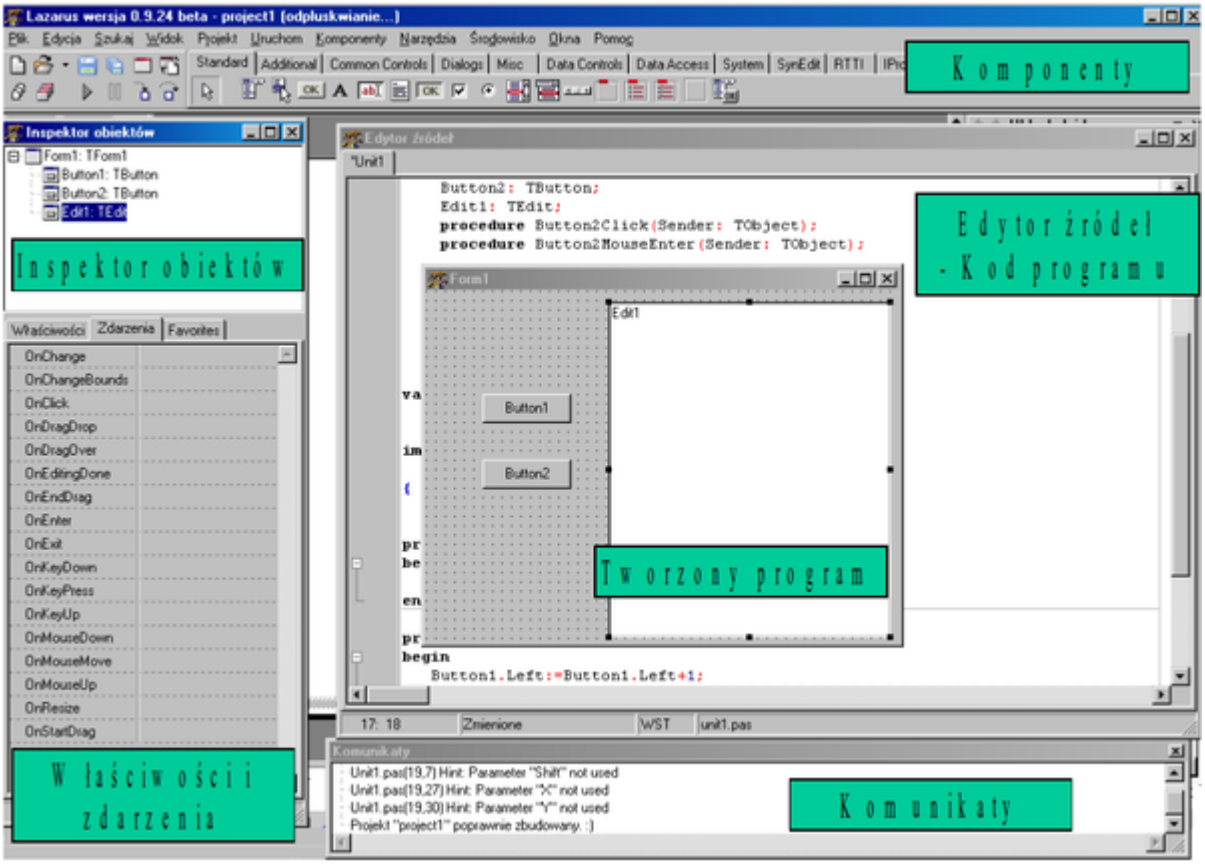
Nazwa implementacji: Kółko i krzyżyk w Lazarusie **Autor:** Piotr Fiorek Andrzej Stefaniuk **Opis implementacji:** Implementacja prezentuje środowisko Lazarus oraz prosty program w języku FreePascal.

Na początku zapoznamy się ze środowiskiem Lazarus.



Inspektor obiektów

- Inspektor obiektów pozwala zobaczyć wszystkie komponenty użyte w programie.
- Pokazuje jednocześnie, jak są one rozmieszczone.
- Dzięki niemu można się szybko przełączać pomiędzy nimi.
- Każdy z obiektów ma wyświetloną nazwę i klasę źródłową.

Właściwości

Z lewej strony ekranu znajduje się okno zawierające właściwości używanych obiektów.
Składa się z dwóch kolumn: lewa zawiera nazwy poszczególnych własności, prawa zawiera ich wartości.
Wiele wartości jest ustalanych domyślnie w trakcie inicjalizacji komponentu.
Wybór wartości może też być dokonywany ręcznie, lub wewnątrz kodu programu.

