

اشاره گرها:

در فصل اول گفتیم داده ها در حافظه ذخیره می شود، اشاره گر ها در واقع مشخص کننده مکان شروع حافظه داده ها هستند. لذا اشاره گرها به خودی خود اهمیت زیادی ندارد و فقط به مکانی از حافظه اشاره می کنند که در آن داده قرار گرفته.

البته گاهی اوقات این قاعده نقض می شود و اشاره گرهایی فقط صرف اشاره به حافظه استفاده می شوند.

در زبان *Delphi* امکان تعریف اشاره گر به هر دو صورت وجود دارد.

```
type
    PX :Pointer           ;// Pointer to memory      نوع 1
    PI : ^Integer;       // Pointer to integer
    PC ; ^Char;          // Pointer to character     نوع 2
    M
```

اشاره گر به مکان حافظه بدون نوع عمدتاً جایی کاربرد دارد که هر نوع داده ای را در مکان خاص حافظه می توانیم داشته باشیم. این کار در برنامه نویسی عادی لزومی ندارد کاربرد دیگر آن بنابه مورد استفاده توضیح داده خواهد شد.

اشاره گری که نوع دارد. هم کاربردهای زیادی دارد. از جمله آن استفاده در لیست های پیوندی است که در ادامه همین فصل توضیح داده شده است.
نحوه استفاده از آنها هم بصورت زیر است.

```

var
  X : ^integer;
M
  X ^ := 10;      //Content of where X points to
  X := $70001555; //Value of Address

```

X^{\wedge} محتوی حافظه ای است که X به آن اشاره می کند و خود X آدرس آن مکان حافظه است.

برای تهی کردن آدرس از دستور $X := nil;$

استفاده می کنیم. nil اشاره گر به هیچ مکان حافظه نیست لذا با دستور بالا دیگر X به هیچ مکان

حافظه اشاره نمی کند.

اشاره گرها بصورت های دیگری هم تعریف می شوند یکی همان متغیرهای نوع زیر برنامه است که

مطرح شد (6-5-2). دیدید که برای اینکه متغیر به هیچ مکانی اشاره نکند از nil استفاده کردیم.

یک کاربرد عام اشاره گرها اختصاص حافظه در حین اجرای برنامه است تنها نکته این است که حتماً

پس از اتمام استفاده حافظه آن را آزاد باید کرد. روش کار بصورت زیر است.

```

var
  X : ^Integer;
M
  New (X );      //AllocateMemory to X
try
  //Do everything necessary
finally
  Dispose(X );  //Deallocate Memory of X
end;

```

ساختار try در بخش اشکال زدایی مفصلاً توضیح داده خواهد شد فقط مختصراً بگوئیم که مجموعه

دستورات فوق باعث می شود در صورت رخدادن هر اتفاقی در فاصله $try, finally$ دستور $Dispose$ اجرا

شود.

دقت کنید برای استفاده از $X^a := a$ حتماً باید یکی از دو حالت زیر باشد یا حافظه به آن اختصاص

داده باشید و یا X به محل یک متغیر دیگر اشاره کند. محل متغیرها با استفاده از عملگر @ بدست می آید.

```
var X : ^Integer; y : Integer;
```

M

```
X := @Y;
```



Olympiad.roshd.ir