

راهنمای سریع
زبان برنامه‌نویسی
Delphi

نویسنده:

محمد شمس

M.Shams.J@Gmail.Com

دلفی محصول شرکت Borland و یکی از پرکاربردترین زبانهای برنامه‌نویسی موجود در دنیای نرم‌افزار است که امروزه در زمینه‌های مختلفی از جمله برنامه‌های گرافیکی، چند رسانه‌ای، پایگاه داده و غیره به خوبی از آن استفاده می‌شود.

در حال حاضر به جرأت می‌توان گفت که یکی از پرکاربردترین زبان‌های برنامه‌نویسی تحت سیستم عامل ویندوز، دلفی است زیرا نه دارای پیچیدگیها و مشکلات زبان C بوده و نه محدودیتهای زبان Basic را دارد.

امروزه با توسعه برنامه‌نویسی ساخت یافته و استفاده از قابلیت‌های شیء‌گرایی (Object Oriented) قدرت این زبان به خوبی نمایان شده است، زیرا دلفی از قابلیت‌های بسیار مفیدی در رابطه با برنامه‌نویسی شیء‌گرا برخوردار است.

به طور کلی میتوان گفت که دلفی تلفیقی است از اشیاء موجود در زبان برنامه‌نویسی پاسکال استاندارد، که در حال حاضر به آن Object Pascal گفته می‌شود، و همچنین مجموعه‌ای از اجزای بصری به نام VCL.

VCL یا Visual Component Library خود سلسله مراتبی است از کلاسهای نوشته شده در شیء پاسکال و روشی است که در محیط مجتمع دلفی به کار برده شده و امکان طراحی سریع برنامه‌ها را فراهم می‌کند.

به عبارتی تمام کامپوننت‌های موجود در این زبان، جزئی از VCL هستند که به سادگی می‌توان آنها را از قسمت اجزاء (Component Palette) انتخاب کرده و روی فرم مورد نظر قرار داد. برای تنظیم ویژگیها و خواص هر یک از کامپوننتها می‌توان از پنجره Object Inspector که با کلید F11 نمایان می‌شود استفاده نمود.

تمام اشیای VCL نسلی از TObject هستند که یک کلاس عمومی بوده و روشهای اساسی مثل ساختن (Construction)، خراب کردن (Destruction) و مدیریت پیامها (Message Handling) را شامل می شود و اکثر کلاسهای دیگر از آن ارث بری می کنند. کامپونتهای VCL نیز نسلی از یک کلاس به نام TComponents هستند که جد تمامی کامپونتهای و اجزای ساخت می باشد.

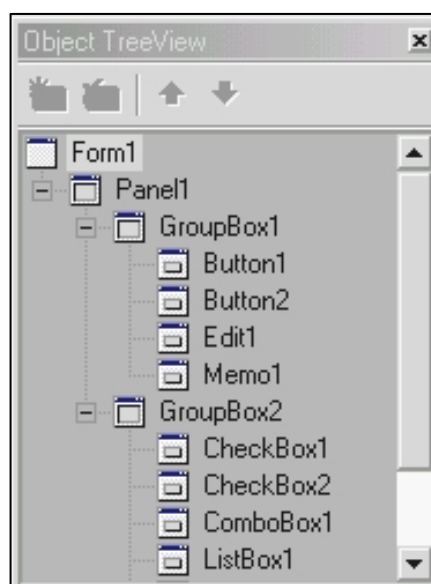
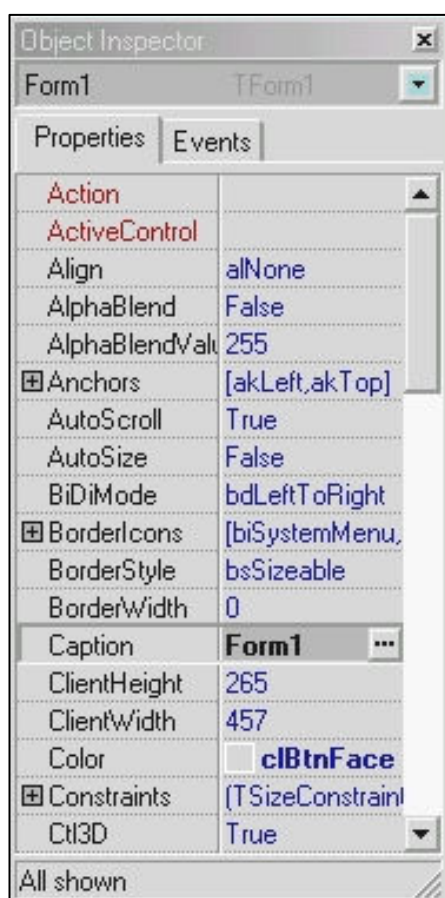
VCL بر خلاف نام آن که اجزای بصری را در ذهن شنونده تداعی می کند، بیشتر شامل اشیای غیر بصری است که به سادگی می توان آنها را به فرم مورد نظر اضافه کرده و از قابلیت های آنها استفاده نمود.

