

# Zestaw 1

Win32API, elementy interfejsu użytkownika

3 marca 2004

## Streszczenie

Rozwiązanie zadań w tym zestawie polega na napisaniu programów w języku C, przy czym w programach wolno korzystać wyłącznie z funkcji bibliotek standardowych C oraz Win32API. Tam gdzie to możliwe należy wybierać funkcje z Win32API zamiast ich odpowiedników z C (na przykład przy obsłudze systemu plików czy alokacji pamięci). Do tworzenia i obsługi okien nie wolno wykorzystywać żadnych interfejsów pośrednich, typu MFC, wxWindows czy GTK.

1. Napisać program, który podczas próby zamknięcia okna poprosi użytkownika o potwierdzenie ("*Czy jesteś pewien, że chcesz zakończyć program?*") i w razie odpowiedzi odmownej zrezygnuje z zamykania okna.  
[1p]
2. Napisać prosty edytor plików tekstowych. Program powinien korzystać z gotowych możliwości rozszerzonego pola tekstowego (*RichEdit*). Użytkownik powinien mieć możliwość odczytu i zapisu tekstu (wykorzystać funkcje *GetOpenFileName* i *GetSaveFileName*).  
[1p]
3. Napisać program, który będzie potrafił przekazywać komunikaty między wieloma instancjami siebie samego.  
Program powinien
  - użyć funkcji **RegisterWindowMessage** do zarejestrowania w systemie wiadomości wspólnej dla wszystkich instancji
  - po naciśnięciu przycisku powinien rozgłaszać jakąś tekstową wiadomość (**GetSystemTime**, **GetTimeFormat**) do wszystkich okien (**SendMessage** z parametrem `HWND_BROADCAST`)
  - po odebraniu nadanej wiadomości aplikacja powinna zareagować zmianą tytułu belki na (**Odebrano wiadomość: m**), gdzie zamiast **m** powinna pojawić się treść wiadomości.[2p]
4. Napisać program, który tworzy okno i w jego obszarze roboczym rysuje wykres funkcji  $f(x) = \sin(x)$  (z osiami).  
Wykres powinien automatycznie dopasowywać się do nowych rozmiarów podczas skalowania okna.  
[1p]
5. Napisać program, który odtworzy następujący wygląd okna z rysunku 1.  
Okno zawiera dwie ramki grupujące (*Group Box*). Pierwsza ramka zawiera dwa pola tekstowe (*Edit Box*), druga zawiera pole wyboru (*Combo Box*) oraz przycisk stanu (*Check Box*).  
Lista rozwijalna pola wyboru powinna być wypełniona nazwami kilku przykładowych miejscowości.  
Po wybraniu przez użytkownika przycisku **Akceptuj**, wybór powinien zostać zaprezentowany w oknie informacyjnym (rysunek 2).  
Naciśnięcie przycisku **Anuluj** powinno zakończyć program.  
[2p]

The image shows a dialog box titled "Katoteka osobowa" with a close button (X) in the top right corner. It is divided into two sections: "Dane osobowe" and "Adres".

**Dane osobowe**

Imie: [text input field]

Nazwisko: [text input field]

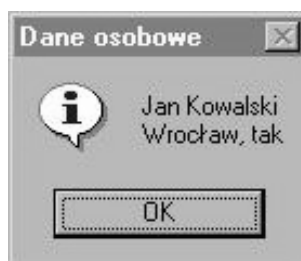
**Adres**

Miejscowosc: [text input field with a dropdown arrow]

Obywatel polski:

At the bottom of the dialog box are two buttons: "Akceptuj" on the left and "Anuluj" on the right.

Rysunek 1: Wygląd okna do zadania [5]



Rysunek 2: Informacja dla użytkownika do zadania [5]

6. Napisać program, który w oknie umieści panel grupujący (*Group Box*), a wewnątrz niego przycisk. Próba obsługi zdarzenia kliknięcia przycisku w funkcji obsługi komunikatów okna głównego nie uda się (*dla czego? gdzie trafia odpowiedni WM\_COMMAND?*).

Zastanowić się jak poradzić sobie w takiej sytuacji. Czy da się wymyślić rozwiązanie ogólne, tzn. radzące sobie przy dalszym zagnieżdżaniu paneli grupujących?

[1p]

7. Napisać program, który zademonstruje działanie trzech wybranych komponentów biblioteki Common Controls (ListView, TreeView, Animate Control, Progress Bar, Status Bar, Tool Bar, itd.). Demonstracja ma polegać na obsłudze kilku wybranych właściwości komponentów (na przykład wypełnieniu ListView kilkoma elementami, zmianie wartości i stylu Progress Bara itp.).

[2p]