### Lazarus

#### Programowanie w Delphi



### Delphi

- Delphi jest to język oparty na Pascal.
- Twórcą jego jest firma Borland. Pierwotnie nosił nazwę Object Pascal (do 2006).
- Jest to język prosty i popularny.
- Wykorzystuje środowisko graficzne, co pozwala na napisanie prostych i efektownych programów.
- Posługuje się technologią RAD (*Rapid Application Development*), co ułatwia tworzenie interfejsu programu.

2

#### Linki do stron o Delphi

- <u>http://www.borland.pl/delphi/</u> strona producenta
- <u>http://www.swissdelphicenter.ch/en/</u> baza wiedzy na temat Delphi
- <u>http://www.torry.net/</u> olbrzymia baza komponentów oraz przykładowych kodów źródłowych
- <u>http://delphi.icm.edu.pl/</u> komponenty dla Kyliksa oraz wszystkich wersji Delphi i C++ Buildera
- http://www.delphi.org.pl Polska Grupa Użytkowników Delphi
- <u>http://delphi.about.com/</u> mnóstwo kodów źródłowych
- <u>http://www.unit1.pl/</u> Strona poświęcona programowaniu w Delphi
- <u>http://4programmers.net/Delphi</u> Strona poświęcona programowaniu w Delphi
- <u>http://4programmers.net/Delphi/Kompendium</u> książka "Delphi 7. Kompendium programisty"

3

# Kompilatory Delphi

Istnieją trzy kompilatory języka Delphi:

- 1. Borland Delphi
  - Edytor twórcy Delphi. Pierwsza wersja wyszła w 1995 do Windows 3.1. Obecnie istnieje 10 wersja tego środowiska.
- 2. Kylix
  - Kompilator do Linuksa. Wyszły trzy wersje i projekt zarzucono.
- 3. Lazarus
  - Darmowy projekt Open Source.

#### Lazarus

- *Lazarus* to zintegrowane środowisko programistyczne oparte o kompilator *Free Pascal*.
  - Jest to wzorowane na Delphi wizualne środowisko programistyczne oraz biblioteka Lazarus Component Library (LCL), która jest odpowiednikiem VCL.
- *Lazarus* jest zgodny z Delphi. Pozwala na rozwijanie programów, podobnie jak w Delphi, na wszystkich platformach obsługiwanych przez FPC.
  - Lazarus i Free Pascal starają się, aby raz napisana aplikacja kompilowała się wszędzie. Ponieważ dostępny jest dokładnie taki sam kompilator, nie trzeba wprowadzać żadnych zmian, aby otrzymać taki sam produkt dla różnych platform.
  - Program napisany w środowisku *Lazarus* można bez żadnych zmian skompilować dla dowolnego obsługiwanego procesora, systemu operacyjnego i interfejsu okienek.
- Program jest udostępniany na licencji GNU GPL, natomiast biblioteki na zmodyfikowanej licencji LGPL (umożliwia to wykorzystanie *Lazarusa* w projektach o zamkniętym kodzie).

#### Lazarus – cechy

- Lazarus posiada:
  - kompilator języka Pascal (FreePascal)
  - edytor kodu źródłowego
  - RAD (Rapid Application Development) szybkie tworzenie programu
  - wizualne tworzenie form (okien programu)
  - RTL zestaw gotowych komponentow
  - możliwość generowania kodu dla
    - kilku platform operacyjnych (win32, linux, mac)
    - i sprzętowych (pentium, PowerPC, Mac)

#### Lazarus – różne platformy

5

7

- Częścią Lazarusa jest Interfejs "*Widgets*", który umożliwia płynne przejście na inne platformy z zastosowaniem różnych interfejsów okienek.
- Aktualnie w różnych stadiach zaawansowania są interfejsy do następujących platform:
  - Windows
  - Unix
  - Mac OS X
  - Środowiska przenośne:
    - Windows CE
    - Qtopia for Linux-based PDAs
    - PalmOS (w przyszłości)
    - Symbian OS (w przyszłości)