این گونه اشیاء در زمان طراحی (Design Time) به صورت یک شمایل (Icon) روی فرم ظاهر می شوند، اما در زمان اجرا نمایان نخواهند شد و به عبارتی Nonvisual هستند. به عنوان نمونه می توان از اجزای Ttimer و TdataSource نام برد که اولی جزء زمان سنج بوده و برای تکرار اجرای کدها در زمان مشخص استفاده شده و دومی نیز کامپوننتی برای اتصال به پایگاه داده می باشد.



(شکل ۲-۱ Component Palette)

برای تغییر خصوصیات اجزای غیربصری هم از Object Inspector استفاده می‌شود، که برای اینکار فقط کافی است که شیء مورد نظر را با کلیک کردن بر روی شمایل آن روی فرم و یا نام آن از Object TreeView انتخاب نموده و سپس خصوصیات آن را مشاهده کرده و یا تغییر دهیم.



(شکل ۲-۲ پنجره‌های Object Inspector و Object TreeView)

برنامه‌نویسی شیء‌گرا

برنامه‌نویسی شیء‌گرا در حقیقت همان برنامه‌نویسی ساخت یافته است که توسعه داده شده و با مجتمع کردن داده‌ها و روالها در واحدهای مستقل، قابلیت استفاده مجدد از آنها را توسط هر برنامه‌نویس دیگری فراهم می‌نماید، لذا کاهش مدت زمان طراحی و افزایش کارایی مهمترین مسائلی هستند که برنامه‌نویسی شیء‌گرا به ارمغان آورده است.

اگر مفهوم رکورد را در زبان پاسکل درک کرده باشید، به راحتی می‌توانید مفهوم شیء را نیز دریابید.

رکورد، متشکل از فیلدهایی است که هر کدام محتوای داده با نوع داده‌ای خاص خود هستند و دسترسی به تمام این فیلدها از طریق نام رکورد میسر است. اشیاء نیز مانند رکورد، مجموعه‌ای از عناصر داده‌ای هستند با این تفاوت که شیء می‌تواند شامل روالها و رویه‌های اجرایی نیز باشد، که اصطلاحاً به آنها متد (Method) گفته می‌شود.

به طور کلی کاری که در اشیاء انجام شده است، ترکیب داده‌ها و کارایی در یک واحد یکپارچه و مستقل است که به این عمل کپسوله‌سازی (Encapsulation) گفته می‌شود.

