

Ćwiczenie 4 - Pierwszy program w Delphi

1. Uruchom Windows, a następnie Delphi.
2. Przygotowanie aplikacji wyświetlającej napis "Witaj".

Przygotuj program wyświetlający okno zatytułowane "Okienko WITAJ" zawierające:

- przycisk "Napis" powodujący pojawianie się napisu "Witaj",
- przycisk "Wyczyść" powodujący usunięcie napisu,
- przycisk "Zamknij" powodujący zamknięcie okienka.

Okno wyświetlane przez program po naciśnięciu przycisku "Napis":



2.1. Przygotowanie formatki okienka

- Z menu wybierz opcję **File|New Application**
- Zlokalizuj położenie okienek:
 - Inspektora obiektów (**Object Inspector**),
 - Palety komponentów (**Component Palette**),
 - Okienka edytora kodu,
 - Formularza projektowego okienka aplikacji (**Form**) - przełączanie pomiędzy edytorem, a formatką klawiszem F12.
- Dołącz do okienka edytora plik projektu: - wybierz z menu opcję: **Project | View Source**
- Umieść w okienku **Form** komponent napisowy **Label** wykonując następujące czynności:
 - wybierz zakładkę **Standard** z palety komponentów,
 - kliknij ikonę **Label** (litera A)
 - kliknij miejsce na formatce gdzie chcesz umieścić napis
 - zaobserwuj jak zmienił się moduł programu zapisany w pliku *Unit1.Pas*

- Umieść w okienku trzy przyciski:
 - postępuj podobnie jak w przypadku etykiety,
 - wybieraj ikonę **Button** w palety **Standard**,
 - elementy rozmieszczone na formatce możesz przy pomocy myszy: przesuwać i zmieniać ich rozmiary,
 - wypróbuj operację usuwania komponentu: umieść dodatkowy przycisk na formatce, a następnie usuń go naciskając klawisz **Delete**. Sprawdź, że jednocześnie z usunięciem komponentu z formatki Delphi koryguje tekst modułu *Unit1*.

2.2. Ustalenie właściwości komponentów

Właściwości komponentów określ wykorzystując Inspektor obiektów.
Zlokalizuj poszczególne elementy Inspektora obiektów:

- **Selektor obiektów** - opuszczana w dół lista zawierająca spis wszystkich komponentów znajdująca się w górnej części Inspektora,
- Zakładki stron Inspektora:
 - strona **Properties** (Właściwości) - umożliwia ustawienie poszczególnych właściwości komponentu,
 - strona **Events** (Zdarzenia) - pozwala na zdefiniowanie obsługi zdarzeń dotyczących komponentu,
- Podział stron na kolumny:
 - lewa kolumna strony Properties zawierająca nazwy właściwości,
 - prawa kolumna strony Properties zawierająca wartości poszczególnych właściwości,
 - suwak pozwalający na przesuwanie listy właściwości.

Ustal właściwości komponentów zgodnie z tabelą:

Komponent	Właściwość	Wartość
Button1	Caption	&Napis
Button2	Caption	&Wyczyść
Button3	Caption	&Zamknij
Form1	Caption	Okienko WITAJ
Label1	Caption	<puste>
	Align	alClient
	Aligment	taCenter
	Color	clYellow
	+Font	????

Przełączanie obiektów w Inspektorze można dokonywać z użyciem selektora obiektów lub poprzez kliknięcie na odpowiedni komponent na formatce.

Wartości należy wpisywać lub wybierać z listy dostępnych wartości.

Określenie wartości dla właściwości **Font** wymaga wyboru odpowiednich cech czcionki (Times New Roman, Italic, rozmiar 36) w okienku dialogowym pojawiającym się po kliknięciu przycisku z trzema kropkami w polu wartości właściwości *Font*.

2.3. Określenie reakcji na zdarzenia.

Każde z trzech zdarzeń, które mogą się pojawić w czasie użytkowania naszej aplikacji (kliknięcie jednego z trzech przycisków) będzie obsługiwane oddzielną procedurą-metodą. Delphi umieści szkielet odpowiedniej metody w pliku *Unit1.Pas* po:

- dwukrotnym kliknięciu na zdarzeniu **OnClick** w prawej kolumnie strony **Events** w oknie **Object Inspektor** lub
- dwukrotnym kliknięciu komponentu (ten krótszy sposób dotyczy tylko zdarzenia pierwszego w kolumnie – w tym przypadku **OnClick**)

