



UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN  
The Faculty of Technical Sciences  
POLAND, 10-957 Olsztyn, M. Oczapowskiego 11  
tel.: (48)(89) 5-23-32-40, fax: (48)(89) 5-23-32-55  
URL: <http://www.uwm.edu.pl/edu/sobieski/> (in Polish)

---

# Podstawy programowania RAD

Środowisko Delphi

Wojciech Sobieski

---

Olsztyn 2004-2011

# Delphi

---

Programowanie obiektowe - (*object-oriented programming*) jest to metodologia tworzenia programów komputerowych definiująca je jako zbiór *obiektów* (pogrupowanych w *klasy*) oraz *metod*.

Klasa - określa strukturę obiektów, danych i operacji, jakie można wykonać na takich obiektach.

Obiekt - część programu komputerowego wykonująca określone zadanie.

Metoda - *procedura* lub *funkcja* będąca składnikiem klasy.

# Delphi

---

Funkcja - blok kodu wykonujący jakąś czynność i zwracający wynik pod swoją nazwą.

Procedura - blok kodu wykonujący jakąś czynność i nie zwracający wyniku.

Procedura obsługi zdarzenia - fragment kodu, który wywoływany jest w wyniku zajścia tego zdarzenia.

Zdarzenie - zachodzi w wyniku interakcji *komponentu* z użytkownikiem lub systemem.

Komponenty - są to części, z których budujemy program.

# Delphi

---

## Analogia - drukarka:

Obiekt: HP 959c  
Klasa: drukarka  
Metody: druk czarno-biały  
druk kolorowy  
wydruk próbny  
....  
Zdarzenia: wysłanie wydruku  
wstrzymanie wydruku  
...



# Delphi

---

## Analogia - młotek:

Obiekt: ???  
Klasa: ???  
Metody: ???  
???  
???  
???  
Zdarzenia: ???  
???  
???



# Delphi

---

Delphi - jest to zintegrowane środowisko programowania wizualnego firmy Borland ([www.bsc.com.pl](http://www.bsc.com.pl)), przeznaczone do programowania obiektowego w systemach Windows. Delphi oparte jest na języku ObjectPascal.

