راهنمای سریع

زبان برنامه‌نویسی

Delphi

نویسنده:

محمّد شمس

M.Shams.J@Gmail.Com

دلفی محصول شرکت Borland و یکی از پرکابردترین زبانهای برنامه‌نویسی موجـود در دنیـای نرم‌افزار است که امروزه در زمینه‌های مختلفی از جمله برنامـه‌هـای گرافیکـی، چنـد رسـانه‌ای، پایگاه داده و غیره به خوبی از آن استفاده می‌شود.

در حال حاضر به جرأت می‌توان گفت که یکی از پرکاربردترین زبان‌هـای برنامـه‌نویـسی تحـت سیستم عامل ویندوز، دلفی اسـت زیـرا نـه دارای پیچیـدگیها و مـشکلات زبـان C بـوده و نـه محدودیتهای زبان Basic را دارد.

امروزه با توسـعه برنامـه‌نویـسی سـاخت یافتـه و اسـتفاده از قابلیتهـای شـئ‌گرایـی (Object Oriented) قدرت این زبان به خوبی نمایان شده است، زیرا دلفی از قابلیتهای بسیار مفیدی در رابطه با برنامه‌نویسی شئ‌گرا برخوردار است.

به طور کلی میتوان گفت که دلفی تلفیقی است از اشیائ موجود در زبان برنامه‌نویسی پاسـکال استاندارد، که در حال حاضر به آن Object Pascal گفته می‌شـود، و همچنـین مجموعـه‌ای از اجزای بصری به نام VCL.

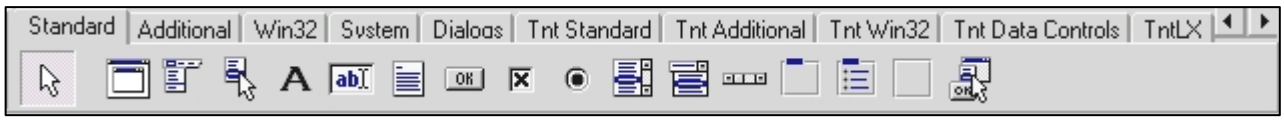
VCL یا Visual Component Library خود سلسله مراتبی است از کلاسهای نوشته شـده در شئ پاسکال و روشی است که در محیط مجتمع دلفی به کـار بـرده شـده و امکـان طراحـی سریع برنامه‌ها را فراهم می‌کند.

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 نظـر قـرار 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 می‌شود و اکثر کلاسهای دیگر از آن ارثبری می‌کنند. کامپوننتهای VCL نیز نسلی از یک کلاس به نام TComponents هستند که جد تمامی کامپوننتها و اجزای ساخت می‌باشد.

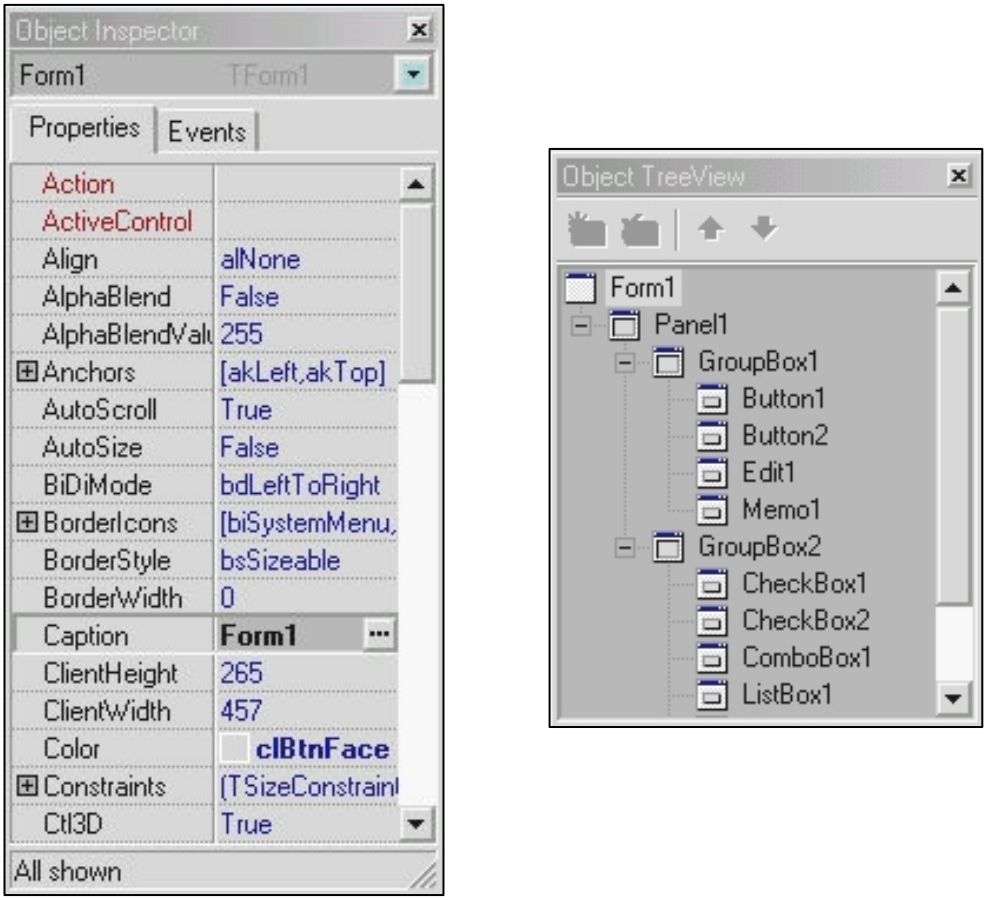
VCL بر خلاف نام آن که اجزای بصری را در ذهن شنونده تداعی می‌کند، بیشتر شامل اشیای غیر بصری است که به سادگی می‌توان آنها را به فرم مورد نظر اضافه کرده و از قابلیتهای آنها استفاده نمود.