Należy wygenerować w opisany wyżej sposób trzy metody obsługi zdarzeń **OnClick** dla trzech przycisków. Należy wpisać następujące treści trzech metod:

```
przycisk "Napis":      Label1.Caption:='Witaj';
przycisk "Wyczyść":   Label1.Caption:=' ';
przycisk "Zamknij":   Form1.Close;
```

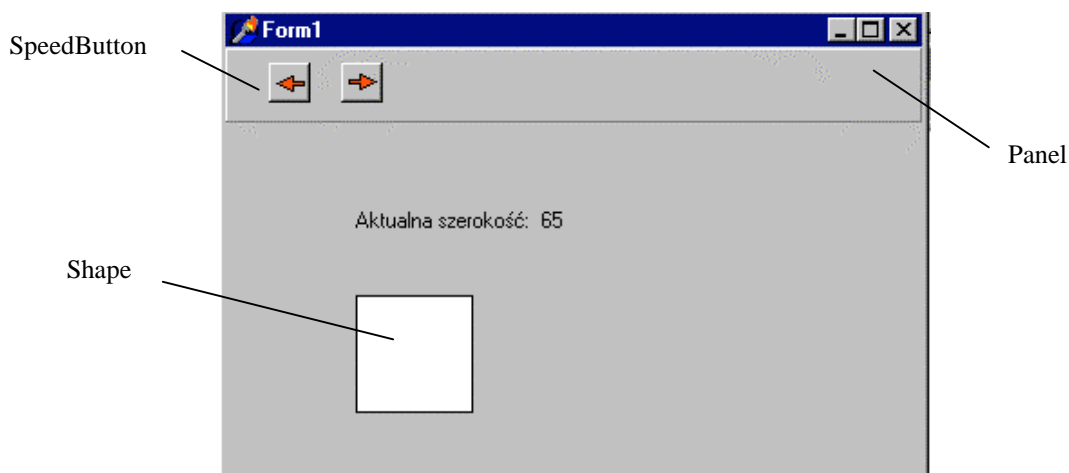
Uruchom aplikację.

Zapisz projekt w oddzielnym folderze o nazwie PIERWSZY.

3. Zrealizuj projekt aplikacji wyświetlającej prostokąt.

W okienku należy rozmieścić dwa przyciski służące do zwiększania i zmniejszania szerokości prostokąta.

Proponowana postać wyświetlanej formatki programu:



3.1. Rozmieść na formatce potrzebne komponenty:

- **Panel**: ustaw właściwości: **Align** = alTop, **Caption** = '' (puste)
- 2 komponenty **SpeedButton**; ustaw właściwość **Glyph** załadując potrzebne bitmapy ikon z katalogu: C:\Program Files\ Common Files\ Borland Shared\ Images\ Buttons,
- komponent **Label**,
- **Shape**; ustaw właściwość **Shape** = stRectangle,

Ustal pozostałe właściwości komponentów zgodnie z rysunkiem.

3.2. Zdefiniuj obsługę zdarzeń **OnClick** dla przycisków.

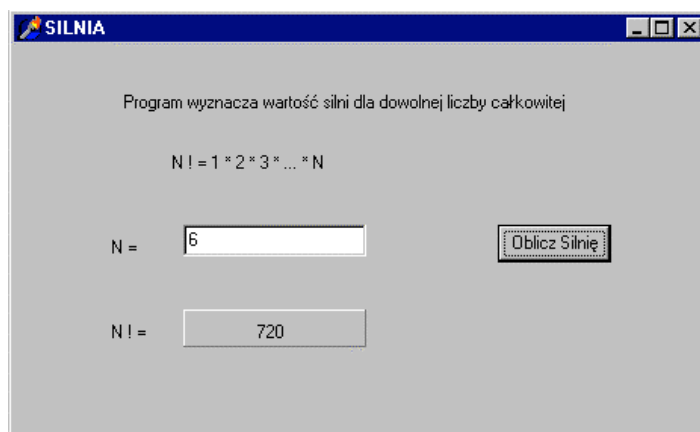
Proponowana procedura obsługi przycisku powodującego zwiększenie szerokości prostokąta:

```
procedure TForm1.SpeedButton1Click(Sender: TObject);  
begin  
Shape1.Width:=Shape1.Width+1;  
Label2.Caption:='Aktualna szerokość: ' + IntToStr(Shape1.Width);  
end;
```

3.3. Przetestuj działanie aplikacji i zapisz projekt w oddzielnym katalogu.

3.4. Rozbuduj aplikację tak by możliwa była także zmiana wysokości prostokąta.

4. Zrealizuj projekt aplikacji służącej do wyznaczania wartości silni.
Postać wyświetlanego okienka:



4.1. Rozmieść w oknie aplikacji: komponent **Panel**, 5 etykiet, przycisk **Button** oraz komponent edycyjny **Edit**. Ustal odpowiednio początkowe właściwości komponentów.