اجزای استاندارد دلفی

کنترل‌های موجود در دلفی بر اساس نوع کاربردشان گروه‌بندی شده و در دسته‌های مختلفی طبقه‌بندی شده‌اند که مهمترین آنها به شرح زیر است:

نام گروه	محتویات
Standard	منوها و کنترل‌های استاندارد سیستم عامل ویندوز
Additional	کنترل‌های پیشرفته
Win32	کنترل‌های مشترک در Win NT و Win 9X
System	اجزاء و کنترل‌هایی برای دسترسی به اجزای سیستم (مانند تایمر و چندرسانه‌ای)

(جدول ۱-۲ دسته‌بندی کنترل‌ها)

تمام خواص بصری که نسلی از TControl هستند، شامل خواص مشترکی به شرح زیر هستند:

نوع و کاربرد خواص	شامل
اندازه (Size) و موقعیت (Position)	Height, Width, Top, Left, AutoSize, Align
نمایش (Display)	BorderStyle, Color, Font, BevelStyle
راهنمایی (Navigation)	Caption, TabOrer, TabStop
خواص مربوط به Drag	DragMode, DragKind, DockMode

(جدول ۲-۲ خواص مشترک کنترل‌های VCL)

البته همه کنترلها با توجه به کاربردشان دارای ویژگیهای خاص خود نیز هستند که در جدول زیر به شرح مهمترین آنها می پردازیم:

خواص مشترک	نمونه	نوع کنترل
<p>Text: متنی که در آن نوشته شده. MaxLength: حداکثر طول متن. ReadOnly: مشخث می کند که متن قابل تغییر است یا خیر. Lines: متن به صورت آرایه ای از رشته ها.</p>	<p>Edit Memo Label MaskEdit RichEdit</p>	متنی
<p>Cancel: با زدن ESC اجرا شود. Default: با زدن Enter فعال شود.</p>	<p>Button SpeedButton BitBtn</p>	دکمه ها
<p>Checked: مقدار منطقی (Boolean) کنترل. State: وضعیت فعال یا غیر فعال بودن انتخابگر.</p>	<p>CheckBox RadioButton</p>	انتخابگرها
<p>Items: آرایه ای از تمام مقادیر موجود در آن. ItemIndex: اندیس گزینه انتخاب شده. ItemCount: تعداد اعضای لیست. Sorted: مرتب سازی محتوای لیست بر اساس حرف الفبا.</p>	<p>ListBox ComboBox TreeView ListView</p>	لیستها
<p>BackColor: رنگ زمینه. BevelStyle: برآمدگی و یا تورفتگی شکل. BorderStyle: تنظیمات حاشیه ها.</p>	<p>GroupBox Panel TabControl PageControl HeaderCtrl</p>	گروه بندی

خواص مشترک	نمونه	نوع کنترل
<p>Picture: تصویر موجود در آن.</p> <p>AutoSize: همسان کردن اندازه کنترل با تصویر موجود در آن.</p> <p>Transparent: نامرئی شدن زمینه کنترل.</p> <p>Center: قرارگیری تصویر در مرکز کنترل.</p> <p>Stretch: همسان کردن اندازه تصویر با اندازه کنترل.</p> <p>Canvas: محتوای کنترل به صورت یک شیء گرافیکی.</p>	<p>Image</p> <p>PaintBox</p> <p>Shape</p> <p>Animate</p> <p>Bevel</p>	تصاویر
<p>Filename: لیستی از فایل‌های انتخاب شده.</p> <p>NamePath: نام فایل Highlight شده در لیست.</p> <p>Filter: گزینش در لیست فایل‌هایی که نمایش داده می‌شود.</p> <p>Title: عنوان کادر پنجره محاوره‌ای.</p>	<p>SaveDialog</p> <p>OpenDialog</p>	محاوره‌ای
<p>Position: مقدار عددی فعلی کنترل.</p> <p>Min: حداقل مقدار عددی.</p> <p>Max: حداکثر مقدار عددی.</p>	<p>ScrollBar</p> <p>TrackBar</p> <p>UpDown</p>	کنترلی
<p>Interval: وقفه یا بازه زمانی مابین دفعات اجرا.</p> <p>Enabled: فعال یا غیر فعال شدن.</p>	<p>Timer</p>	سیستمی

(جدول ۲-۳ مهمترین کنترل‌ها)

