Kierunek: **ETI**
Przedmiot: P**rogramowanie w środowisku RAD - Delphi**
Rok **III** Semestr 5

Ćwiczenie 1
Proste aplikacje - podstawowe komponenty.

1. Informacje o programie i środowisku

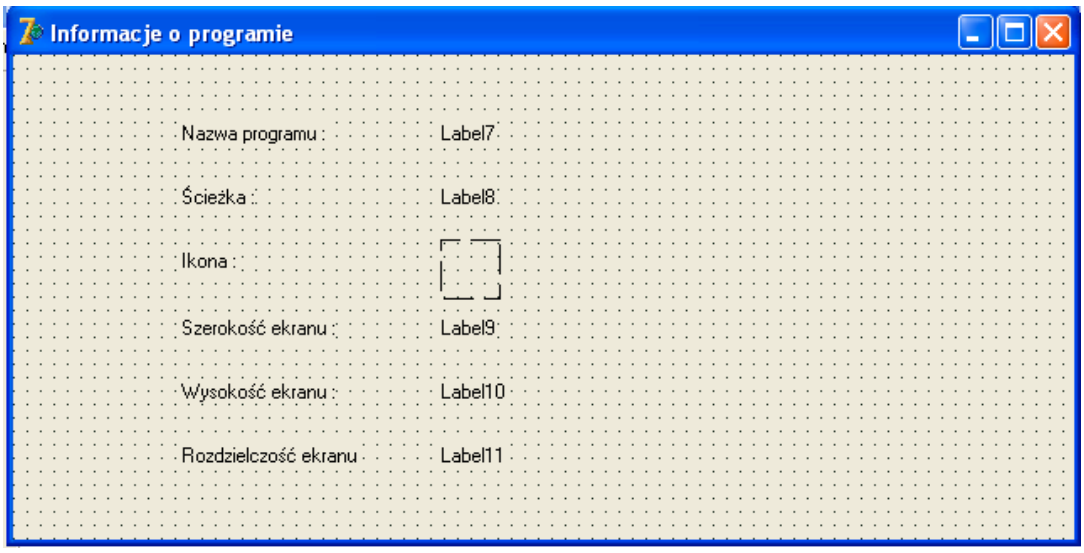
- Wygeneruj nową aplikację za pomocą opcji: **File | New | Application**
- Zlokalizuj położenie okienek:

  - Palety komponentów (**Component Palette**),
  - Okienka edytora kodu,
  - Formularza projektowego okienka aplikacji (**Form**) - wypróbuj przełączanie pomiędzy edytorem, a formatką klawiszem F12.
  - Inspektora obiektów (**Object Inspector**),

- 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 zawierającą nazwy właściwości,
    - prawa kolumna strony Properties zawierającą wartości poszczególnych właściwości,
    - suwak pozwalający na przesuwanie listy właściwości.

- Rozmieść w okienku projektowym formatki 11 komponentów napisowych **Label** oraz jeden komponent obrazka **Image** zgodnie z zamieszczonym niżej rysunkiem:



Rozmieszczenie komponentu **Label** wymaga następujących czynności:

- wybranie zakładki **Standard** z palety komponentów,
- kliknięcie ikony **Label** (litera A)
- kliknięcie miejsca na formatce gdzie należy umieścić napis
- dostosowanie pozycji napisu poprzez przesuwanie komponentu myszą lub poprzez bezpośrednią zmianę wartości właściwości **Left** i **Top**. Po wybraniu (kliknięciu) komponentu wartości te należy wpisywać w okienku Inspektora Obiektów w odpowiednim wierszu strony **Properties**

Zaobserwuj jak zmienia się moduł programu zapisany w pliku *Unit1.Pas* po każdorazowym dodaniu komponentu na formatkę.

Komponentu Image należy poszukiwać na zakładce **Additional** palety komponentów

- Ustal odpowiednie treści napisów etykiet i tytułu okienka. W tym celu dla formatki **Form** i dla sześciu etykiet **Label** wpisz odpowiednie teksty we właściwości **Caption** tych komponentów.

