa, 149
fizyczna, 144
gwiazdy, 145
komórkowa, 150
logiczna, 144
magistrali, 146
mieszana, 151
pierścienia, 147
podwójnego pierścienia, 149
transformacja, 264
translacja, 264
transmisja
grupowa Multicast, 157
pojedyncza Unicast, 156
rozgłoszeniowa Broadcast, 157
trasowanie, 163
trojan, *Patrz:* koń trojański
tryb
konsoli, 136
tekstowy, *Patrz:* tryb konsoli

U

UEFI, 124
układ współrzędnych, 260
ultradźwięki, 270

urządzenia
peryferyjne, 118, 119
wejścia-wyjścia,
Patrz: urządzenia peryferyjne
zewnętrzne,
Patrz: urządzenia peryferyjne

W

warstwa
aplikacji, 158, 160, 161, 162, 163, 179, 182
dostępu do sieci, 162, 163
fizyczna, 158, 163
internetu, 162, 163
łącza danych, 158, 161
prezentacji, 158, 159
procesowa, *Patrz:* warstwa aplikacji
programów użytkowych,
Patrz: warstwa aplikacji
protokołu internetowego,
Patrz: warstwa internetu
sesji, 158, 159
warstwa sieciowa, 156, 158, 159, 161
transportowa, 158, 159, 161, 162, 163
zerowa, 158
wartość domyślna, 53
wersja beta, 301
węzeł, 193, 244
wideo, 281
wirus, 188
witryna WWW, 193, 206
WWW, 178, 192, 193
wykres, 26
przestawny, 29
wyrażenie, 82
wzmacniak, 153, 154

Z

zapora sieciowa, 187, 189
zapytanie, *Patrz:* kwerenda
zasoby, 128
zatruwanie DNS, 191
zerowa długość dozwolona, 54
zgodność wsteczna, *Patrz:* kompatybilność
złącze
AT-BUS, 122
cylindryczne BNC, 146, 152
LPT, 122
PCI, 122
PS/2, 122
Serial ATA, 122
USB, 122
zombi, 190

Ź

źródło światła, 260, 263

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!
http://program-partnerski.helion.pl

GRUPA WYDAWNICZA
Helion SA