ایجاد فرمها و شروع برنامه‌نویسی

برای ایجاد یک پروژه جدید می‌توان از منوی File و گزینه New Project استفاده نمود که پس از این کار، دلفی یک فرم جدید را روی صفحه ظاهر کرده و در قسمت ویرایشگر کد (Code Editor) نیز یک Class برای آن به وجود می‌آورد. کد ذکر شده مشابه زیر خواهد بود:

Unit یونیت ;

interface

نام یونیت‌های استفاده شده

uses

Windows, Messages, SysUtils, Variants, Classes ... ;

تعریف انواع داده‌ای، کلاسها، اشیاء و غیره

Type

TForm1 = class(TForm)

private

{ Private declarations }

public

{ Public declarations }

end;

تعریف متغیرها

Var

Form1: TForm1;

شروع قسمت پیاده‌سازی

Implementation

{ \$R *.dfm }

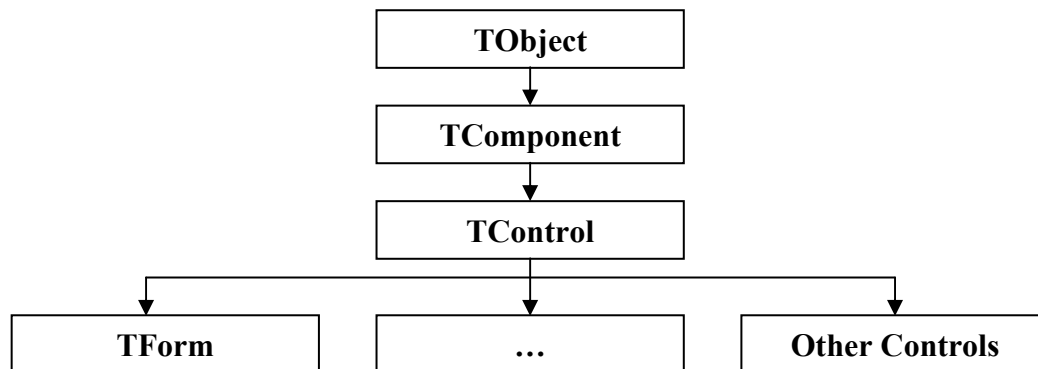
end.

Tform1 نوع کلاس جدید می‌باشد که از نوع TForm منشعب شده و متغیری به نام Form1

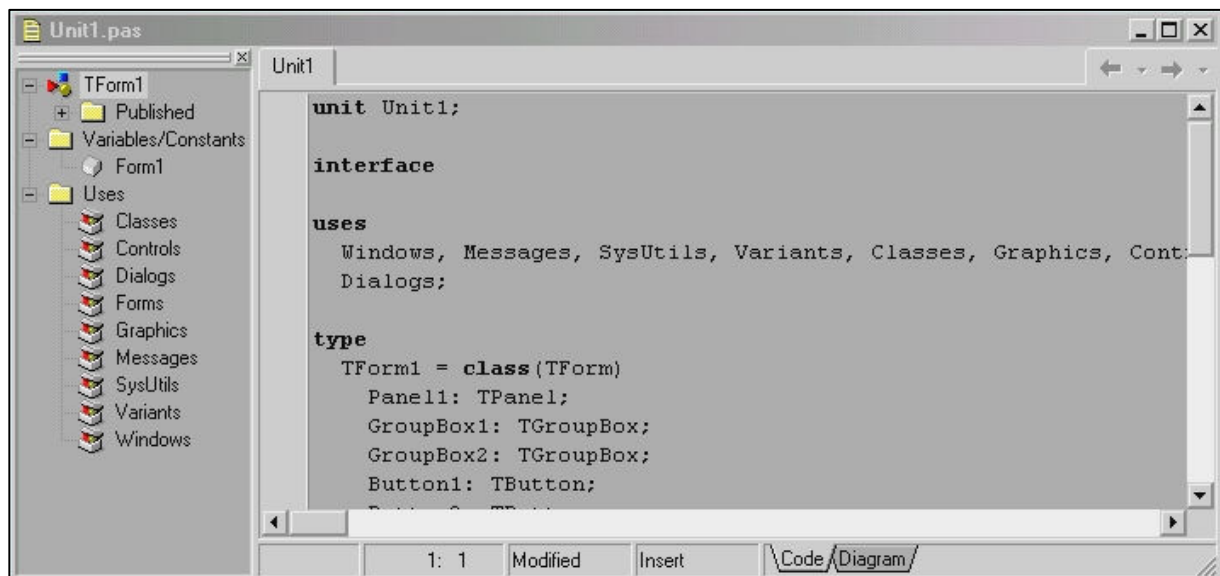
از نوع آن تعریف شده که در واقع همان فرم ما بوده و برای استفاده از خصوصیات و متدهای فرم باید به آن رجوع شود.

