

# Lazarus vs Delphi



Zoznam autorov:

Michal Gašpierik, Peter Kovaľ, Peter Štitik, Tibor Balasz, Ján Škvarka

31. januára 2008

Táto publikácia vznikla ako študentský projekt v rámci predmetu GNU/Linux a jeho prostriedky na Fakulte matematiky fyziky informatiky Univerzity Komenského v Bratislave.

V práci sú použité názvy programových produktov, firiem a pod., ktoré môžu byť ochrannými známkami alebo registrovanými ochrannými známkami príslušných vlastníkov.

Sadzba programom pdfL<sup>A</sup>T<sub>E</sub>X

---

Copyright ©2007 Autori

Ktokoľvek má povolenie vyhotoviť alebo distribuovať doslovný alebo modifikovaný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom za predpokladu, že bude zachované oznámenie o copyrighte a autoroch. Pritom musí byť zachované aj toto a oznámenie, a to v nezmenenej podobe.

# Obsah

<b>1</b>	<b>Porovnanie prostredí</b>	<b>4</b>
1.1	Porovnanie menu . . . . .	4
1.2	Porovnanie panelu Object Inspector . . . . .	5
1.3	Porovnanie panelu s komponentami . . . . .	6
<b>2</b>	<b>Porovnanie odlišností v kóde</b>	<b>7</b>
2.1	Porovnanie pripojených knižní . . . . .	7
2.2	Odlišností funkcií . . . . .	8
2.3	Citlivosť na veľké a malé písmena . . . . .	10
<b>3</b>	<b>Odlišnosť uložených projektov</b>	<b>11</b>

# Úvod

V tomto dokumente sa budeme venovať porovnaniu dvoch dnes už známych programov a to Borland Delphi a Lazarus. Tieto programy sa dnes používajú na spracovanie zdrojových kódov v objektovo orientovanom pascale známom ako Delphi.

Ich spoločnou črtou je spoločný základ - chuť vytvárať nové veci.

Všetky rozdiely ktoré sú popísane v tomto dokumente boli nájdené pri prechádzaní zdrojových kódov v knihe „Informatika pre stredné školy: Programovanie v Delphi“ od autora RNDr. Andreja Blaha.

Programovací jazyk Delphi sa v dnešnej dobe používa na všetkých stredných a vysokých školách so záujmom o informatiku. Je to vynikajúci jazyk na získanie základných programátorských zručností a logiky.

Postupným bádáním sa dá zistiť, že tieto rozdieli sú skôr na úrovni aplikačnej ako funkčnej. Napríklad vizuálna stránka Lazarusa a Borland Delphi. Taktiež tieto dva programy by sa dali nazvať dvojčkami, kto si raz zvykne robiť v jednom bude vedieť robiť aj v druhom. A preto sú oba tieto programy vhodné na učenie základov programovania v jazyku Delphi za pomoci tejto knihy. Rozdieli v zdrojovom kóde sú ozať len minimálne a ich riešenie je zväčša intuitívne.

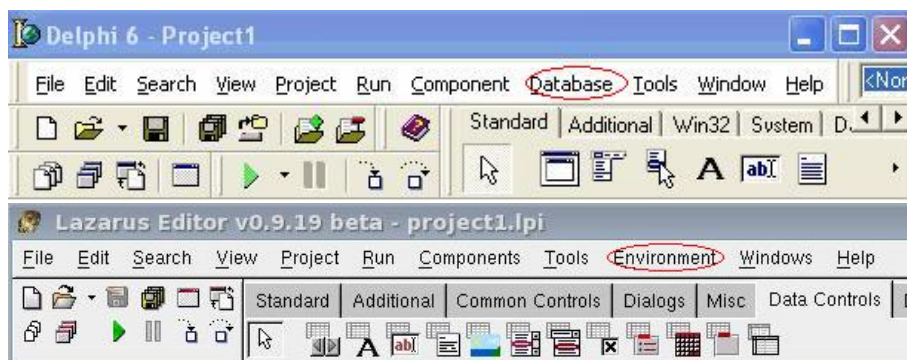
Prajeme vám veľa úspešných zdrojových kódov pri objavovaní programátorských zákutí.

Autori.

# Kapitola 1

## Porovnanie prostredí

### 1.1 Porovnanie menu



Na to aby sme mohli začať programovať v jednom z týchto programov musíme poznať ich prostredie. A najlepšie je začať od menu. Ako si môžete všimnúť oba programy majú veľmi podobné menu.