#### Lazarus – materiały WWW

- <u>http://pl.wikipedia.org/wiki/Lazarus</u> Artykuł w Wikipedii
- <u>http://www.lazarus.freepascal.org/</u> strona projektu Lazarus
- <u>http://lazarus-ccr.sourceforge.net/</u> dokumentacja Lazarusa
- <u>http://wiki.lazarus.freepascal.org/index.php/Lazarus\_Do</u> cumentation - Wiki dokumentacja do Lazarusa
- <u>http://www.skinhat.com/lazarus/</u> Biblioteka GL do Lazarusa
- http://www.freepascal.org/ Free Pascal

#### Lazarus na Windows

G.SceneVevier 1: TGL5	keneVever 7	internet and an arrest		1.00	and the second	
and the second of a state of the second		inbrightnessfact	100.67 FPS			×
Panel 1: TPanel		ground\ripple.jp tors\knight\knig	nation stand	🗈 📧 🖬 🖓 Smooth	Load Weapon	
SBStop: TSpeedBu	K Lazarus Source Er	litor			Teachersteiner	EC
Glabel 1 Tabel	Unit1 Lazarus Actorip	r				
oties Events Favortes	procedure TF	ormi.FormMouseWhat				
tion	WheelDelta	: Integer: Mousel				
ctiveControl #Form1		Statement of the local division of the local				
gn utoficnal Animation						
utoSize						
rderIcons		D				
rderStyle		- no buscener	Left: 0 Top: 38			
aption		Wd	th: 526 Height: 380		Contraction of the second	
anti-picht		2	-			
					A REAL PROPERTY.	
entWidth				-0		
entWidth Jor				the second se	And the second se	
lentWidth olor onstraints				- Do -	<	191
ientWidth Nor snatraints jisor					STR.	5
entWidth Nor snatraints snor sidGite		THUR T				

#### Lazarus na Ubuntu



#### Lazarus na Mac OS X



11

#### Lazarus na Pocket PC



#### Widok środowiska Lazarus



#### Inspektor obiektów

Form1: TForm	1 Button Button it	
Właściwości Zda	rzenia Favorites	

• Inspektor obiektów pozwala zobaczyć wszystkie komponenty użyte w programie.

13

- Pokazuje jednocześnie jak są one rozmieszczone.
- Dzięki niemu można się szybko przełączać pomiędzy nimi.
- Każdy z obiektów na wyświetloną nazwę i klasę źródłową.

Właściwości	Zdarzenia	Favorites	
Action		_	
Align		lone	
	[ał	(Top,akLeft]	
AutoSelect	Fa	lse	
AutoSize	Fa	lse	
BidiMode		bdLeftToRight	
⊞ BorderSpac	ing (Tl	ControlBorderSpacing	
CharCase	ec	Normal	
Color		cWindow	
⊞ Constraints	(T	GizeConstraints)	
Cursor	crl	Default	
DragCursor	crl	Drag	
DragMode	dm	Manual	
EchoMode	em	Normal	
Enabled	Tn	Je	
🗆 Font	(T)	Font)	
CharSet	DE	FAULT_CHARSET	
Color		cWindowText	
Height	0		
Name	de	fault	
Pitch	fpl	Default	
Size	0		
⊟Style	0		
fsBold	Fa	lse	
2 10 10			

#### 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.

Właściwości Zdarze	nia	Favorites
OnChangeBounds		<u> </u>
OnClick	But	tton1Click
OnDragDrop		
OnDragOver		
OnEndDrag		
OnEnter		
OnExit		
OnKeyDown		
OnKeyPress	But	tton1KeyPress
OnKeyUp		
OnMouseDown		
OnMouseEnter		
OnMouseLeave		
OnMouseMove		
OnMouseUp	But	ton1MouseUp
OnResize		
OnStartDrag	But	ton1StartDra 💌 🛄
		<b>•</b>

#### Zdarzenia

- 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 dana operacje.
- Różnym zdarzeniom można przypisać ten sam podprogram.

