

Programowanie obiektowe

Programowanie w Delphi

Programowanie obiektowe

- Programowanie obiektowe (ang. **object-oriented programming**) to metodyka tworzenia programów komputerowych, która definiuje programy za pomocą obiektów .
 - Obiekty to elementy łączące *stan* (czyli **dane**) i *zachowanie* (czyli **procedury**, tu: **metody**).
 - Obiektowy program komputerowy wyrażony jest jako zbiór takich obiektów, komunikujących się pomiędzy sobą w celu wykonywania zadań.

Klasa i obiekt

- Obiekt to podstawowy element w programowaniu.
- Klasa to zbiór reguł opisujących obiekty.

Opis rzeczywistości

Obiekt	Azor
	
Klasa	Rasa Psa
	

Diagram klas - przykład



5

Obiekt

- Obiekt *Azor* należy do klasy **Bokser**
 - Klasa **Bokser** należy do klasy **Pies**
 - Klasa **Pies** należy do klasy **Psowate**
 - Klasa **Psowate** należy do klasy **SSAK**
- Obiekt *Azor* należy jednocześnie do klas **Bokser, Pies, Psowaty, Ssak**.
- Posiada wszystkie cechy klas nadrzędnych.
 - Ma jednocześnie cechy **boksera, psa, psowatego, ssaka**.

6

Cechy Klasy

- Każda klasa ma szczególne cechy.
- Klasa Bokser ma następujące:
 - Stan (*Właściwości*)
 - Wzrost
 - Masa
 - Wiek
 - Szybkość
 - Zachowanie (*Zdarzenia*)
 - Gryzienie
 - Jedzenie
 - Pilnowanie
 - Spanie

7

Zmiany cech obiektu Azor

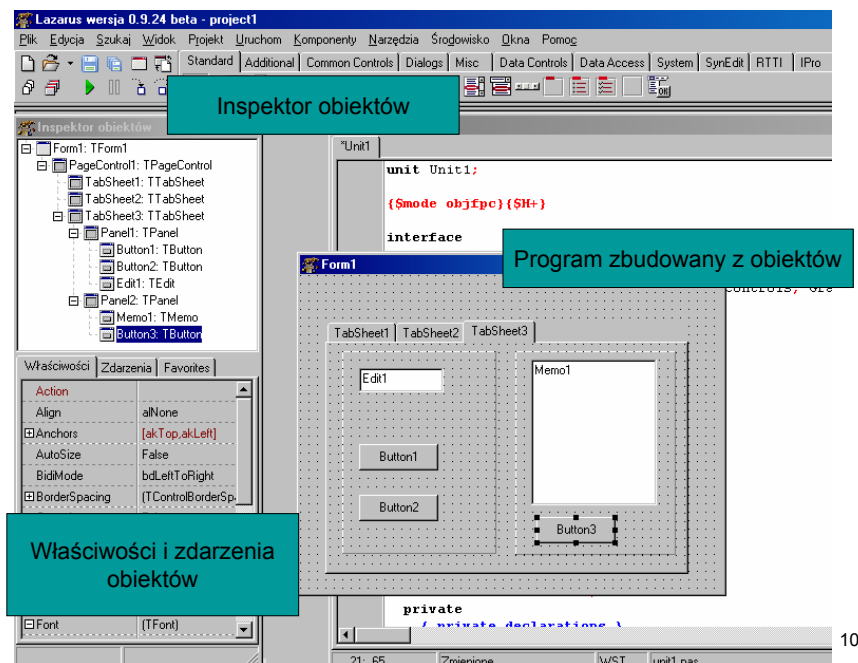
```
Azor:Bokser //Obiekt Azor należy do klasy Bokser
//Zmiana właściwości obiektu Azor
Azor.wzrost:=55cm;
Azor.masa:=33kg;
// Zmiana zdarzeń obiektu Azor
Azor.jeść;
Azor.gryzie:=true;
Azor.szczeka:=(inny_pies, głośno, przez_płot);
```

8

Programowanie obiektowe w Delphi

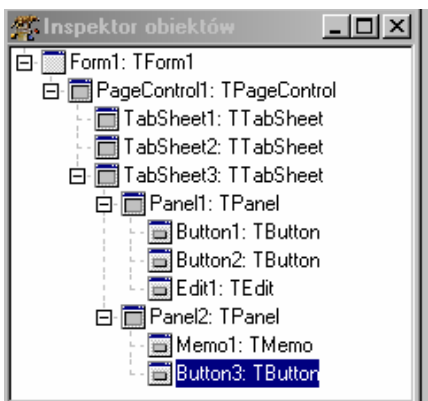
- Delphi jest językiem obiektowym
- Zawiera szerokie wsparcie dla obiektowości.
- Cechy obiektów są podzielone na:
 - właściwości
 - zdarzenia.