# Delphi

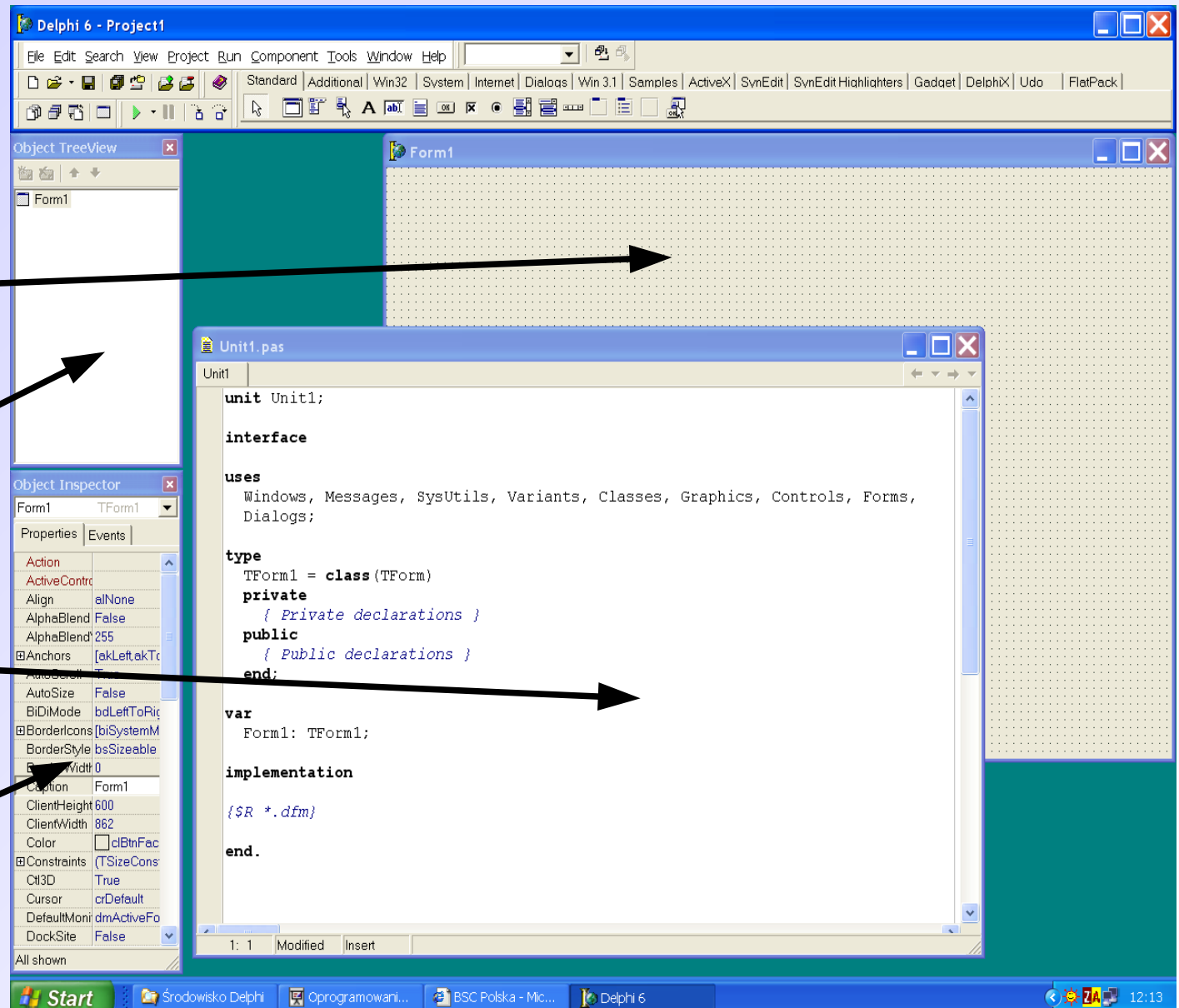
Główne okno  
Delphi

Okno  
formularza

Drzewo  
obiektów

Okno kodu

Inspektor  
obiektów



# Delphi

---

Tworzenie nowego projektu (programu):

1. Utworzenie katalogu projektu – należy to zrobić PRZED uruchomieniem Delphi. Najlepiej utworzyć katalog główny, np. “Projekty”, i w nim zapisywać swoje programy w oddzielnych katalogach. **Niedopuszczalne jest zapisywanie projektów bez przemyślenia struktury katalogów.**

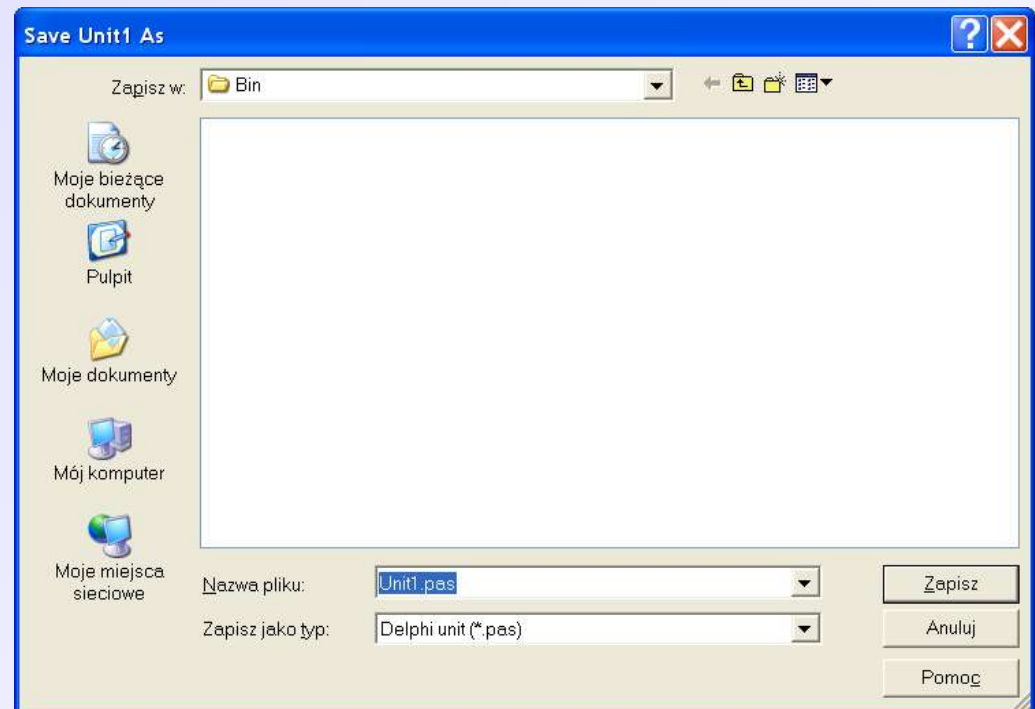


# Delphi

Tworzenie nowego projektu (programu):

2. Zapisanie nowego projektu (Plik → Save Project as...).

Pierwsze okno dialogowe odnosi się do głównego pliku \*.pas zawierającego kod źródłowy programu. Proponowana nazwa to “ModulGlowny.pas”.

