

# Windows Application Programming Interface

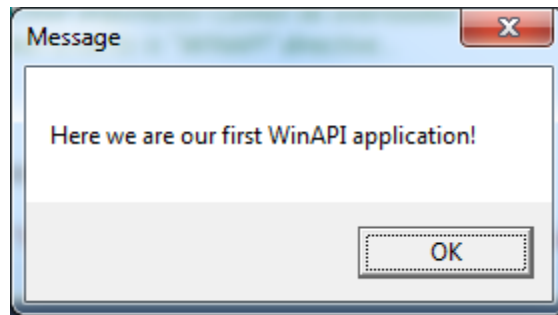
Tomasz Dzieciak

Torun, 18.01.2012r.

## Plan prezentacji:

- Podstawy WinAPI
  - pierwszy program
    - funkcja WinMain()
    - podstawy MessageBox-a
  - pierwsze okno
    - przechwyt komunikatów
    - a co ze śmieciami?
    - klasa okna
    - tworzenie okna
    - pętla komunikatów

# Podstawy WinAPI - pierwszy program



pierwszy program - funkcja WinMain()

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR  
lpzCmdLine, int nCmdShow)
```

hInstance - uchwyt instancji programu, jest to liczba, która jednoznacznie identyfikuje program

hPrevInstance - uchwyt do poprzedniej instancji, obecnie nieużywany (= NULL)

lpzCmdLine - argumenty wiersza poleceń

nCmdShow - sposób wyświetlania okna głównego

pierwszy program - podstawy MessageBox-a

```
int MessageBox(HWND hWnd, LPCTSTR lpText, LPCTSTR lpCaption, UINT uFlags);
```

hWnd - uchwyt okna nadrzędnego do komunikatu

lpText - tekst komunikatu

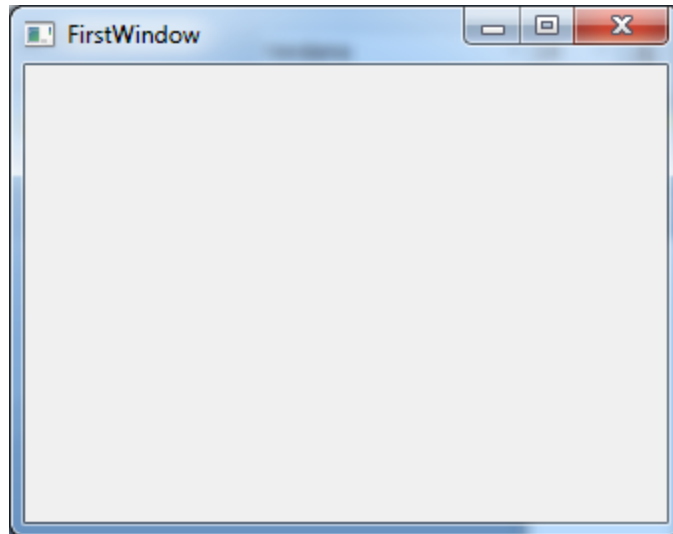
lpCaption - tytuł komunikatu

uFlags - dodatkowe parametry okna komunikatu

## podstawy MessageBox-a - uFlags

- ∅ rodzaj przycisków, jakie pojawią się w oknie
- ∅ domyślny przycisk (jeżeli do wyboru jest kilka)
- ∅ ikonkę, jaką ma być opatrzony komunikat
- ∅ modalność okna komunikatu
- ∅ sposób wyświetlania (wyrównanie) tekstu zawiadomienia
- ∅ parametry samego okna komunikatu

## Podstawy WinAPI - pierwsze okno



## pierwsze okno - przechwyt komunikatów

```
LRESULT CALLBACK WindowEventProc(HWND hWnd, UINT uMsg,  
WPARAM wParam, LPARAM lParam);
```

hWindow - uchwyt okna, w którym wystąpiło zdarzenie

uMsg - rodzaje zdarzenia (kilkaset! stałych)

wParam - 4-bajtowa liczba przechowująca dodatkowe informacje o zdarzeniu

lParam - j/w, rzadziej używane



pierwsze okno - a co ze śmieciami?



```
return DefWindowProc(hWindow, uMsg, wParam, lParam);
```

## pierwsze okno - klasa okna

```
struct WNDCLASSEX
{
    UINT cbSize;                // rozmiar struktury
    HINSTANCE hInstance;        // uchwyt instancji programu
    LPCTSTR lpszClassName;      // nazwa klasy
    WNDPROC lpfnWndProc;        // wskaźnik na procedurę zdarzeniową
    UINT style;                 // styl okna
    HICON hIcon;               // ikonka
    HICON hIconSm;             // mała ikonka (pasek start, tray)
    HCURSOR hCursor;           // kursor myszy
    HBRUSH hbrBackground;      // kolor tła okna
    LPCTSTR lpszMenuName;       // wskaźnik na strukturę menu
    int cbClsExtra;            // dodatkowa pamięć dla klasy okna
    int cbWndExtra;            // dodatkowa pamięć dla instancji
};
```

## pierwsze okno - klasa okna

```
HWND CreateWindowEx(  
    DWORD dwExStyle,           // opisuje rozszerzony styl okna  
    LPCTSTR lpClassName,      // nazwa klasy okna  
    LPCTSTR lpWindowName,     // tytuł okna  
    DWORD dwStyle,            // styl okna - pobawimy się!  
    int x,                     // współrzędna pozioma okna  
    int y,                     // współrzędna pionowa okna  
    int nWidth,                // szerokość okna  
    int nHeight,              // wysokość okna  
    HWND hWndParent,          // uchwyt do okna nadrzędnego  
    HMENU hMenu,              // uchwyt menu  
    HINSTANCE hInstance,     // uchwyt do instancji programu  
    LPVOID lpParam            // dodatkowy parametr  
);
```

## pierwsze okno - pętla komunikatów

```
MSG msgKomunikat;  
while (GetMessage(&msgKomunikat, NULL, 0, 0))  
{  
    TranslateMessage (&msgKomunikat);           // tłumaczy komunikaty  
    DispatchMessage (&msgKomunikat);          // wysyła do odpowiedniego okna  
}
```