Zdarzenia

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

- Zdarzenia to podprogramy, które reagują na określone wydarzenia związane z danym komponentem.
- Zakładka ta składa się z dwóch kolumn: lewa zawiera nazwy poszczególnych zdarzeń, prawa zawiera procedury i funkcje przypisane do nich.

- Zdarzenie OnClick odpowiada sytuacji, gdy dany komponent zostanie kliknięty myszką. Przypisany podprogram wykona daną operację.
- Różnym zdarzeniom można przypisać ten sam podprogram

Favorites

- Trzecia zakładka zawiera ulubione właściwości i zdarzenia, które są często używane w odniesieniu do danego komponentu.

Komponenty

- Środowisko Lazarus zawiera zbiór podstawowych komponentów wykorzystywanych w programach.

- Są to np. przyciski, okienka, napisy, suwaki, pola wyboru, menu, tabelki itp.
- Pogrupowane są w kilku zakładkach, co pozwala je szybko wyszukać.
- Istnieje możliwość dodania nowych komponentów i stworzenia własnych.

Komunikaty
- Komunikaty to opisy, uwagi, ostrzeżenia i błędy wygenerowane przez kompilator Lazarus.
- Jest to podstawowe źródło informacji w razie problemów z uruchomieniem programu.

Tworzony program

- Okienko to przedstawia program, który jest tworzony.
- Jest to tryb WYSIWYG (What You See Is What You Get).
- Pozwala to na prostą i efektywną pracę.
- Zmianę właściwości i dodanie kodu realizujemy klikając na dany komponent.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Edytor źródeł - Kod programu

- To okno zawiera kod tworzonego programu.

- Pozwala na ręczną edycję kodu.
- Poszczególne części są wyróżnione innym kolorem lub tłustym drukiem.
- Domyślnie są uwzględnione wcięcia w kodzie.

Menu Edytora

- Klasyczne menu edytora.
- Pod nim zebrano kilka najczęściej używanych poleceń jak: zapisz, uruchom, nowy projekt.

Tworzenie programów w Lazarus

Program w języku Delphi składa się z kilku plików źródłowych:

*.pas – plik z kodem źródłowym programu

*.lpr – Plik Lazarus Project. Zawiera budowę całego programu.

*.lpi – Plik Lazarus Project Information. Zawiera dokładne dane o projekcie.

*.lfm — plik Lazarus Form. Zawiera opis formularza.

*.lrs — plik zasobów.

Z reguły nazwa projektu zaczyna się od słowa project natomiast pliku z kodem od słowa unit. Każdy program składa się z kilku plików, które mają podobne nazwy. Może to wprowadzić zamieszanie lub projekty mogą się skrzyżować. Należy więc utworzyć oddzielne katalogi dla każdego projektu. Powinny składać się z numeru projektu i imienia (lub nazwiska) twórcy: np. projekt1Andrzej.

Budowa programu

Program składa się z kilku wyróżnionych i wymaganych części:

- unit - Unit to część projektu zapisywana w oddzielnym pliku.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

- interface – Metody, z których korzysta Unit. Są to różne właściwości i operacje, jakie może wykonywać.
- uses - składniki używane przez dany Unit
- type – definicja występujących typów. Deklaracja komponentów, procedur i funkcji przez nie używanych, private - zmienne prywatne niedostępne z innych unitów, public - zmienne dostępne z innych unitów
- var - zmienne globalne występujące w Unicie
- implementation - implementacja metod– procedury i funkcje: procedure, function

