

ساختار کلی :

تا کنون کلیات و مفاهیم اولیه که در برنامه نویسی دانستن آنها به نوعی لازم است با شرح و تفصیل

بیان شد. حال به نکات کمی عملی تری می پردازیم. بدنه کلی برنامه *Delphi* بصورت زیر است:

```
Program < ProgramName > ;  
uses <UnitName1 > [<UnitName2 > ...];  
M //Subroutines  
const <Constant Definitions >  
var <variable Definitions >  
begin  
M //Program body  
end.
```

ساختار 1



```

Unit <UnitName>;

interface

uses <UnitName1>[, <UnitName2> ...];

const <Constant Definitions>;

type <type Definitions>

var <variable Definitions>

M //SubroutineDefinitions

implementation

M //Subroutine &Class implementations

begin

M //Initialization codes

end.

```

ساختار 2

حال به تفصیل به بخشهای متفاوت آن می پردازیم. در ابتدا یک تقسیم بندی که در فصل اول به آن اشاره نشد ذکر می کنیم. برنامه های نوشته شده در *Delphi* دو نوع فایل دارند. یکی فایل پروژه که بصورت ساختار 1 است و دیگری فایل های کتابخانه ای که به ساختار 2 هستند.

این تقسیم بندی برای طبقه بندی و جداسازی متن برنامه نوشته شده است.

به خاطر داشته باشید که از این به بعد از نام " کد¹" برای بیان متن برنامه استفاده خواهیم کرد.

معمولاً یک پروژه برنامه *Delphi* حداقل شامل یک فایل پروژه (اجباری) و یک کتابخانه (اختیاری)

است که در کتابخانه کدهای لازم برای نمایش یک پنجره و در پروژه مقدمات اجرای برنامه تحت ویندوز

نوشته می شود در ابتدا که پروژه جدید ایجاد می کنید کدهای اولیه لازم برای هر دوی این فایلها بصورت خود کار نوشته شده است و قابل اجراست. در برنامه های دلفی به ازای هر پنجره که به آن *Form* گویند یک کتابخانه ایجاد خواهد شد. حال به توضیح جزئیات هر یک می پردازیم. اگر در کار کردن با محیط دلفی مشکل دارید به 1-1-8-6 و 2-1-8-6 مراجعه کنید.

1. بخش 1 پروژه :

program : در این بخش بجای *< ProgamName >* اسم برنامه خط را بنویسید نوشتن کل این بخش اختیاری است و فقط جهت تفکیک محتوا بکار می آید و به نوعی یک توضیح² است.

uses: در این بخش کتابخانه هایی را که استفاده کرده اید می نویسید، معمولاً این بخش توسط خود *Delphi* در حین افزودن کتابخانه اضافه می شود و نیازی به تغییرات دستی ندارد هر چند گاهی اوقات ممکن است این کار را انجام دهید.

Subroutines //: در این بخش زیر برنامه های احتمالی نوشته می شود. شکل و فرم نوشتن زیر برنامه در این بخش در 1-5-6 توضیح داده شده .

const: در این بخش ثابت های مورد استفاده در برنامه را تعریف می کنید. ثوابت در برنامه بدین علت استفاده می شوند که بجای اعداد عبارات با مفهوم نوشته شود و لذا فهم آن آسانتر باشد علاوه بر آنکه تصحیح برنامه به میزان زیادی ساده می شود. جزئیات تعریف ثوابت در بخش 1-4-6 آمده است.

var: در این بخش متغیرهای مورد استفاده در برنامه تعریف می شوند. متغیرها در واقع مکانهایی از حافظه هستند که به شکل خاصی با آنها برخورد می شود مثلاً محتوای آنها مانند عدد برخورد می شود یا

بصورت متن یا هر چیز دیگر، جزئیات تعاریف متغیر همه در 6-4-1-1 آمده. تمامی داده ها که در فصل 6-2 به آن اشاره شد متغیر هستند.