4.2. Zdefiniuj obsługę zdarzenia **OnClick** dla przycisku:

```
procedure TForm1.Button1Click(Sender: TObject);  
var i, N, Silnia: Integer;  
begin  
  try  
    N:=StrToInt(Edit1.Text);  
  
    {tu obliczenie wartości Silnia = N! }  
  
    Label5.Caption:=IntToStr(Silnia);  
  except  
    MessageDlg('Uwaga! Błąd', mtError, [mbOK], 0);  
    Edit1.SetFocus;  
  end;  
end;
```

UWAGA: Przydatne procedury do konwersji tekstów wpisywanych w polu edycyjnym *Edit* (własność *Text*):

StrToFloat - konwersja łańcucha na liczbę rzeczywistą,
FloatToStr - konwersja liczby rzeczywistej na łańcuch
StrToInt - konwersja łańcucha na liczbę całkowitą,
IntToStr - konwersja liczby całkowitej na łańcuch

Przydatne procedury wyświetlania komunikatów (poniżej nagłówki procedur):

```
function MessageDlg(const Msg: String; AType: TMsgDlgType;
  AButtons: TMsgDlgButtons; HelpCtx: LongInt): Word;
```

Msg - napis w okienku
AType - typ okienka może przyjmować wartości: - *mtWarning* (!), *mtError* (STOP),
mtInformation (i), *mtConfirmation* (?), *mtCustom* - bez bitmapy (tytuł - nazwa pliku programu),
AButton - zbiorowy - określa jakie przyciski będą występować w okienku.
Dopuszczalne wartości: *mbYes*, *mbNo*, *mbOK*, *mbCancel*, *mbHelp*,
mbAbort, *mbRetry*, *mbIgnore*, *mbAll*.
HelpCtx - określa który ekran Helpu jest dostępny w czasie wyświetlania okienka.

Funkcja zwraca wartość przyciśniętego klawisza: *mrNone*, *mrOK*, *mrCancel* itp.

```
function MessageDlgPos(const Msg: String; AType: TMsgDlgType; AButtons:
  TMsgDlgButtons; HelpCtx: LongInt; X,Y: Integer): Word;
```

jw. dodatkowo *X* i *Y* wyznacza miejsce okienka (poprzednie wyświetlane jest centralnie).

4.3. Przetestuj działanie aplikacji i zapisz ją w oddzielnym katalogu.

ZADANIA DOMOWE

1. Napisz program wyświetlający wizytówkę wg zamieszczonego poniżej wzorca:



Wykorzystaj komponenty etykiety oraz komponent **Panel** ze strony **Standard** i **Bevel** ze strony **Additional**. Dla komponentu **Bevel** dobierz odpowiednio właściwości **Shape** i **Style**.

Zmodyfikuj aplikację w taki sposób, by możliwe było przełączanie przy pomocy przycisku postaci wyświetlanej wizytówki z wymianą nr telefonu prywatnego na służbowy.

2. Zrealizuj aplikację, która po wprowadzeniu dwu liczb:
- x – wartość argumentu
 - eps – dokładność obliczeń
- wyznacza wartość $\exp(x)$ obliczoną z dokładnością *eps* z wykorzystaniem wzoru:

$$\exp(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

3. Zmodyfikuj aplikację z punktu 4 tak by wartość N wybierać z listy.
- Dokonaj w aplikacji kolejnych zmian: usuń komponent **Edit** i w jego miejsce wstaw **ListBox** z wprowadzonymi kilkoma wartościami N. Dokonaj odpowiedniej modyfikacji procedury obsługi zdarzenia **OnClick** (punkt 5.2.).
- Wskazówka:** Komponent **ListBox** posiada właściwość **Items**, która jest tablicą łańcuchów. Wypełnianie tablicy w fazie projektowania zapewnia edytor aktywowany w Inspektorze Obiektów (przycisk z trzema kropkami). Wybraną wartość z listy można otrzymać jako: `ListBox1.Items[ListBox1.ItemIndex]`