9



10

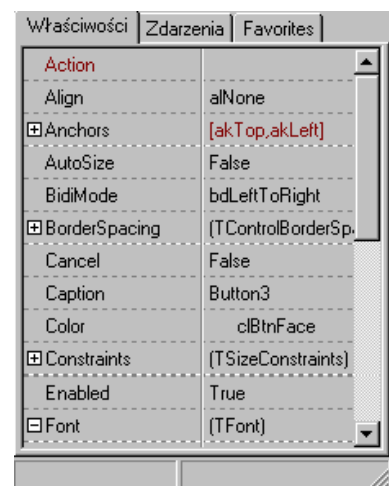
Inspektor Obiektów



- Ma strukturę drzewa.
- Pozwala to zaobserwować hierarchię obiektów w programie.
- Zawiera nazwę każdego obiektu i jego klasę źródłową.

11

Właściwości i zdarzenia obiektów



- Właściwości to cechy obiektu.
- Opisują jego wygląd.
 - Są to np. wielkość, kolor, rodzaj użytej czcionki.

12

Zdarzenia obiektów



- Zdarzenia to operacje wykonywane przez ten obiekt.
 - Można je zdefiniować w zależności od rodzaju zdarzenia np.
 - kliknięcie myszą na niego,
 - przeciągnięcie w inne miejsce,
 - naciśnięcie jakiegoś klawisza,
 - zmiana rozmiaru.

13

Definiowanie obiektów

- Przy wstawianiu komponentów do programu edytor automatycznie tworzy nowe obiekty, przypisując je do odpowiednich klas.

```
type
  TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Memo1: TMemo;
    PageControl1: TPageControl;
    Panel1: TPanel;
    TabSheet1: TTabSheet;
  end;
```

14

Obiekty w kodzie programu.

- Hierarchię w Delphi piszemy od najwyższego obiektu (klasy) do najniższego.
- Poszczególne poziomy oddzielamy kropkami.
 - Możemy napisać nazwę obiektu bieżącego
 - Button3
 - lub podać całą hierarchię
 - Form1.Button3

15

Cechy obiektu – edycja w kodzie

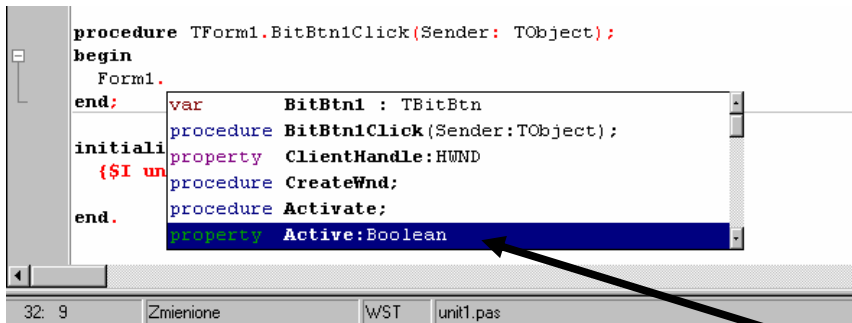
- Właściwości i zdarzenia obiektu również oddzielamy kropkami.

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  Form1.color:=ClRed;
  Button3.Font.Style:=FsUnderline;
  Panel1.Width:=300;
  Memo1.Text:='Domek Mamy';
  Form1.Close;
end;
```

16

Automatyczna podpowiedź

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Form1.
end;
var
  BitBtn1 : TBitBtn
initiali
  procedure BitBtn1Click(Sender:TObject);
  property ClientHandle:HWND
  {$I un
  procedure CreateWnd;
  procedure Activate;
end.
property Active:Boolean
```

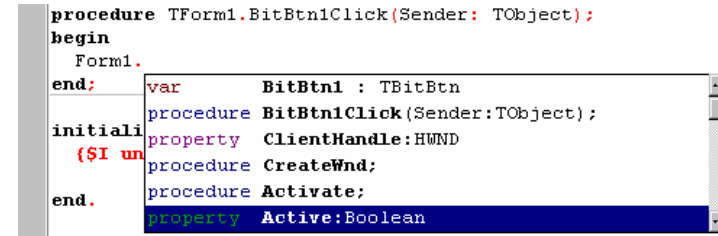


- Chcąc edytować cechy obiektu należy wpisać jego nazwę.
- Po wpisaniu kropki pojawi się podpowiedź zawierającą zbiór właściwości, funkcji i procedur, które może zrealizować ten obiekt.

17

Wstawienie odpowiedniej wartości

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Form1.
end;
var
  BitBtn1 : TBitBtn
initiali
  procedure BitBtn1Click(Sender:TObject);
  property ClientHandle:HWND
  {$I un
  procedure CreateWnd;
  procedure Activate;
end.
property Active:Boolean
```



- Podpowiedź pokazuje jednocześnie typ zmiennych, które może przybrać dana właściwość.
- W tym przykładzie są to wartości logiczne (*true* lub *false*).

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Form1.Visible:=false;
end;
```

18

19

20