

Historia Języka C

- Idea C pochodzi od języka BCPL, opracowanego przez M. Richardsa
- 1970 – język B stworzony przez K. Thompsona
- Język C wprowadzono po raz pierwszy z systemem Unix w wersji 5; został wymyślony przez D. Ritchie
- 1978 – pierwsza książka – opis języka: B. Kernighan, D. Ritchie: The C Programming Language
- 1989 – wprowadzenie standardu ANSI C, zaadaptowanego przez ISO
- 1995 – uzupełnienie 1 standardu i zdefiniowanie podzbioru C języka C++
- 1999 – standard języka C99

Cechy języka C (1)

- Język średniego poziomu – łączy w sobie najlepsze cechy języków wysokiego poziomu (BASIC, Pascal, Ada) z możliwościami sterowania i elastycznością charakterystyczną dla assemblera. (Tzw. Assembler wysokiego poziomu)
- Pozwala operować na bitach, bajtach i adresach, przy zachowaniu dużej przenośności
- Nie wykonuje prawie żadnego sprawdzania błędów wykonania (np. przekroczenia granicy tablicy)
- Nie wymaga ścisłej zgodności typów między parametrem, a argumentem
- Możliwość bezpośredniego operowania na bitach, bajtach słowach i wskaźnikach (predysponowany do programowania systemowego).
- Niewielka liczba słów kluczowych (C89: 32, C99: 37).
- C to język strukturalny
 - Możliwość definiowania podprogramów korzystających ze zmiennych lokalnych, będących odizolowanymi fragmentami niewrażliwymi na pozostałe części programu.
 - Bezpośrednio obsługuje kilka konstrukcji pętli (while, do-while, for)
 - Główny składnik strukturalny w języku C to funkcja
 - Istnieje również możliwość definiowania bloków

Cechy języka C (2)

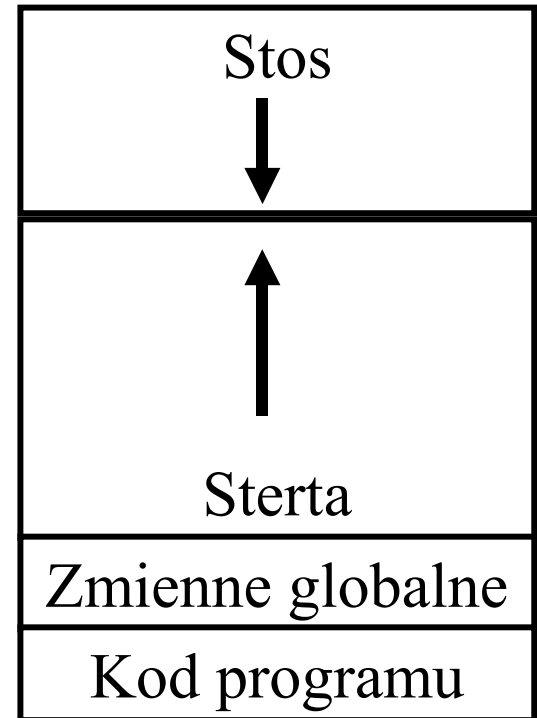
- **C to język programistów**
- **Zastosowania:**
 - Programowanie systemowe (usługi UNIX, LINUX, Windows)
 - Programowanie aplikacji
 - Wprowadzenie C++ nie usunęło języka C do lamusa, wciąż istnieje szeroka pula zagadnień, w których język C pozostaje dominujący:
 - Systemy wbudowane (sterowniki mikroprocesorowe, aplikacje czasu rzeczywistego)
 - Oprogramowanie na platformę Linux
 - Istnieje stworzona pokaźna pula programów napisanych w języku C, które wciąż pracują, i które trzeba pielęgnować
- **Język C++ przejął standard języka C, stąd dobre opanowanie języka C umożliwia szybkie opanowanie C++.**

Tworzenie typowego programu w języku C

- **Zwykle stosuje się standardową bibliotekę funkcji, które wspomagają proces tworzenia aplikacji (obsługa wejścia/wyjścia programu, funkcje matematyczne, przetwarzanie tekstów)**
- **Kod programu przekształcany jest na kod maszynowy (kompilacja), następnie łączony z kodem bibliotek, które zastosowano (konsolidacja) otrzymując program**
- **Można przygotować aplikację złożoną z wielu plików źródłowych, które kompilowane są osobno**

