

Ćwiczenie I – Wprowadzenie do programowania w Windows.

Przygotowanie do ćwiczenia:

1. Zapoznać się ze sposobem konstruowania aplikacji Windows.

Przebieg ćwiczenia:

1. Uruchomić programu szablon1.exe i zapoznać się z jego obsługą.
2. Stworzyć własną aplikację Windows na podstawie programu szablon:
 - a. Utworzyć nowy projekt,
 - b. Stworzyć menu dla programu,
 - c. Dołączyć plik zasobów do projektu,
 - d. Przekopiować plik źródłowy „szablon1.c” do folderu z plikami nowego projektu
 - e. Dołączyć plik szablon1.c do projektu,
 - f. Jeśli zachodzi konieczność, zmodyfikować stałe symboliczne przywiązane do opcji menu w taki sposób, żeby były one identyczne w pliku szablon1.c i w edytorze zasobów (chodzi o stałe symboliczne ID_PLIK_KONIEC i IDR_MENU1),
 - g. Skompilować i uruchomić program.
3. Utworzyć własną aplikację konsoli (np. Prosty program „Hello World”).
4. Zmodyfikować aplikację Windows w taki sposób, że stworzona zostanie dodatkowa opcja menu (np. „Zmień napis”) i pod wpływem jej wybrania zmieniać się treść napisu umieszczonego w oknie:
 - a. Utworzyć nową opcję w graficznym edytorze menu
 - b. Utworzyć w programie statyczną zmienną, która będzie przechowywać informację, który z napisów ma być wyświetlany (np. zmienna *flaga* równa 0 oznacza, że ma być wyświetlony napis „Pozdrowienia” a zmienna *flaga* równa 1 oznacza, że ma być wyświetlany napis „Cześć”).
 - c. Uzupełnić obsługę komunikatu WM_COMMAND o fragment kodu odpowiedzialny za zmianę wartości zmiennej *flaga*:

```
case WM_COMMAND: // Komunikat przychodzący po wybraniu
                  // opcji menu itp
switch(LOWORD(wParam))
{
    // Wybrano opcję menu Plik->Koniec:
    case ID_PLIK_KONIEC:
        DestroyWindow(hwnd);
        return 0;
    case ID_PLIK_ZMIEN_NAPIS:
        flaga=(flaga==0)?1:0;
        InvalidateRect(NULL, NULL, TRUE);
        return 0;
}
```

Fragment kodu programu zaznaczony na czerwono powoduje, że przy każdorazowym wybraniu opcji menu o identyfikatorze ID_PLIK_ZMIEN_NAPIS

stan zmiennej *flaga* ulegnie zmianie z 0 na 1 lub z 1 na 0. Wywołanie funkcji `InvalidateRect` spowoduje wywołanie komunikatu `WM_PAINT` i odrysowanie całej zawartości okna programu.

- d. Obsługa komunikatu `WM_PAINT` musi również zostać zmodyfikowana:

```
case WM_PAINT: // Komunikat przychodzi w przypadku konieczności
                // "odmalowania" lub namalowania okna
                // Wymaż okno, uzyskaj kontekst urządzenia:
                HDCPaint=BeginPaint(hwnd,&PaintStruct);
                // Pobierz rozmiary obszaru roboczego okna
                GetClientRect(hwnd,&rect);
                // Umieść centralnie w oknie tekst:
                if(flag==0) DrawText(    HDCPaint,
                                        "Pozdrowienia",
                                        -1,
                                        &rect,
                                        DT_SINGLELINE|
                                        DT_CENTER|
                                        DT_VCENTER);
                else DrawText(    HDCPaint,
                                "Cześć",
                                -1,
                                &rect,
                                DT_SINGLELINE|
                                DT_CENTER|
                                DT_VCENTER);

                // Zakończ rysowanie
                EndPaint(hwnd, &PaintStruct);
                return 0;
```

Przy każdym rysowaniu okna sprawdzany jest stan zmiennej *flaga* i w zależności od stanu zmiennej wypisywany jest odpowiedni napis.

Uwaga:

W przyszłości mechanizm podobny do zaproponowanego powyżej posłuży do generowania kolejnych klatek animacji 3D.