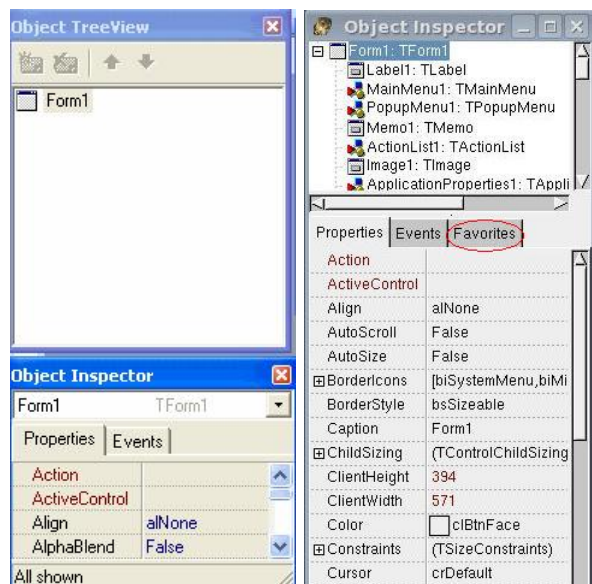
Ale každý program je predsa len trochu špecifický svojimi črtami. Napríklad Borland Delphi je už dlhé roky vyvíjaný a preto má iste prednosti pred mladším Lazarusom. Jeho hlavnou výhodou je priama podpora databáz („Databáze“) a príjemná pomoc.

Kdež to Lazarus vznikol ako požiadavka programátorov na príjemnejšiu správu vývoja. Preto je tu vyzdvihnutý popis prvkov a nastaviteľnosť („Environment“). Je pravda že tento prvok obsahuje aj Borland Delphi no je ukrytá v kontextovej ponuke nástroje („Tools“).

A Lazarus hoci nemá implementovanú pomoc ako Borland Delphi je prístupnejší pretože už na základnej stránke tohto programu Lazarus Project <http://www.lazarus.freepascal.org/> existuje fórum ohľadne pomoci užívateľom. Čo je mnohokrát užitočnejšie ako hodinové hľadanie v Pomoci podľa kľúčových slov. Samozrejme má to svoje tienisté stránky stále ak niečo hľadáte musíte byť

na internete. Ale v dnešnej dobe by to už nemala byť až taká prekážka.

## 1.2 Porovnanie panelu Object Inspector



Ďalším objektom ktorému sa budeme venovať je "Object Inspector". Je to jedna z najpoužívanějších častí ovládacích prvkov programu.

Načo vlastne slúži? Je to vlastne samotné nastavovanie vlastností pre jednotlivé komponenty nami vytvoreného programu.

Ako vidíme na obrázku Object Inspector sa delí na dve časti a to „Object Tree View” Stromová štruktúra komponentov a samotný „Object Inspector” ktorý popisuje vlastností samotného komponentu. V Borland Delphi sú tieto dva prvky ako samostatné okná no jednoduchým pretiahnutím myšou sa dajú spojiť do jedného okna tak to vidíme v Lazaruse.

Základným rozdielom medzi Object Inspectorom v Borland Delphi a Lazaruse je záložka „Favorites” kde si môžeme uložiť najčastejšie používané vlastnosti daného komponentu a tak ušetriť svoj čas stálym prehľadávaním celého Object Inspectoru. Pridávanie do záložky Favorites je veľmi jednoduché. Stačí kliknúť na vlastnosť ktorú chceme pridať to Favorites a pravým tlačidlom myši zobrazí kontextovú ponuku kde nájdeme položku pridať k Favorites. Takto si môžeme upraviť Favorites pre každý komponent zvlášť.

### 1.3 Porovnanie panelu s komponentami

Delphi:



Lazarus:



Posledným no nie zanedbateľným komponentom programu je panel s komponentami. Možno na prvý pohľad by sa zdalo že sú úplne rozdielne no niekedy zdanie klame.

Keďže Borland Delphi je program ktorý je určený výlučne pre operačné systémy od Microsoftu nájdeme na tomto paneli záložky ktoré sú určené výlučne len pre tento účel. Narozdiel od toho Lazarus je multiplatformový a záložky v tomto panely sa nastavujú podľa použitého operačného systému. Obrázok panela z Lazarusa ukazuje ako sú rozložené pod operačným systémom Linux. Preto pre tento účel je zbytočná záložka Win32. No ostatné zostáva ako aj v jeho konkurentovi.

Ako som sa už vyššie spomínalo Borland Delphi má podporu pre správu a prácu s databázami čo v Lazaruse zatiaľ chýba. Preto niektoré záložky sú prispôbené aj na tento účel.

Lenže záložky a komponenty možné požiť v nami vytvorenom programe sa dajú dopĺňať a tak si upravovať aj tento panel. Na internete je viacero modulov ktoré takéto komponenty dopĺňajú stačí si len vybrať ten správny pre daný program a nič nám nebráni rozšíriť ako Borland Delphi tak Lazarus. Len si treba dávať pozor pre ktorý program sú moduly určené a pre ktorú verziu daného programu.