Mapa pamięci języka C

- Skompilowany program tworzy 4 osobne obszary pamięci:
 - Obszar kodu programu (instrukcje)
 - Obszar zmiennych globalnych (zdefiniowane poza funkcjami, widoczne dla wszystkich funkcji zdefiniowanych „pod” nimi)
 - Stos – przechowywanie zmiennych lokalnych, zachowywanie stanu rejestrów przy wywoływaniu podprogramów itd.
 - Sberta – obszar wolnej pamięci, którą program może dynamicznie zaalokować.



Podstawowe elementy języka C

- Zestaw znaków
- Nazwy i słowa zastrzeżone
- Typy danych
- Stałe
- Zmienne i tablice
- Deklaracje
- Wyrażenia
- Instrukcje

Zestaw znaków języka C

- Duże litery alfabetu łacińskiego [A..Z]
- Małe litery alfabetu łacińskiego [a..z]
- Cyfry [0..9]
- Znaki specjalne:
! * + \ " < # (= | { > %) ~ ; } / ^ - [: , ? & _] ' oraz znak odstępu (spacja)
- UWAGA:
Nowe narzędzia do tworzenia oprogramowania zezwalają nawet na tworzenie nazw zmiennych i funkcji z zastosowaniem narodowych znaków diakrytycznych. Kod nie będzie wtedy zgodny ze starszymi wersjami standardu ANSI C, co zmniejszy jego przenoszalność.

Nazwy i słowa zastrzeżone

- **NAZWA** służy do identyfikowania elementów programu (stałych, zmiennych, funkcji, typów danych)
- Nazwa składa się z ciągu liter i cyfr, z zastrzeżeniem, że pierwszym znakiem nazwy musi być litera. Znak podkreślenia `_` traktowany jest jako litera
- Język C rozróżnia duże i małe litery!
- W języku C zdefiniowano tzw. **SŁOWA ZASTRZEŻONE (KLUCZOWE)**, posiadające szczególne znaczenie dla języka. Tych słów nie wolno użyć w programie jako nazw (np. zmiennych, stałych, funkcji itd.)

Słowa kluczowe języka C

C89:

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

C99:

_Bool	_Imaginary	restrict
_Complex	inline	

Podstawowe typy danych

- **int** – reprezentuje liczbę całkowitą
 - **char** – reprezentuje małą liczbę całkowitą o rozmiarze wystarczającym do przechowania pojedynczego znaku
 - **float** – reprezentuje liczbę rzeczywistą (reprezentowaną w kodzie koprocatora)
 - **double** – reprezentuje liczbę rzeczywistą o podwójnej precyzji
 - Długość danych zależy od implementacji
 - Wprowadzono również tzw. modyfikatory typów danych:
 - **short** – liczba krótka
 - **long** – liczba długa
 - **signed** – liczba ze znakiem
 - **unsigned** – liczba bez znaku
- Np.: unsigned int, long int (long), long double...

Stałe (1)

- **Stałe całkowitoliczbowe:**
 - Stałe dziesiętne (dozwolony zestaw znaków: 0 1 2 3 4 5 6 7 8 9 + -), np.:
0 1 897 -234 +665
 - Stałe ósemkowe (dozwolony zestaw znaków: 0 1 2 3 4 5 6 7 + -),
Uwaga: pierwszą cyfrą musi być 0,
np.: 0 0122 -0777 +0234
 - Stałe szesnastkowe (dozwolony zestaw znaków: 0 1 2 3 4 5 6 7 8 9 a b c d e f A B C D E F + -),
Uwaga: pierwszymi znakami muszą być 0x lub 0X,
np.: 0x 0xad3 0X233F
- **Stałe rzeczywiste (dozwolony zestaw znaków: 0 1 2 3 4 5 6 7 8 9 . + - E e),**
litera E lub e reprezentuje bazę systemu, tj. 10,
Uwaga: 1.2×10^{-3} można zapisać 1.2e-3 lub 1.2E-3,
np.: 0. 0.2 1.123 13.13E2
- **Stałe znakowe – pojedyncze znaki „zamknięte” pomiędzy apostrofami: ‘ ‘**
Uwaga: Stałe znakowe są w istocie kodami liter i innych znaków zgodnymi z ASCII lub UNICODE
np.: ‘A’ ‘#’ ‘ ’(spacja)