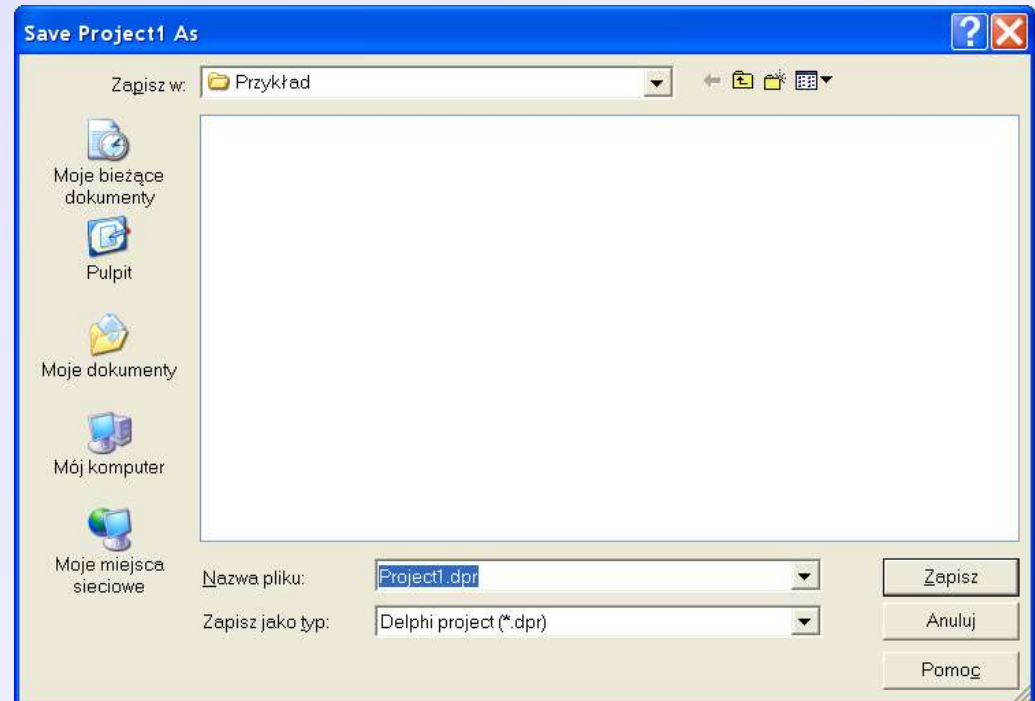


# Delphi

Tworzenie nowego projektu (programu):

2. Zapisanie nowego projektu (Plik → Save Project as...).

Drugie okno dialogowe odnosi się do nazwy całego projektu – tak będzie się również nazywał plik \*.exe po kompilacji kodu.



# Delphi

---

Pliki projektu:

- \*.~dfm - kopia pliku \*.dfm
- \*.~pas - kopia pliku \*.pas
- \*.dcu - plik skompilowany (bez możliwości uruchomienia)
- \*.dfm - plik formularza
- \*.pas - plik modułu
- \*.cfg - plik zawierający ustawienia kompilatora
- \*.dof - plik zawierający ustawienia projektu
- \*.dpr - plik projektu
- \*.exe - plik skompilowany (z możliwością uruchomienia)
- \*.res - plik zasobów

# Delphi – struktura modułu

**Moduł** – jest to plik tekstowy, który może być kompilowany do programu wykonywalnego. Nazwa modułu wypisywana jest w pierwszej linijce po słowie “unit”.



```
ModulGlowny.pas
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

end.
```

# Delphi – struktura modułu

**Interface** – sekcja zawierająca nazwy wszystkich elementów modułu (wewnętrznych i zewnętrznych).



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;


implementation

{$R *.dfm}

end.
```

# Delphi – struktura modułu

**Uses** – sekcja zawierająca nazwy modułów dodatkowych. Każdy nowy projekt ma wpisane podstawowe moduły, np. *Dialogs*, do wyświetlania okien komunikatów. Sekcję tę można rozszerzać poprzez dodanie nowych wpisów, np. *Math* dołączy moduł do zaawansowanych obliczeń matematycznych.



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

end.
```

# Delphi – struktura modułu

**Type** – sekcja zawierająca nazwy użytych w module obiektów oraz metod (czyli procedur i funkcji). Metody mogą odnosić się tylko do bieżącego modułu (obszar *private*) lub też mogą być “widoczne” w innych modułach tworzonego projektu (obszar *public*). W sekcji *type* można również definiować własne struktury.



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

end.
```

# Delphi – struktura modułu

**Var** – sekcja zawierająca nazwy użytych w module zmiennych.



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

end.
```