## Kapitola 2

# Porovnanie odlišností v kóde

### 2.1 Porovnanie pripojených knižní

Najväčšie odlišnosti medzi týmito dvoma programami sa nachádza práve v tejto časti a to zdrojový kód. A zo zdrojových kódov je to práve časť o využívaní knižníc od tohto miesta sa budú odvíjať všetky zmeny zdrojového kódu.

Pripojenie knižníc v oboch programoch prebieha automaticky už pri vytvorení nového dokumentu („Projektu“). Borland Delphi má svoje knižnice ktoré načítava stále na začiatku pri spustení. Narozdiel od toho Lazarus pre svoju multiplatformovú výhodu kontroluje umiestnenie v operačnom systéme a tak i načítanie knižníc do programu.

Ukážeme si dva prípady Borland Delphi štandardné knižnice:

**uses**

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

Lazarus bežiaci pod Linuxom a jeho štandardne knižnice s týmto operačným systémom:

**uses**

```
Classes, SysUtils, LResources, Forms, Controls, Graphics
```

Z ukážok vidno, že Lazarus toho potrebuje menej na spracovanie programu a preto aj programy vytvárané pre Linux a písané v Lazaruse sú rýchlejšie a presné. Nezaberajú veľa miesta lebo sa do nich nekopíruje veľké množstvo zbytočných knižníc.



## 2.2 Odlišností funkcií

Ako bolo vyššie spomenuté tieto odlišnosti nie sú len tak sú spôsobené napájaním rôznych knižníc do zdrojového kódu. A takto vidíme ich výsledok. V Lazaruse ako základná grafická knižnica sa používa LCL kdež to Borland Delphi používa VCL. A na kóde sa to odzrkadlí v rozličnosti volaných funkcií.

Predstavme si náš program ktorý sa zaoberá kreslením na kresliace plátno TImage.Canvas. Máme na ňom zopár objektov dajme tomu nejaké štvorce a obdĺžniky. Chceme ich od seba farebne odlišiť. Je to veľmi jednoduché pri ich vytváraní im zvolíme rozličné farby z RGB škály. Ale pozor pre rozličnosť grafických knižníc sa v oboch programoch používajú rôzne príkazy na škálovanie farby z RGB hodnôt. V Borland Delphi by sa mal požiť príkaz v tvare: RGB(integer,integer,integer); na rozdiel od toho v Lazaruse vyzerá príkaz: RGBtoColor(integer,integer,integer); Z daného príkladu vidno, že Borland Delphi je viac užívateľsky orientovaný program. A Lazarus logicko programátorsky kde už z kódu vidno čo sa v ňom deje ( jednoduché vysvetlenie kódu: Chceme zmeniť RGB kód na farbu ).

Existuje aj mnoho ďalších rozličnosti zdrojového kódu. Ale pri výučbe programovacieho jazyka Delphi za pomoci knihy „Informatika pre stredné školy - Programovanie v Delphi“ v týchto dvoch prostrediach je ich zbytočne spomínať.

Pre lepšie pochopenie si uvedieme dva konkrétne príklady z prostredia Borland Delphi priamo z učebnice na strane 30:

```

procedure TForm1.Button1Click(Sender: Object);
begin
  Randomize;
  Image1.Canvas.Brush.Color := RGB(100+Random(156), 0, 0);
  Image1.Canvas.Ellipse(100-50, 150-50, 100+50, 150+50);
  Image1.Canvas.Brush.Color := RGB(0, 100+Random(156), 0);
  Image1.Canvas.Ellipse(200-50, 150-50, 200+50, 150+50);
  Image1.Canvas.Brush.Color := RGB(0, 0, 100+Random(156));
  Image1.Canvas.Ellipse(300-50, 150-50, 300+50, 150+50);
end;

```

A z prostredia Lazarus, už po úprave:

```

procedure TForm1.Button1Click(Sender: Object);
begin
  Randomize;
  Image1.Canvas.Brush.Color := RGBtoColor(100+Random(156), 0,
0); // v tomto riadku vidíme prvý krát úpravu RGBtoColor();
  Image1.Canvas.Ellipse(100-50, 150-50, 100+50, 150+50);
  Image1.Canvas.Brush.Color := RGBtoColor(0, 100+Random(156),

```

```
0);  
Image1.Canvas.Ellipse(200-50, 150-50, 200+50, 150+50);  
Image1.Canvas.Brush.Color := RGBtoColor(0, 0, 100+Random(156));  
Image1.Canvas.Ellipse(300-50, 150-50, 300+50, 150+50);  
end;
```

Týmto rozličnostiam sa venuje mnoho odborných aj laických web stránok za všetky spomeniem len jednu: Lazarus pre užívateľov Delphi [http://wiki.lazarus.freepascal.org/Lazarus\\_For\\_Delphi\\_Users/sk](http://wiki.lazarus.freepascal.org/Lazarus_For_Delphi_Users/sk).