Stałe (2)

- **Escape-sekwencje** – kody znaków niedrukowanych służących do podstawowego formatowania wyjścia znakowego programu lub plików tekstowych,
Uwaga: znak sekwencji rozpoznawany jest po tym, że składa się z 2 znaków, w tym pierwszy jest zawsze backslash (\),
np.: \n \t \" \' \? \\ \0
- **Łańcuchy znaków** – stała łańcuchowa (tekstowa) składa się z ciągu o dowolnej liczbie znaków. Ciąg ten mus być ograniczony znakami udzysłowu. Łańcuchy mogą zawierać escape-sekwencje.
Np.: "Wynik =" "To jest element \n tekstu"
- **Stałe symboliczne** – nazwa zastępująca łańcuch znaków. Do definicji złąży pseudoinstrukcja #define
np.:
#define NAZWA text
#define ROZMIAR_PAMIECI 1024

Zmienne

- Zmienna to nazwa (identyfikator) reprezentująca określony typ danych.
- Deklaracja zmiennej:
int a;
float x1, x2, x3;
char c = 'A';

Instrukcje

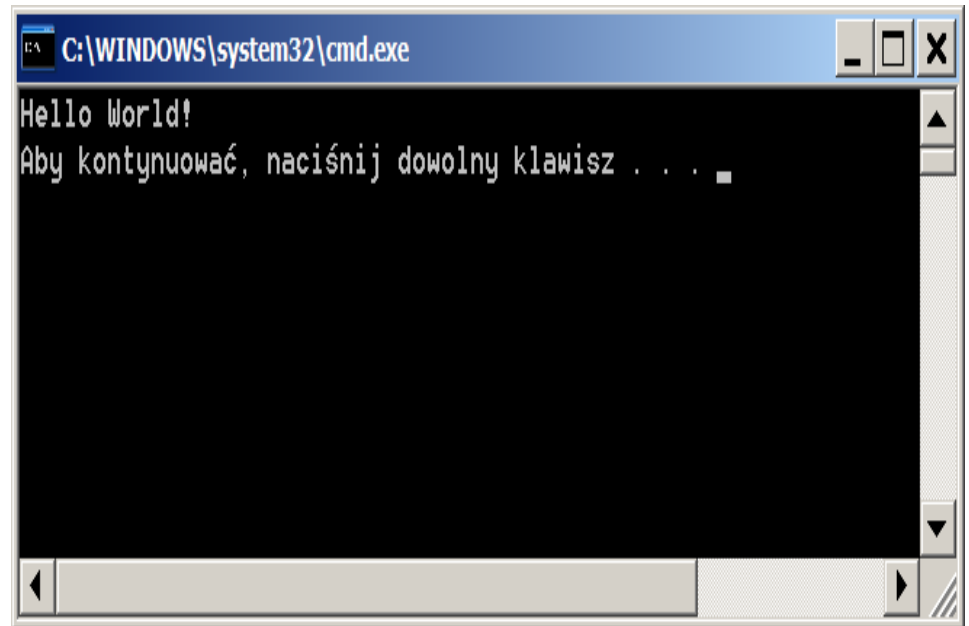
- Instrukcje to te fragmenty programu, które powodują jakąś czynność (akcję) komputera w trakcie wykonywania programu.
- Instrukcje można podzielić na cztery grupy:
 - Instrukcje obliczające wartość wyrażeń
np.: `a = 3 + b;`
 - Instrukcje grupujące
np.:

```
{ a = 5;  
  b = 8;  
  pole = a * b;  
  printf("pole=%d", pole);  
}
```
 - Instrukcje sterujące, np.: `while`, `if`...
 - Instrukcje wywołania funkcji.

Podstawowy szablon aplikacji w C

```
#include <stdio.h>    // Komentarz do końca
                      //linijki

void main(void)
{
    /*komentarz standardowy*/
    printf("Hello World!\n");
}
```



Podstawowy szablon aplikacji w C

```
#include <stdio.h>    // Komentarz do końca
                      //linijki

void main(void)
{
    /*komentarz standardowy*/
    printf("Hello World!\n");
}
```