# Delphi – struktura modułu

**Implementation** – sekcja zawierająca kod programu: procedury obsługi zdarzeń, procedury i funkcje użytkownika.



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

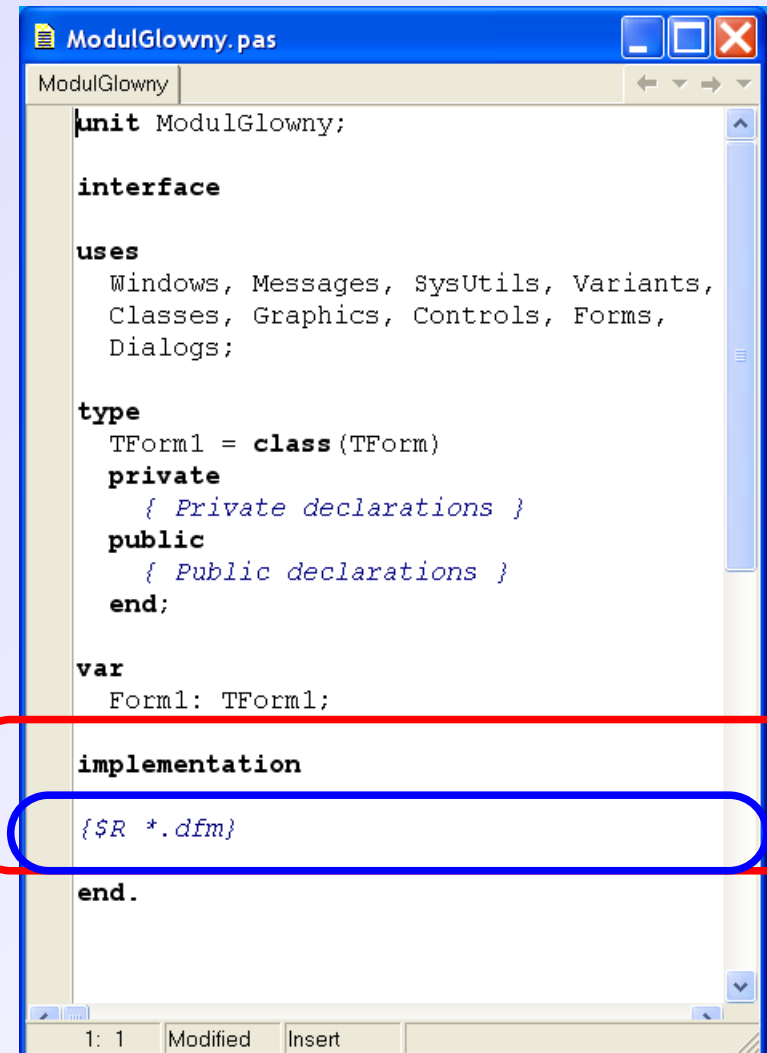
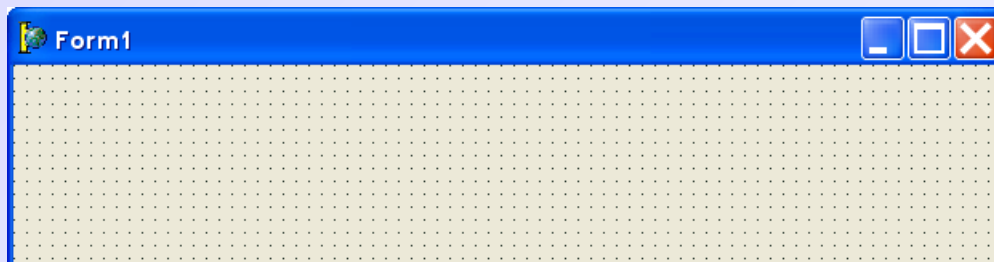
implementation

{$R *.dfm}

end.
```

# Delphi – struktura modułu

Programy tworzone w Delphi domyślnie posiadają Graficzny Interfejs Użytkownika (GUI). W sekcji *implementation* zawarta jest również informacja, w którym pliku należy szukać opisu związanego z danym modułem formularza.

A screenshot of a Delphi source code editor window titled "ModulGlowny.pas". The code is as follows:

```
unit ModulGlowny;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants,  
  Classes, Graphics, Controls, Forms,  
  Dialogs;  
  
type  
  TForm1 = class(TForm)  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form1: TForm1;  
  
