

## متغیرهای زیر برنامه ای :

پس از این بحث مفصل به یک نوع داده ای که تاکنون نمی توانستیم مطرح کنیم می پردازیم. متغیرها می توانند از نوع تابع باشند و حتی اجرا شوند! این کار بدین علت است که چنانکه گفته شد توابع هم در حافظه ذخیره می شوند و لذا قاعده‌تاً با آدرسشان در حافظه قابل دستیابی اند.

نحوه تعریف کلی به سادگی قابل بیان نیست و چند نکته دارد لذا با چند مثال این نوع تعریف را انجام می دهیم. شما سعی کنید با تعمیم این نحوه تعریف و استفاده از آن دیدگاه خود را در این باب مورد آزمایش قرار دهید.

تعاریفی مانند زیر همه متغیرهای زیر برنامه ای هستند:

```
var
a : procedure( x : interger ; y : interger );
b : function( m , n : interger ) : interger ;
c : procedure ;
d : function : boolean ;
M
begin
a( k , l );
k := b( k , l );
c ;
if d then
M
end.
```

حدس می زنید این متغیرها چه کاربردی داشته باشند؟

یکی از فوائد بارز این نوع متغیرها در بخش نحوه ارتباط با کاربر گفته شد. نسبت دادن زیر برنامه های مورد نظر ما به متغیرهای تابعی می تواند این کار را ممکن کند که در مواقع مقتضی پاسخ مورد نظر ما به رخدادها اتفاق افتد مثلاً فرض کنید بدنه درونی برنامه که خود دلفی نوشته هنگام فشردن کلیدهای کارکتری در بخشی دستورات زیر را اجرا می کند:

```
if assigned (FOnKeyPressed) then
  FOnKeyPressed (key );
```

که *FOnKeyPressed* یک متغیر تابعی است و *assigned* نشان می دهد آیا مقدار دارد یا خیر. حال کافیسست تابع مورد نظر خود را به متغیر *FOnKeyPressed* نسبت دهیم هرگاه کلید فشرده شود برنامه ما اجرا خواهد شد.

نحوه مقدار دادن به متغیرهای تابعی و تهی کردن آنها بسیار ساده و بصورت زیر است

مقدار دادن متغیر

```
< proc _var_name >:=< procedure _name >;
```

تهی کردن متغیر

```
< proc _var_name >:= nil;
```

مثلاً



```
function max( x , y : interger ) : interger ;
```

**M**

```
var
```

```
a : function( x , y : interger ) : interger ;
```

```
begin
```

```
a := nil ;
```

**M**

```
a := max ;
```

```
end.
```

این متغیرها علاوه بر کاربرد بالا در بالا بردن سرعت اجرای برنامه ها هم کارآمد هستند و لذا بسیار

کارآمد هستند. روی نحوه استفاده از این متغیرها در افزودن سرعت برنامه فکر کنید.