17

Anchors	[akTop,akLeft]
Caption	Button1
ModalResult	mrNone
Name	Button1
OnClick	Button1Click 💌

[Set and all ⊐ in the Equation]

#### Favorites

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

#### 18

#### Komponenty

 Image: Standard Additional Common Controls Dialogs
 Misc
 Data Controls Data Access
 Syste

 Image: Standard Additional Common Controls Dialogs
 Misc
 Data Controls Data Access
 Syste

 Image: I

- Środowisko Delphi 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 WYOSIWYG (What You See Is What You Get).
- Pozwala to na prostą i efektywną pracę.
- Zmianę właściwości i dodanie kodu realizujemy klikając na dany komponent.

21

#### 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.

Program w języku Delphi

#### 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.

25

27

- \*.lfm -- plik Lazarus Form. Zawiera opis formularza.
- \*.lrs plik zasobów.
- Z reguły nazwa:
  - projektu zaczyna się od słowa project
  - pliku z kodem od słowa unit

#### Zapisywanie projektów

- 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.
  - Tworzymy je w C:\lazarus\projectsessions\
  - Można je też zapisywać na pendrive i nosić ze sobą.
- Powinny składać się z numeru projektu i imienia (lub nazwiska) twórcy.
  - Mają one mieć wzór typu: projekt2Jacek.

26



Z następnego okna wybieramy pierwsza pozycję – APLIKACJA.

#### Nowy projekt cz.2



#### Budowa programu

#### Budowa programu cz.1

Program składa się z kilku wyróżnionych i wymaganych cześci

- **unit** Unit to część projektu zapisywana w oddzielnym pliku.
- interface Metody z których korzysta Unit. Sa 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 - procedura
  - function funkcja

end.

• initialization – polecenie inicializacji zasobów edytora

30

#### Budowa programu cz.2

29

unit Unitl; {\$mode objfpc}{\$H+} interface	//początek programu – nazwa Unita //polecenie dla kompilatora	
uses //lista zasob	ów używanych przez Unit. Część jest standardowa i dodawana automaty	ycznie.
Classes, SysUtils, LResou	arces, Forms, Controls, Graphics, Dialogs, StdCtr	cls;
<pre>type { TForm1 } //komentarz TForm1 = class(TForm) Button1: TButton; Button2: TButton; Edit1: TEdit; Edit2: TEdit; Label1: TLabel; procedure Button1Clicl procedure Button2Clicl</pre>	<pre>- definicja typu TForm1     //obiekt TForm1 należy do klasy TForm     //komponent Przycisk     //komponent Okienko edycyjne     //komponent Etykieta k(Sender: TObject); //deklaracja procedura k(Sender: TObject);</pre>	
<pre>private   { private declaration; public   { public declarations end;</pre>	//deklaracja zmiennych prywatnych– niedostępnych z innych Unitów s } //deklaracja zmiennych publicznych – dostępnych z innych Unitów }	
<pre>var Forml: TForml; x,y,z:real;</pre>	//zmienne	31

#### Budowa programu cz.3

```
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);
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}
                //koniec Unitu
```

#### Zapisywanie, kompilacja i uruchamianie

#### Zapisanie programu

🌋 Lazarus wersja 0.9.24 beta	- project1	
<u>Plik E</u> dycja <u>S</u> zukaj <u>W</u> idok P <u>r</u>	2 ZapiszUnit1 (*.pas)	? X
Nowy moduł Nowy formularz	Zagisz w: 🔁 Familiada 💽 🔇 🎓 🖽 🕶	
Nowy  Dtwórz Ctrl+0  Verywróć Utwórz.poprzedni  Zapisz 2ctrl+S Zapisz jako Zapisz jako Zapisz Apisz Noth+Ctrl+S Zapisz wzystko Shift+Ctrl+S Zamknij Ctrl+F4 Zamknij wzystko	Moje biezące dokumenty Project_familiada Project_familiada dr Project_familiada dpr Project_familiada dpr Project_familiada dpr Wint_familiada ddp Unit_familiada df DUnit_familiada df DUnit_familiada df	
Czyść katalogi Drukuj Ctrl+P	dokumenty ST Mój komputer	
ter anchors a	Nazwa pliku:         Unit         Za           Moje miejsca sieciowe         Zapisz jako typ:         All File Types(".")         Ar	ipisz nuluj
BorderSpacing (TControlBor		