implementation  
  
  {$R *.dfm}  
  
end.
```

The "implementation" section and the line "{\$R \*.dfm}" are highlighted with a red oval. The status bar at the bottom shows "1: 1 Modified Insert".

# Delphi – struktura modułu

W obszarze *implementation* można również dodawać sekcje *uses* lub *var* (tak jak w obszarze *interface*), ale zdefiniowane elementy będą miały zasięg lokalny i nie będą “widziane” w innych modułach projektu.



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

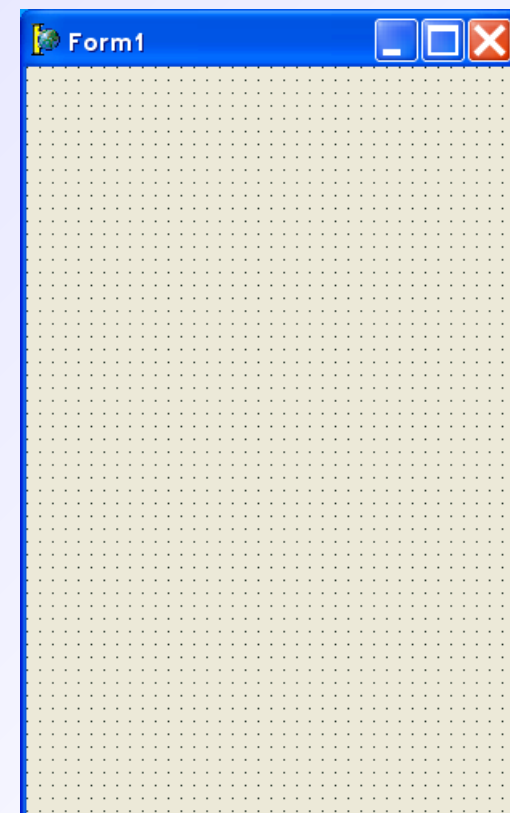
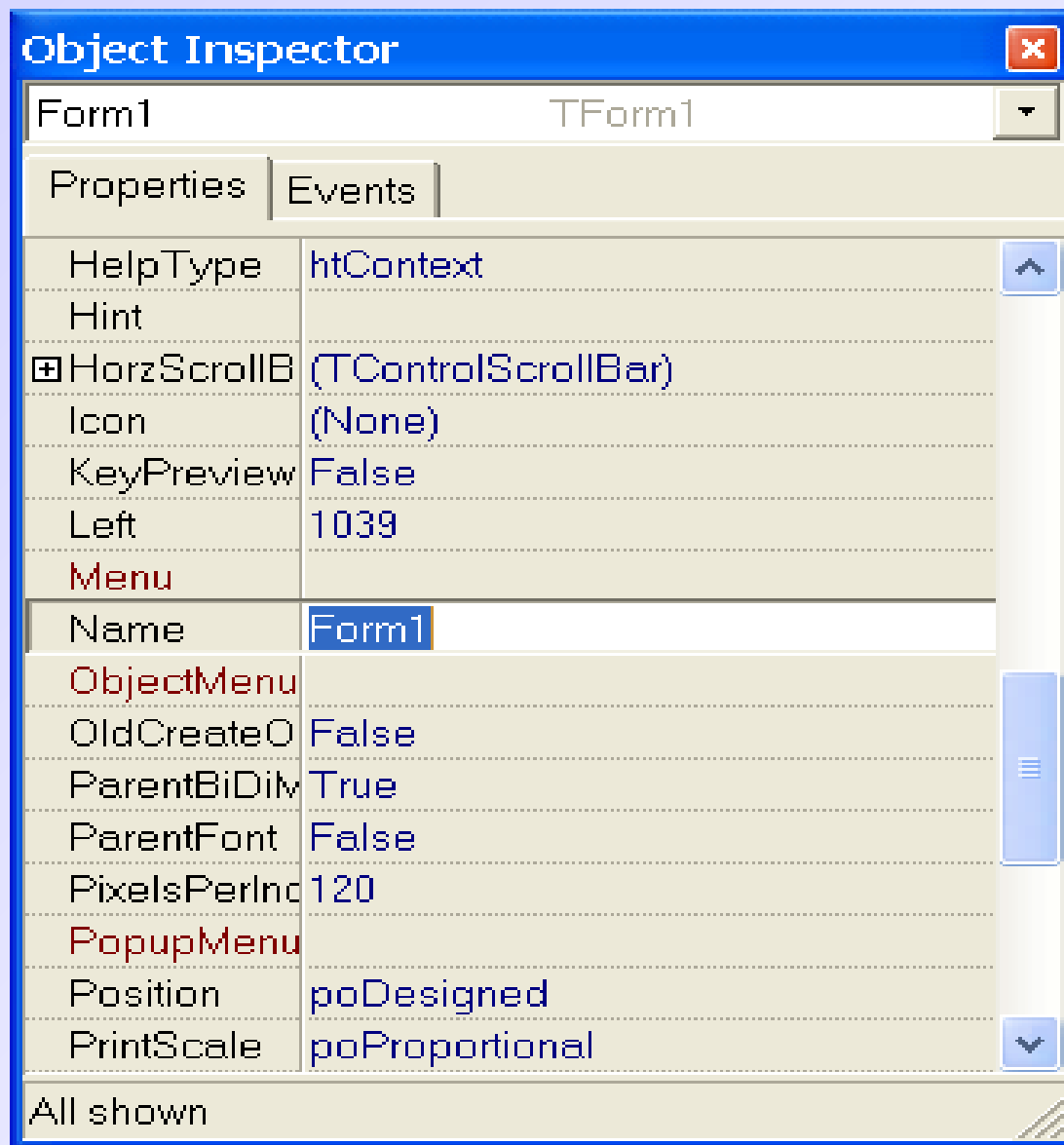
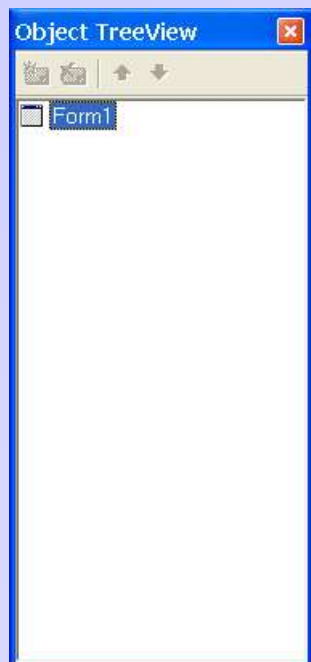
var
  Form1: TForm1;

implementation

{$R *.dfm}

end.
```