- W oknie edytora Delphi wyświetl zapis głównego pliku projektu o domyślnej nazwie Project1.dpr. Posłuż się opcją menu:

  Project | View Source

- Uzupełnij wygenerowany automatycznie zapis pliku zgodnie z tekstem zamieszczonym poniżej.

```
program Project1;

uses
  Forms,
  SysUtils,
  Unit1 in 'Unit1.pas' {Form1};

{$R *.RES}

begin
Application.Initialize;
Application.CreateForm(TForm1, Form1);
with Form1 do
   begin
   Label7.Caption := ExtractFileName(Application.ExeName);
   Label8.Caption := ExtractFilePath(Application.ExeName);
   Image1.Height := Application.Icon.Height;
   Image1.Width := Application.Icon.Width;
   Image1.Canvas.Draw(0, 0, Application.Icon);
   Label9.Caption := IntToStr(Screen.Height);
   Label10.Caption := IntToStr(Screen.Width);
   Label11.Caption := IntToStr(Screen.PixelsPerInch);
   end;
Application.Run;
end.
```

W zapisie wykorzystano:

SysUtils	moduł biblioteczny w którym zdefiniowane są użyteczne procedury m.in. ExtractFileName i IntToStr
Application	główny obiekt aplikacji wyposażony w szereg składowych przechowujących informacje o programie m.in. nazwę i ikonę aplikacji oraz składowych –metod jak: CreateForm – metoda kreująca formatkę, Run – metoda uruchamiająca główną pętlę aplikacji
Screen	predefiniowany obiekt pomocniczy aplikacji przechowujący m.in. parametry ekranu

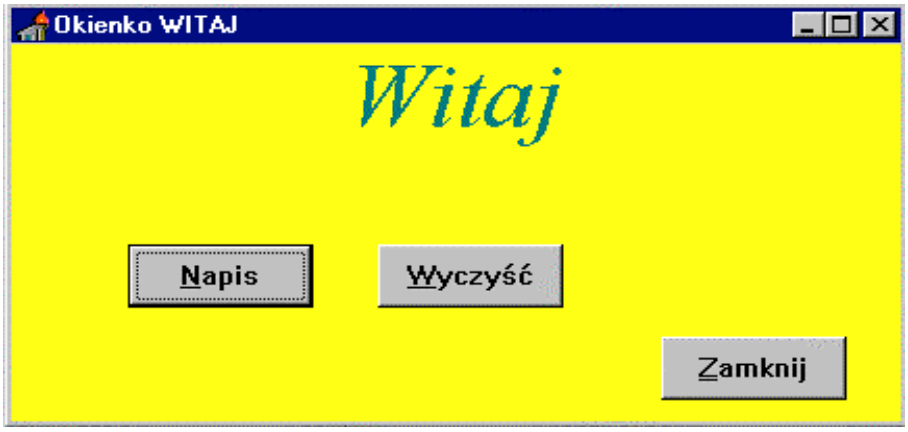
- Zapisz projekt w oddzielnym katalogu P1 pod nazwą Program1. Zaobserwuj ile plików zapisywanych jest w związku z każdym projektem.
  Uruchom aplikację.

2. Pierwsza aplikacja interaktywna

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":



- Wygeneruj nową aplikację. Rozmieść w okienku formatki trzy komponenty **Buton** i jeden komponent **Label**. Wszystkie te komponenty dostępne są na zakładce Standard palety komponentów.

- 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*.

- Określ reakcje aplikacji 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**)

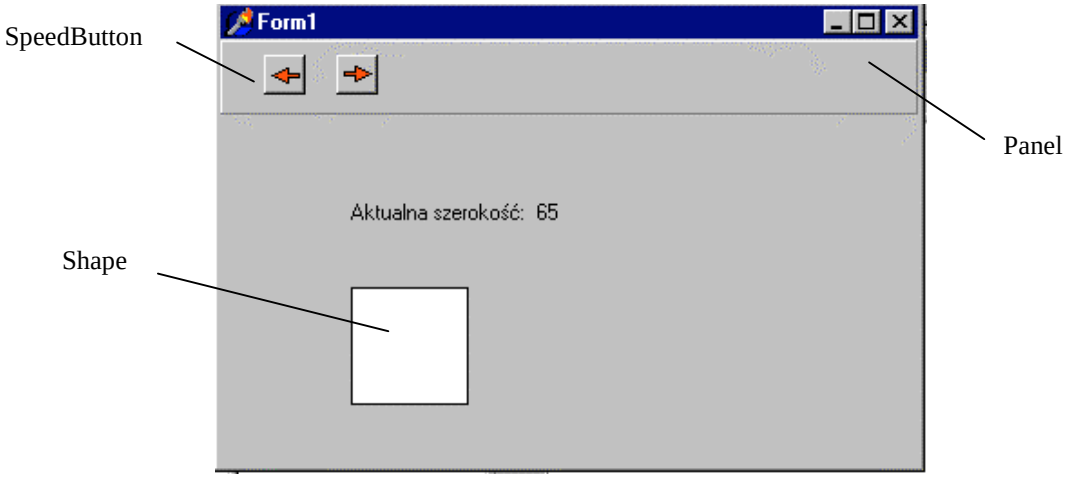
Należy wygenerować w opisany wyżej sposób trzy metody obsługi zdarzeń **OnClick** dla trzech przycisków. Należy wpisać następujące treści trzech metod:

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

- Zapisz projekt w oddzielnym folderze o nazwie P2. Uruchom i przetestuj aplikację.

3. Zmiana rozmiarów prostokąta

- Zrealizuj projekt aplikacji wyświetlającej prostokąt.
W okienku należy rozmieścić dwa przyciski służące do zwiększania i zmniejszania szerokości prostokąta.
Proponowana postać wyświetlanej formatki programu:



Rozmieść na formatce potrzebne komponenty:

- **Panel**: ustaw właściwości: **Align** = alTop,
  Caption = ‘’ (puste)
- 2 komponenty **SpeedButton**; ustaw właściwość **Glyph** załadowując potrzebne bitmapy ikon z odpowiedniego katalogu. Przy domyślnej konfiguracji jest to katalog: C:\Program Files\ Common Files\ Borland Shared\ Images\ Buttons,
- komponent **Label**,
- **Shape**; ustaw właściwość **Shape** = stRectangle,

Ustal pozostałe właściwości komponentów zgodnie z rysunkiem.

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

  Proponowana procedura obsługi przycisku powodującego zwiększenie szerokości prostokąta:

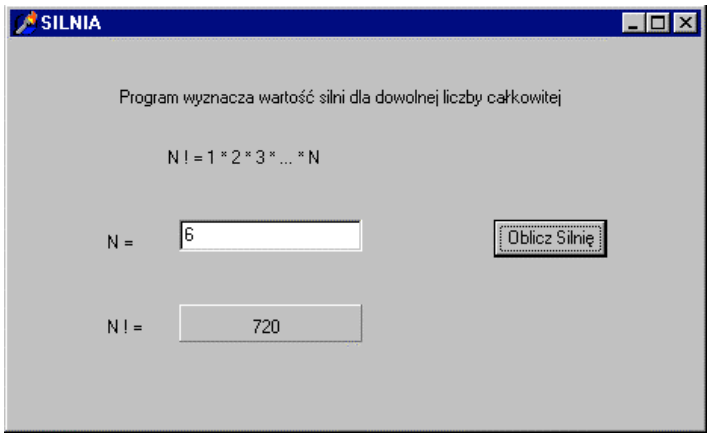
```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
Shape1.Width:=Shape1.Width+1;
Label2.Caption:='Aktualna szerokość: ' + IntToStr(Shape1.Width);
end;
```

- Zapisz projekt w oddzielnym katalogu i przetestuj działanie aplikacji .

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

4. Wyznaczanie wartości silni

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



- Rozmieść w oknie aplikacji: komponent **Panel** (dla wyświetlania na nim wyniku obliczeń), 5 etykiet, przycisk **Button** oraz komponent edycyjny **Edit**. Ustal odpowiednio początkowe właściwości komponentów.

- Zdefiniuj obsługę zdarzenia **OnClick** dla przycisku:

```
procedure TForm1.Button1Click(Sender: TObject);
var i, N, Silnia, kod: Integer;
begin
Val(Edit1.Text, N, kod);
if kod = 0 then
   begin

   {tu obliczenie wartości Silnia = N! }

   Label5.Caption:=IntToStr(Silnia);
   end
else MessageDlg('Uwaga! Błąd', mtError, [mbOK], 0);
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
Val(łańcuch, x, kod)	- zamiana *łańcucha* na liczbę *x*. Jeśli postać zapisu liczby w łańcuchu jest błędna, to *kod* (<>0) podaje numer znaku błędnego.

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).

- Zapisz projekt w oddzielnym katalogu i przetestuj działanie aplikacji.

5. Aplikacja prezentująca wizytówkę

- Opracuj aplikację wyświetlającą wizytówkę wg zamieszczonego poniżej wzorca:

Wykorzystaj komponenty etykiet 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.