Program body:// این بخش بدنه برنامه است که متن مقداردی اولیه و بقیه موارد لازم جهت شروع برنامه ویندوز در آن نوشته شده در تغییر دادن این بخش باید دقت کنید. هیچکدام از دستورات موجود را حذف نکنید. معمولاً نیازی به تغییر این بخش دیده نمی شود ولی جهت آشنایی دستورات موجود توضیح داده می شود. از آنجا که برنامه نویسی تحت Windows مورد بحث ما نیست به راحتی می توانید این بخش را کنار گذاشته (بخش *Program body* را) و عنوان بعدی را مطالعه کنید.

در برنامه نویسی Windows معمولاً روال بر پردازش پیغام های رسیده از سیستم عامل است به این صورت که ابتدا پنجره های مورد نیاز ایجاد می شود و سپس برنامه در یک حلقه قرار می گیرد که پیغام های رسیده از سیستم عامل را پردازش و مدیریت می کند بصورت خلاصه *Application CreatForm* همان بخش ایجاد پنجره هاست که در عین حال اولین پنجره ایجاد شده را هم پنجره اصلی برنامه قرار می دهد و به این وسیله باعث می شود بستن این پنجره کل برنامه را متوقف کند. *Application*.run هم بخش حلقه ذکر شده است که در آن پیغام ها پردازش می شود. این پیغام ها محتویاتی از قبیل اینکه کلیدی توسط کاربر زاده شده، موشواره حرکت کرده و یا کلیک شده و ... است.

بنابراین می توانید با تغییر این بخش اجرای برنامه را منوط به مساله خاصی کنید و یا اصولاً پنجره قرار ندهید و بدون قرار گرفتن در حلقه برنامه را اجرا کنید هر چند اصلاً توصیه چنین کاری نمی شود.

2. بخش 2 کتابخانه

Unit: بجای *< UnitName >* اسم کتابخانه را بنویسید، این نام اجباری است و در فایل‌های دیگر این

نام استفاده می‌شود.

این نام در بخش *Uses* دیگر فایل‌های می‌آید.

کلمه *Interface* یک کلمه اجباری است و دقیقاً در همان جا که ذکر شده باید قرار گیرد.

Uses: دقیقاً مشابه همان عبارت *Uses* در پروژه فایل‌های کتابخانه دیگری را ضمیمه این کتابخانه می

کند تا بتوان از آنها در این کتابخانه استفاده کرد. به تدریج که با برنامه کار کنید علت این امر روشن خواهد شد.

const: بخش تعریف ثوابت این ثوابت در فایل‌هایی که از کتابخانه استفاده کنند تعریف شده خواهد

بود.

var: بخش تعریف متغیرها: این متغیرها در فایل‌هایی که از کتابخانه استفاده کنند تعریف شده خواهد

بود.

// SubroutineDeclarations: در این بخش تعریف زیر برنامه‌ها می‌آید که در فصل 6-5-1 به

تفصیل در مورد آن سخن گفته شده.

کلمه *Implementetion* هم اجباری است. پس از این کلمه هم می‌توان *var, const, uses* تعریف کرد

ولی آن دسته از ثوابت و متغیرها که در این جا تعریف شوند در فایل‌هایی که از کتابخانه استفاده می‌کنند

تعریف شده نخواهند بود به عبارت دیگر محلی اند.

Imp Subroutin & Clans: در این بخش متن زیر برنامه‌ها و کلاسهای تعریف شده قرار دارد که

در فصل 6-5-1 بررسی می‌شود منظور از پیاده سازی کلاس پیاده سازی توابع کلاسها (در *Delphi* به اشیا

Class و به نمونه از شیئی *Object* گفته می شود) همان متن زیر برنامه های درون کلاسها و همچنین *destructor, Constructor* احتمالی است که اگر با شیئی گرایشی آشنا ندارید نیاز به توجه به این فایلها نیست.

// Intialization Codes : این بخش مقدار دهی های اولیه برای متغیرهای محلی و عمومی کتابخانه

انجام می شود. که معمولاً نیازی به نوشتن این بخش هم نیست البته اگر این کدها را نخواستید فقط *begin* را حذف کنید! پیش فرض کتابخانه در *Delphi* فاقد مقدار دهی اولیه است لذا در واقع اگر بخواهید اینها وجود داشته باشد باید *begin* را قرار دهید.

¹code

²Comment