# Delphi – właściwości obiektów



# Delphi – właściwości obiektów

---

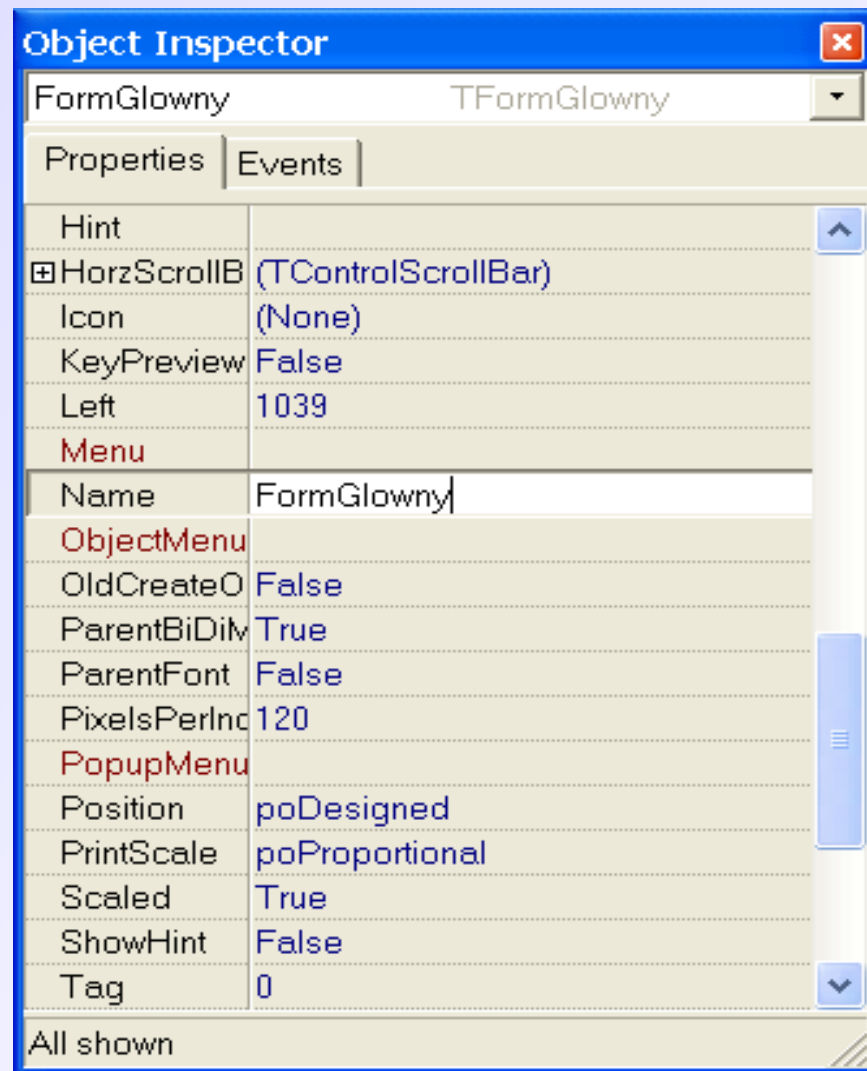
Wybrane właściwości obiektu TForm:

Name	– nazwa formularza
Caption	– napis widoczny na pasku formularza
Align	– opcje automatycznego położenia
BorderStyle	– typ okna
Enabled	– dostępność obiektu
Visible	– widoczność obiektu
Width	– szerokość formularza
Height	– wysokość formularza
Top	– odległość od górnego brzegu ekranu
Left	– odległość od lewego brzegu ekranu
Color	– kolor formularza

# Delphi – właściwości obiektów

Definiowanie właściwości:

1. poprzez ustawienie właściwości w Inspektorze Obiektów.