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



(شکل ۱-۲ Component Palette)

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



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

برنامه‌نویسی شئ‌گرا

برنامه‌نویسی شئ‌گرا 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 شئ‌گرا به ارمغان آورده است.

اگر مفهوم رکورد را در زبان پاسکل درک کرده باشید، به راحتی می‌توانید مفهوم شئ 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 طریقه نام رکورد میسر است. اشیاء نیز مانند رکورد، مجموعه‌ای از عناصر داده‌ای هستند با این تفاوت که شئ 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 9X و Win NT
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
راهبری(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 کـه در جـدول زیر به شرح مهمترین آنها می‌پردازیم:

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

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

(جدول ۳-۲ 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 نام 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;
```

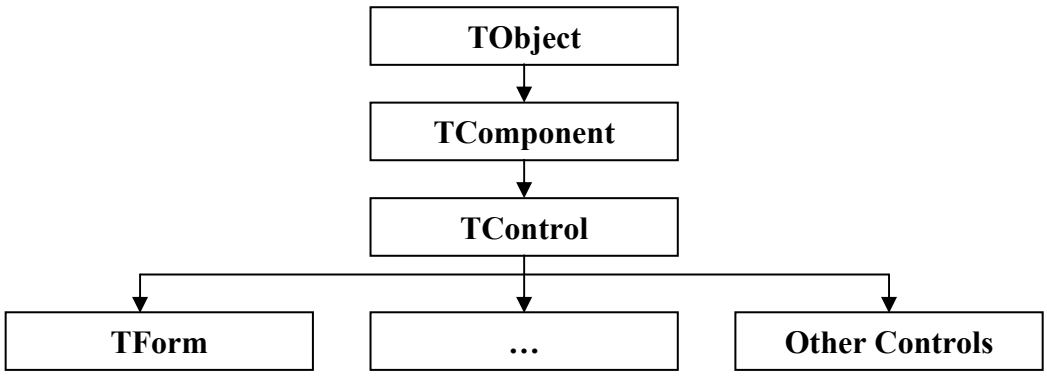
شروع قسمت پیاده‌سازی