در صورتی که کنترل‌های دیگر و یا متدهایی به این پروژه اضافه شوند، تعریف همگی آنها در بدنه تعریف نوع TForm1 درج خواهد شد، زیرا همگی زیر مجموعه‌هایی از TForm1 خواهند بود.

وقتی یک شیء از شیء دیگر منشعب می‌شود، تمام خصوصیات، متدها و رویدادهای آن را به ارث می‌برد. در این حالت به شیء منشعب شده نسل (Descendant) و به والد آن، جد (Ancestor) گفته می‌شود.



(شکل ۲-۳ سلسله مراتب وراثت در VCL)



(شکل ۲-۴ پنجره CodeEditor)

زمانی که فیلد، خاصیت و یا متد جدیدی تعریف می‌شود، دارای یکی از ویژگیهای زیر خواهد بود.

- Private : شیء فقط از درون یونیتی که در آن تعریف شده قابل دسترس است و نه خارج از آن.

- Protected : از درون یونیت حاوی کلاس آن و دیگر کلاسهای منشعب شده از آن قابل دسترس است.

- Public : در تمام یونیتها قابل دسترس است.

- Published : مانند قبلی بوده ولی خواص آن در زمان طراحی نیز قابل دسترس هستند.

همچنین در هنگام ساخت هر یک از اشیاء، هر کدام دارای یک مالک می‌شوند که توسط خصوصیت Owner آنها مشخص می‌شود و حافظه هر شیء در هنگام آزاد شدن حافظه مالک آن آزاد خواهد شد.

به طور پیش فرض مالکیت تمام اجزایی که روی فرم قرار دارند به عهده همان فرم بوده و مالکیت خود فرم نیز به عهده برنامه آن (Application) است، از این رو در هنگام خاتمه برنامه، حافظه تمام فرمها و زیرمجموعه‌های آن نیز آزاد می‌شود.

البته مالکیت فقط مربوط به نسلهای Tcomponent می‌شود که کنترلهای خود دلفی هستند، ولی اگر خود برنامه‌نویس اقدام به ساخت شیء جدیدی بنماید، مسئولیت آزادسازی به عهده خود شیء می‌باشد.

قواعد زبان دلفی

زبان برنامه‌نویسی دلفی بر مبنای زبان پاسکال بنا نهاده شده و می‌توان گفت که نسخه‌ای بصری (Visual) از همان زبان است که برای توسعه سیستم عامل ویندوز بهینه‌سازی شده است.

با توجه به آموزش زبان پاسکال به عنوان زبانی پایه برای یادگیری اصول برنامه‌نویسی، زبان دلفی گزینه‌ای مناسب برای شروع برنامه‌نویسی شیء‌گرا و تحت سیستم عامل ویندوز است. پس برای برنامه‌نویسی با این زبان باید با قواعد (Syntax) زبان پاسکال آشنا بود. البته تغییراتی نیز در برخی از اصول، قواعد و فرمانها داده شده و قسمتهایی نیز به آنها اضافه شده است.

از این رو به شرح مختصر قواعد کلی موجود در زبان برنامه‌نویسی دلفی می‌پردازیم.