initialization

– polecenie inicjalizacji zasobów edytora

•

```
unit Unit1;                 //początek programu – nazwa Unita
{$mode objfpc}{$H+}        //polecenie dla kompilatora
interface
uses                //lista zasobów używanych przez Unit. Część jest standardowa i dodawana automatycznie.
 Classes, SysUtils, LResources, Forms, Controls, Graphics, Dialogs, StdCtrls;
type
 { TForm1 }        //komentarz – definicja typu TForm1
 TForm1 = class(TForm)               //obiekt TForm1 należy do klasy TForm
   Button1: TButton;               //komponent Przycisk
         Button2: TButton;
   Edit1: TEdit;                //komponent Okienko edycyjne
         Edit2: TEdit;
   Label1: TLabel;                //komponent Etykieta
   procedure Button1Click(Sender: TObject);        //deklaracja procedura
   procedure Button2Click(Sender: TObject);

 private               //deklaracja zmiennych prywatnych– niedostępnych z innych Unitów
   { private declarations }
 public                        //deklaracja zmiennych publicznych – dostępnych z innych Unitów
   { public declarations }
 end;

var                              //zmienne
 Form1: TForm1;
 x,y,z:real;

Implementation               //Część implementacyjna Unitu
{ TForm1 }               //komentarz – implementacja metod TForm1
procedure TForm1.Button1Click(Sender: TObject);        //definicja procedury
begin
 x:=StrToFloat(Edit1.Text);
 y:=StrToFloat(Edit2.Text);
 z:=x+y;
 Edit3.Text:= FloatToStr(z);
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

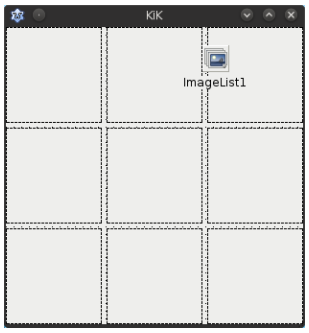
```
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
 x:=StrToFloat(Edit1.Text);
 y:=StrToFloat(Edit2.Text);
 z:=x-y;
 Edit3.Text:= FloatToStr(z);
end;

initialization        //polecenie dla kompilatora – Inicjalizacja zasobów edytora
 {$I unit1.lrs}
end.                 //koniec Unitu
```

Napisz prostą grę w Kółko i Krzyżyk

Okno Formularza:



Należy dodać obiekty TImage oraz TPictureList. W przypadku TPictureList należy określić jego rozmiary (powinien być taki sam jak rozmiar pól obrazkowych, które zamierzamy stworzyć, w przeciwnym wypadku obrazki będą przycięte) oraz po dwukrotnym kliknięciu w obiekt listę obrazków przedstawiających kółko oraz krzyżyk.

Obrazkom należy w zakładce „Events” przypisać funkcję, która będzie obsługiwać kliknięcie w obrazki. Aby stworzyć taką funkcję, wystarczy dwa razy kliknąć w jeden z obrazków, a w pozostałych wybrać funkcję z listy.W oknie „Object Inspector” można również ustawić nazwy poszczególnych elementów.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Kod:

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
 Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ExtCtrls;

type

 { TKiK }

 TKiK = class(TForm)
   // obiekty umieszczone an planszy
   Image1: TImage;
   Image2: TImage;
   Image3: TImage;
   Image4: TImage;
   Image5: TImage;
   Image6: TImage;
   Image7: TImage;
   Image8: TImage;
   Image9: TImage;
   ImageList1: TImageList;
   // procedury używane w programie
   procedure FormCreate(Sender: TObject);
   procedure Image1Click(Sender: TObject);
 private
   tura: Integer;  // zmienna zliczająca tury
 public
   { public declarations }
 end;

var
 KiK: TKiK;

implementation

{$R *.lfm}

// procedura wywoływana podczas startu programu
procedure TKiK.FormCreate(Sender: TObject);
begin
 tura := 0;  // wyzerowanie licznika tur
end;

// procedura wywoływana w momencie kliknięcia w kafelek
procedure TKiK.Image1Click(Sender: TObject);
var
 klikniety: TImage;  // zmienna do przechowywania informacji który kafelek został kliknięty
 wybieracz: Integer;  // zmienna przechowująca informację o tym jaki element umieścić
begin
 klikniety := (Sender as TImage);  // przypisanie kliknięgo elementu do zmiennego
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

```
 wybieracz := tura mod 2;  // określenie jaki obiekt teraz umieścić
 if klikniety.Tag = 0 then  // sprawdzenie czy kafelek nie był już wcześniej kliknięty
 begin
   ImageList1.GetBitmap(wybieracz, klikniety.Picture.Bitmap);  // przypisanie odpowiedniego obrazka do kafelka
   klikniety.Tag := 1;  // zapisanie informacji, że kafelek został kliknięty
   tura := tura + 1;  // zwiększenie licznika tur
 end;
end;

end.
```

UWAGA: Podczas zapisywania projektu nie wolno zapisywać plików ".pas" (plik z kodem) oraz ".lpi" (plik projektu) pod tymi samymi nazwami.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.