- Opcje zapisu:
  - Zapisz (CTRL+S)
  - Zapisz wszystko (CTRL+Shift+S)



#### Kompilacja

- Wchodzimy w Menu URUCHOM (*RUN*)
  - Wybieramy opcję BUDUJ (*BUILD*)
- Kompilację można wywołać klawiszami CTRL + F9.

#### Uruchomienie programu



34

- Wchodzimy w Menu URUCHOM (*RUN*)
   Wybieramy opcję URUCHOM (*RUN*)
- Lub naciskamy zielona strzałkę na pasku zadań.
- Uruchomienie można wywołać klawiszem F9.
- Chcąc zakończyć działanie należy wybrać pole WSTRZYMAJ lub nacisnąć znak pauzy.
   36

#### Wybór i wstawienie elementu Wstawianie komponentów i edycja 🕿 Lazarus wersia 0.9.24 beta - project1 Edycja Szukaj Widok Projekt Uruchom Komponenty Narzędzia Środowisko Okna Pomoc ich właściwości 🗅 🙆 • 📄 🖨 🖬 🖬 Standard Additional Common Controls Dialogs Misc Data Controls Data Access Sys 🖉 🕮 🕂 🖶 🚛 🖶 🗰 💭 🕵 a 6 🔤 ок 🖅 🛛 🔤 - C TBitBtn Ē Należy wybrać komponent na 🌋 Form1 - 🗆 × odpowiedniej zakładce i na niego kliknąć. • Następnie wstawić go w BitBtn1 odpowiednim miejscu na formatce. 37 38 Edycja kodu programu Edycja właściwości interf • Należy kliknać \*Unit1 uses • Należy kliknąć na \_ 🗆 🗵 na komponent, Właściwości Zdarzenia Favorites TForm1 = class (TForm) komponent, by móc BitBtn1: TBitBtn; Lance . procedure BitBtn1Click(Sender: TObject); by móc Caption BitBtn1 private edytować jego Color clBtnFace { private declarations } BitBtn1 edytować jego \_ 🗆 × nublic E Constraints (TSizeConstraints właściwości lub { public declarations Cursor crDefault end; zdarzenia lub False Default zdarzenia. var True Enabled BitBtn dodać polecenia Form1: TForm1 mhl ∃Font (TFont) implementation (TBitmap) Glyph wykonywane po end Height 30 { TForm1 } var kliknięciu na HelpContext n. Form: procedure TForm1.BitBtn1Click(Sender: TObject); HelpKeyword begin komponent. HelpType htContext implem STACJA11 end Hint initia Kind bkCustom initialization Layout blGlyphLeft {SI unit1.lrs} 22: 36 Left 40 end Margin -1 ModalResul mrNone D3D1.4 32: 3 Zmienione WST unit1.pas 39 40

# Edycja właściwości obiektów w kodzie programu

#### Automatyczna podpowiedź



- W trakcie pisania wystarczy wpisać nazwę obiektu i kropkę.
- Pojawi się wtedy podpowiedź zawierającą zbiór właściwości, funkcji i procedur, które może zrealizować ten obiekt.
- Wystarczy więc poszukać i wybrać odpowiedni.

42

#### Wstawienie odpowiedniej wartości



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

procedure TForm1.BitBtn1Click(Sender: TObject);

#### begin

```
Form1.Visible:=false;
```

end;

## Metody danego komponentu

- **var** komponenty znajdujące na aktualnym komponencie.
- property właściwość komponentu
- **procedure** procedura wykonywana przez ten komponent
- **function** funkcja wykonywana przez ten komponent