تعاریف:

تعریف نوع داده	نوع = نام Type
	private
	...
	{ تعاریف محلی }
	...
	public
	...
	{ تعاریف عمومی }
	...
	end;

Uses ؛ . . . ، نام یونیت، نام یونیت

Const مقدار = [نوع :] نام ؛ تعریف ثابتها

Var ؛ نوع : نام متغیر ؛ تعریف متغیرها

Label ؛ نوع : نام شناسه ؛ تعریف شناسه

؛ مقدار =: نام متغیر ؛ مقداردهی متغیرها

{ نام راهنما \$ } ؛ رهنمودهای کامپایلر

{ \$R نام فایل منبع } ؛ منابع مورد استفاده

(* ... *) ؛ توضیحات

// ... ؛ توضیحات

تعریف روال:

Procedure [کلمات رزرو]; ; [= مقدار ۱]; نوع پارامتر ۱ : نام پارامتر ۱) نام

Begin

...

بدنه روال

...

End;

تعریف رویه:

Function : نوع خروجی : (..... ; [= مقدار ۱]; نوع پارامتر ۱ : نام پارامتر ۱) نام
[کلمات رزرو]

Begin

...

بدنه رویه

...

End;

بلوکهای خاص فرامین:

Try

...

فرامین

...

Finally

...

فرامینی که حتماً در پایان اجرا خواهند شد

شد

...

End;

Try

...

فرامین

...

Except

...

فرامینی که در صورت خطا اجرا خواهند

...

End;

شرطها:

If شرط **Then** فرمان ;

If شرط **Then** فرمان

Else فرمان ;

Case Of نام متغیر

مقدار : فرمان ;
مقدار، مقدار : فرمان ;
محدوده مقادیر : فرمان ;

Else

فرمان

End;

If شرط **Then**

Begin

...

...

End

Else

Begin

...

End ;

حلقه‌های تکرار:

For مقدار اولیه =: متغیر **to** مقدار نهایی **do**

Begin

...

End;

While شرط صحیح **Do**

Begin

...

فرامین

...

End;

Repeat

...

فرامین

...

Until ; شرط

عملگرها:

اولویت	شرح
۱	اولویت بندی: ()
۲	اشاره گرها: @ ^ نفی: Not
۳	محاسباتی: Shl Shr * / Div Mod محاسباتی: And
۴	محاسباتی: + - Or Xor
۵	شرطی: = < > <= >= in

(جدول ۲-۴ عملگرها)

انواع داده‌ای رایج:

نوع	نام	محدوده	حجم
صحیح	Byte	۰..۲۵۵	۸ بیت
	Shortint	-۱۲۸..۱۲۷	۸ بیت
	Word	۰..۶۵۵۳۵	۱۶ بیت
	Smallint	-۳۲۷۶۸..۳۲۷۶۷	۱۶ بیت
	Cardinal	۰..۴۲۹۴۹۶۷۲۹۵	۳۲ بیت
	Integer	-۲۱۴۷۴۸۳۶۴۸..۲۱۴۷۴۸۳۶۴۷	۳۲ بیت
	Longint	-۲۱۴۷۴۸۳۶۴۸..۲۱۴۷۴۸۳۶۴۷	۳۲ بیت
	Int64	-۲ ^{۶۳} ..۱-۶۳ ^۲	۶۴ بیت
اعشاری	Single	1.5 x 10 ⁻⁴⁵ .. 3.4 x 10 ³⁸	۳۲ بیت
	Real	5.0x 10 ⁻³²⁴ .. 1.7 x 10 ³⁰⁸	۶۴ بیت
	Double	5.0 x 10 ⁻³²⁴ .. 1.7 x 10 ³⁰⁸	۶۴ بیت
	Extended	3.6 x 10 ⁻⁴⁹⁵¹ .. 1.1 x 10 ⁴⁹³²	۸۰ بیت

(جدول ۲-۵ انواع داده‌ای رایج)