## 2.3 Citlivosť na veľké a malé písmena

Táto rozličnosť nie je založená na použitých knižniciach ale na rozličnosti použitia operačného systému.

Keďže Borland Delphi je určená pre konkrétny operačný systém od Microsoftu ako tu už bolo spomenuté jeho rozlišovanie veľkých a malých písmen je rovnaká ako v tomto operačnom systéme. V skratke Borland Delphi nerozlišuje medzi malými a veľkými písmenami v zdrojovom kóde. Ako v názvoch funkcií tak v názvoch načítaných a uložených súborov.

Narozdiel od toho je Lazarus všestrannejší. Pri písaní premenných a názvoch funkcií rovnako nerozlišuje v malých a veľkých písmenách. Problém nastáva pri načítavaní a ukladaní súborov. Užívateľ si musí uvedomiť že každý operačný systém má svoje špecifické vlastnosti a tak vo Windows je rozlišovanie veľkých a malých písmen zanedbateľné. No v ostatných operačných systémoch založených na Unixe je to iné, tieto operačné systémy sú Case Sensitiv (rozlišujú veľké a malé písmena) a to isté treba použiť aj v zdrojovom kóde. Pri načítavaní treba dávať kde a aké písmeno použijeme či veľké alebo malé. Rovnako je to aj pri operačných systémoch od firmy Macintosh.

Nie je úplne jedno aký kód zadáte (príklad zo strany 116):

```
Memo1.Lines.LoadFromFile('readme.txt'); // otvára súbor s názvom readme.txt
```

Tento jednoduchý príkaz môže fungovať rôzne pod rôznymi operačnými systémami. Vezmime si prostredie Lazarusa a operačný systém Linux, kde po zadaní takéhoto kódu otvoríme súbor readme.txt. Hoci by sa v úložisku našli aj iné variácie tohto súboru napríklad: Readme.txt, rEadme.txt, v tomto prípade by boli ignorované. Narozdiel od toho Borland Delphi pod systémom Windows by takéto modifikácie ani nepovolil vytvoriť. Ale čo sa týka načítania fungovalo by to pre každý z týchto súborov nech už sa volá akokoľvek.

## Kapitola 3

# Odlišnosť uložených projektov

Pri uložení projektu z pomoci oboch programov vznikajú rôzne skriptovacie súbory kde sa určuje ako nami vytvorený program bude vyzerat' pri Borland Delphi to sú:

.cfg, .dpr, .dof, .res, .dfm, .pas

A pri Lazaruse:

.lrs, .lfm, .pas

Všetky tieto prípony majú svoj špecifický význam a záleží len na programe ako si rozdelí jednotlivé informácie o programe. Napríklad Borland Delphi v súbore s príponou .pas je uložený samotný zdrojový kód nami vytvoreného programu tú istú príponu používa aj Lazarus no v tomto súbore je uložených aj zopár konfiguračných informácií čo Borland Delphi ukladá do samostatného súboru s príponou .cfg.

Takýmto spôsobom by sme boli schopný rozobrať každú jednu príponu a s trochou šikovnosti i previesť jeden zdrojový kód na iný. A tejto problematike konverzii zdrojových kódov z Lazarusa do Borland Delphi sa venuje samostatná internetová stránka ktorá sa snaží poukázať na rozdielnosti v ukladaní zdrojových kódov týmito Programami. Konkrétne pre slovenských používateľov to je: [Converzia kódu http://wiki.lazarus.freepascal.org/Code\\_Conversion\\_Guide/sk](http://wiki.lazarus.freepascal.org/Code_Conversion_Guide/sk).

# Záver

V tomto dokumente sme sa venovali najzákladnejším rozdielom v programoch Borland Delphi a Lazarus. Ich zhodnosť je založená na použití rovnakého základu - chuť tvoriť nové veci.

Z uvedených rozdielov je zjavné, že oba programy sú vhodné na učenie základov programovania v jazyku Delphi za pomoci knihy „Informatika pre stredné školy - Programovanie v Delphi“ od RNDr Andreja Blaha. Je len na Vás ktorý z týchto programov si zvolíte.

Každý z týchto programov má svoje výhody i nevýhody na Vás zostáva voľba ktoré prvky sú pre Vás dôležitejšie.

Do budúca Vám prajeme veľa dobrých rozhodnutí.