```
Implementation
{$R *.dfm}
end.
```

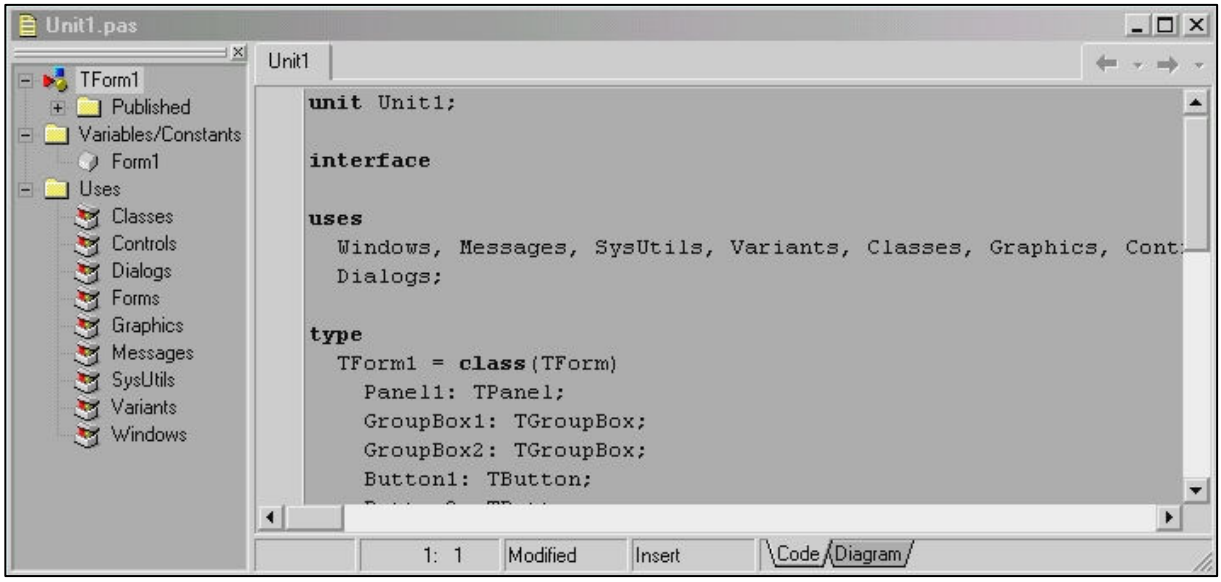
Tform1 نوع کلاس جدید می‌باشد که از نوع Tform منشعب شده و متغیری به نام Form1 از نوع آن تعریف شده که در واقع همان فرم ما بوده و برای استفاده از خصوصیات و متدهای فرم باید به آن رجوع شود.

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



(شکل ۳-۲ سلسله مراتب وراثت در VCL)



(شکل ۴-۲ پنجره CodeEditor)

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

- Private : شئ 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 شئ در هنگام آزاد شدن حافظه مالک آن آزاد خواهد شد.
به طور پیش‌فرض مالکیت تمام اجزایی که روی فرم قرار دارند به عهده همان فرم بوده ومالکیت 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 شئ جدیدی بنماید، مسئولیت آزادسازی به عهده خود شئ 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 شئ‌گرا 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

Type نام = نوع

```
private
    ...
  { تعاریف محلی }
    ...
public
    ...
  { تعاریف عمومی }
    ...
end;
```

Uses نام یونیت، نام یونیت ، . . . ;	یونیتهای مورد استفاده
Const نام [: نوع] = مقدار ;	تعریف ثابتها
Var نام متغیر : نوع ;	تعریف متغیرها
Label نام شناسه : نوع ;	تعریف شناسه
نام متغیر =: مقدار ;	مقداردهی متغیرها
{$ نام راهنما }	رهنمودهای کامپایلر
{$R نام فایل منبع}	منابع مورد استفاده
(* ... *)	توضیحات
// ...	توضیحات

تعریف روال:

Procedure نام (نام پارامتر۱ : نوع پارامتر۱ [= مقدار۱]; ..... ; .....); [کلمات رزرو]

Begin

…
بدنه روال
…

End;

تعریف رویه:

Function نام (نام پارامتر۱ : نوع پارامتر۱ [= مقدار۱]; ..... ; .....) : نوع خروجی ;
[کلمات رزرو]

Begin

…
بدنه رویه
…

End;

بلوکهای خاص فرامین:

Try

…

فرامین

…

Except

…

فرامینی که در صورت خطا اجـرا خواهنـد شد

…

End;

Try

…

فرامین

…

Finally

…

فرامینی که حتماً در پایان اجرا خواهند شد

…

End;

شرطها:

If شرط **Then** فرمان ;

If شرط **Then** فرمان
Else فرمان ;

Case نام متغیر **Of**

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

Else

فرمان

End;

If شرط **Then**

Begin

…

…

End

Else

Begin

…

End ;

حلقه‌های تکرار:

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

Begin
…
End;

While شرط صحیح **Do**

Begin
…
فرامین
…
End;

Repeat

…
فرامین
…
Until شرط ;

عملگرها:

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

(جدول ۴-۲ عملگرها)

انواع داده‌ای رایج:

نوع	نام	محدوده	حجم
صحیح	Byte	۲۵۵..۰	۸ بیت
	Shortint	-۱۲۸..۱۲۷	۸ بیت
	Word	۶۵۵۳۵..۰	۱۶ بیت
	Smallint	-۳۲۷۶۸..۳۲۷۶۷	۱۶ بیت
	Cardinal	۴۲۹۴۹۶۷۲۹۵..۰	۳۲ بیت
	Integer	-۲۱۴۷۴۸۳۶۴۸..۲۱۴۷۴۸۳۶۴۷	۳۲ بیت
	Longint	-۲۱۴۷۴۸۳۶۴۸..۲۱۴۷۴۸۳۶۴۷	۳۲ بیت
	Int64	-۲^۶۳..۲^۶۳-۱	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 ۵-۲ انواع داده‌ای رایج)