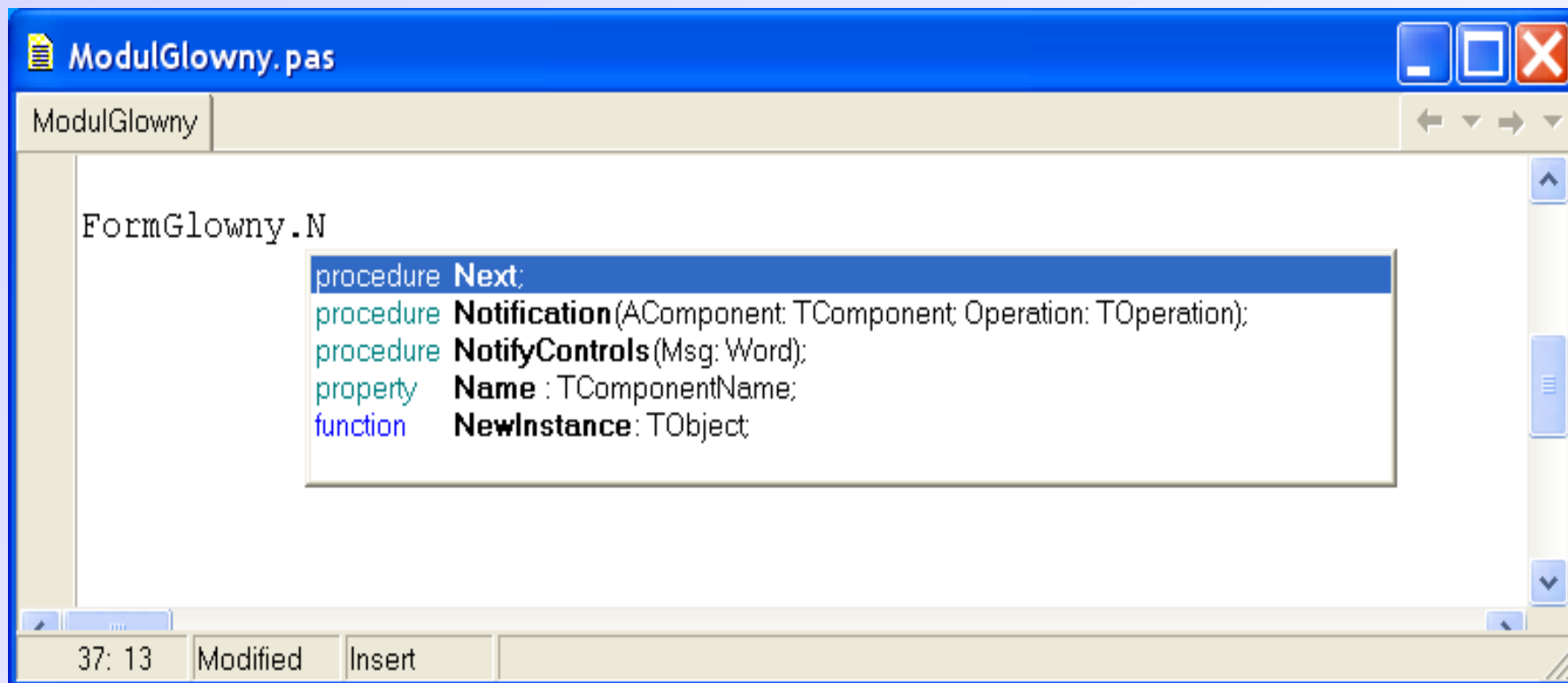


# Delphi – właściwości obiektów

---

Definiowanie właściwości:

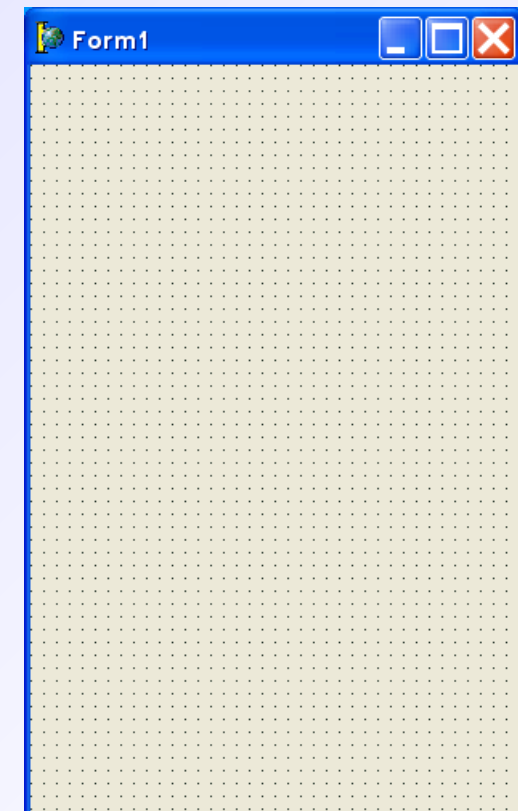
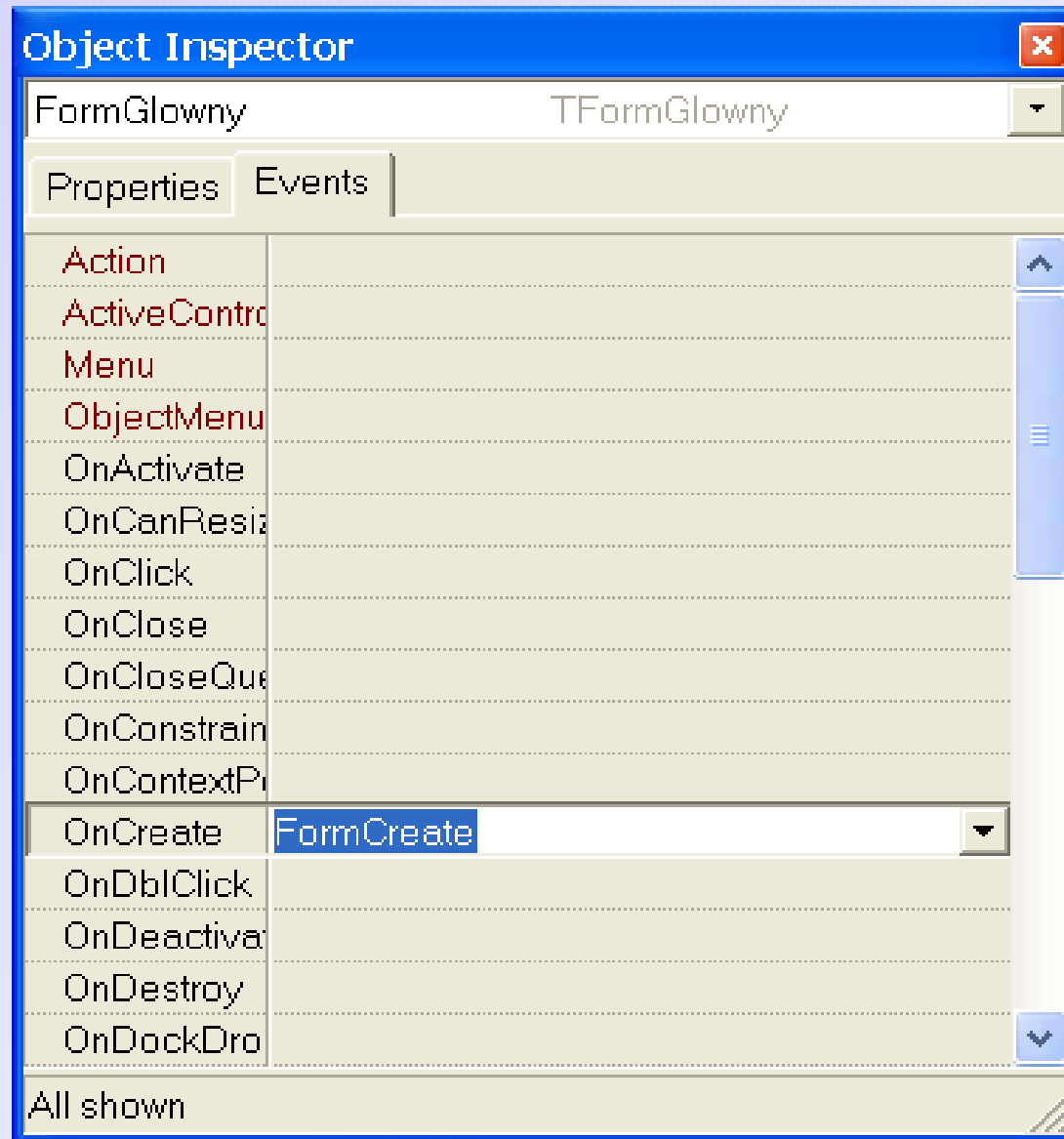
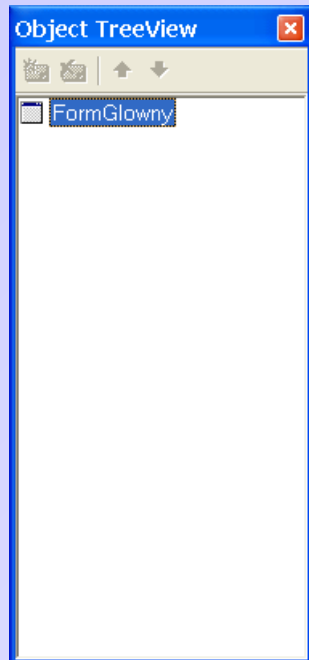
2. poprzez użycie odpowiedniego wpisu w kodzie źródłowym.



```
ModulGlowny.pas  
ModulGlowny  
FormGlowny.N  
procedure Next;  
procedure Notification(AComponent: TComponent; Operation: TOperation);  
procedure NotifyControls(Msg: Word);  
property Name : TComponentName;  
function NewInstance: TObject;
```

37: 13 Modified Insert


# Delphi – zdarzenia





# Delphi – zdarzenia

Definiowanie obsługi zdarzeń odbywa się w sekcji *implementation* – pojawia się tam “ramka” procedury.



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms, Dialogs
;

type
  TFormGlowny = class (TForm)
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormGlowny: TFormGlowny;

implementation

  {$R *.dfm}

  • procedure TFormGlowny.FormCreate(Sender: TObject);
  begin
  |
  end;

end.
```

28: 1 Modified Insert

# Delphi – zdarzenia

Wewnątrz procedury wpisuje się operacje jakie mają być wykonane po zajściu tego zdarzenia.



```
ModulGlowny.pas
ModulGlowny
unit ModulGlowny;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms, Dialogs
;

type
  TFormGlowny = class (TForm)
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormGlowny: TFormGlowny;

implementation

  {$R *.dfm}

procedure TFormGlowny.FormCreate(Sender: TObject);
begin
  • ShowMessage('Hello World');
  • end;

  • end.
```



UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN  
The Faculty of Technical Sciences  
POLAND, 10-957 Olsztyn, M. Oczapowskiego 11  
tel.: (48)(89) 5-23-32-40, fax: (48)(89) 5-23-32-55  
URL: <http://www.uwm.edu.pl/edu/sobieski/> (in Polish)

---

**Dziękuję za uwagę**

**Wojciech Sobieski**

---

Olsztyn 2004-2011