رکوردها:

در ابتدای فصل از فوائد انتزاع داده جمع کردن اطلاعات همرده در زیر لوای یک متغیر به عنوان یکی از مهمترین مزایای انتزاع انواع داده مطرح شد. این کار توسط رکوردها امکان پذیر است. برای توضیح از نحوه تعریف شروع می کنیم:

$$
\begin{array}{l}
type \left\langle \begin{array}{l} TRecord \\ \\ Name \end{array} \right\rangle = recond \\
\quad < element\,1 >:< \mathrm{var}-type > \\
\quad [< element\,2 >:< \mathrm{var}-type > \\
\qquad\qquad \mathbb{M} \qquad\qquad] \\
end;
\end{array}
$$

که $< TRercordName >$ اسم نوع داده جدید، $< element >$ اسم عنصر جدید و $< var-type >$ نوع متغیر است. برای یکجا جمع داده های مربوط مثل بخشهای حقیقی و موهومی عدد مختلط از همین روش استفاده می شود:

```
type   TComplex = record
       RealPart : real;
       ImagPart : real;
end;
```

که بخشهای $realPart$ و $ImagPart$ یک عدد $TComplex$ را کنار هم جمع کرده نحوه استفاده از این نوع داده هم بسیار ساده است پس از تعریف یک متغیر از نوع رکورد بصورت زیر از آن استفاده می کنیم:

```
var
X :<TRecordName >;
M
X .<element1>:=<Value >;
```

به عنوان مثال در همین مورد *Tcomplex*

```
var
  X :Tcomplex;
M
X.realPart := 123.4;
X.ImagPart := 14.5;
```

دیده می شود این کارچگونه یکجا جمع کردن (کپسوله کردن) را انجام می دهد.

انواع رکورد به شرح زیرند.

رکوردهای معمولی : که همان موارد توضیح داده شده است.

رکوردهای چند طبقه 1: این نوع داده وقتی پیدا می شود که درون اعضای یک رکورد از رکورد دیگری استفاده کنید مثلاً

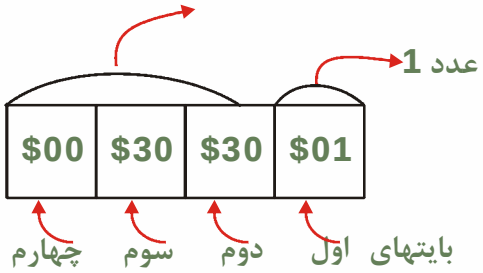
```
Type
  TComplex =
              M
  THierarchical = record
              C1:TComplex;
              C2: real;
end;
```

نکته خاصی در این مورد وجود ندارد تنها دلیل طرح آن بیان این نکته بود که امکان چنین تعریفهایی وجود دارد.

رکوردهای شرطی : این رکوردها وقتی کارآیی دارند که اطلاعات مورد نیاز در مورد یک موجودیت خاص بر حسب برخی خصوصیات آن متفاوت می شود. مثلاً حساب بانکی بر حسب قرض الحسنه بودن یا سپرده بودن اطلاعات متفاوتی دارد. این نوع رکوردها بصورت زیر تعریف می گردد:

```
Type
  <TRecord  Name >= record
        <element >:< var−type >;
           M
        case <element  n >:<ordinal  var−type >of
           <value1>:(<element  n+1>:< var−type >[;<element  n+2>:< var−type >]);
                M
end;
```

در این حالت تا $<element\ n>$ **اوضاع کاملاً مشابه استفاده از رکوردهای معمولی است تنها تفاوت این است که نوع** $<element\ n>$ **حتماً از انواع شمارشی یعنی اعداد صحیح، حروف** $(Char)$ **و** $Enum$ **ها است. بخش پس از** $<element\ n>$ **بسته به مقدار** $<element\ n>$ **باید از هر کدام از ادامه ساختارهای نوشته شده در جلوی مقدار مربوطه** $(Value)$ **استفاده کنیم. در واقع داده های بعد المان** n **ام که در حافظه چیده شده اند می توان بصورت چند دید متفاوت نگاه کنید. مثل آنکه می توانید از یک عدد** $Interger$ **چهار بـایتی بایـت اول را بصورت** $Byte$ **و سه بایت بعد را بصورت سه** $Char$ **بخوانید. مثال:**



کد عدد حرف " Æ " عدد $30 است.

این روش باعث صرفه جویی در حافظه می شود

مثال.

```
Type  TConditionalRec = record
                  X  : Byte;
                  caseY  : Byte  of
                      1: (Z1 : real; Z2 : real);
                      2: (Z3 : Integer; Z4 : Integer);
  end;
var   X  :TConditional  Record
```

در نمونه بالا بسته به اینکه مقدار Y چه باشد می توانیم از Z_1 و Z_2 و یا Z_4, Z_3 استفاده کنیم. دقت شود که اینکار فقط نگریستن های متفاوت به یک بخش حافظه است.

Hierarchical [1]