

Записи с вариантной частью

Иногда в записях в зависимости от ситуации могут присутствовать различные наборы полей или поля могут иметь различные типы. Было бы нерационально отводить место под все возможные поля, если все они одновременно все равно не используются. В этих случаях можно вводить в запись вариантную часть, в которой перечисляются варианты полей и условия их использования.

Вариантная часть оформляется как оператор множественного выбора **case**, который размещается в объявлении записи после объявлений обязательных полей:

```
type <имя типа записи> = record
  <список имен обязательных полей> : <тип 1>;
  . . . . .
  <список имен обязательных полей> : <тип N>;

  case <имя селектора> : <порядковый тип> of
    <список значений 1> : (<вариант 1>);
    <список значений N> : (<вариант N>);
end;
```

Вариантная часть начинается ключевым словом **case**. Имя селектора - любой допустимый идентификатор. Его указание не обязательно. Если имя селектора указывается, то в запись включается **дополнительное поле** с соответствующим именем, по значению которого можно узнать, какой именно вариант полей используется. Например:

```
type TCaseRec = record
  fTitle : string[32];
  case Num : byte of
    0: (f1 : byte);
    1: (f2 : word);
    2: (f3 : integer);
end;
```

В данном примере **Num** - это имя дополнительного поля.

Если имя селектора не указано, то в объявлении записывается только **тип селекторного поля** и, соответственно, не записывается и символ двоеточия после него. Например:

```
type TCaseRec = record
  fTitle : string[32];
  case byte of
    0: (f1 : byte);
    1: (f2 : word);
    2: (f3 : integer);
end;
```

При отсутствии имени селектора программа по каким-то другим признакам должна разбираться, какие поля записи в данный момент используются.

После имени селектора или сразу после слова **case**, если имя селектора отсутствует, указывается **порядковый тип**. Например: **Boolean, Byte, Word, Integer, Cardinal**, и.т.п. Например:

```

type TCaseRec = record
  fTitle : string[32];
  case Boolean of
    True   : (f1 : byte);
    False  : (f2 : word);
end;

```

Допустимо также в качестве порядкового типа использовать перечисления, например:

```

type
  TShapeList = (Rectangle, Triangle, Circle, Ellipse, Other);
TFigure = record
  fTitle : string[32];
  case TShapeList of
    Rectangle : (Height, Width : Real);
    Triangle  : (Side1, Side2, Angle : Real);
    Circle     : (Radius : Real);
    Ellipse, Other : ();
end;

```

Все имена полей, которые используются в записи, включая ее вариантную часть, **должны быть уникальными**.

Константы записей с вариантной частью

Константы записей с вариантной частью записываются путем перечисления с разделителем «;» полей **одного** из вариантов записи. Законченное перечисление заключается в круглые скобки. Поле записи кодируется его именем и разделяется с константой значения в этом поле разделителем «:». Имена полей должны соответствовать выбранному варианту записи. Например, запишем все возможные константы для следующего объявления вариантной записи:

```

type TCaseRec = record
  fTitle : string[32];
  case Boolean of
    True   : (f1 : byte);
    False  : (f2 : word);
end;

```

Первый вариант константы может иметь вид:

```

const
cCaseRec1 : TCaseRec =
(
  fTitle : 'True - вариант';
  f1      : 01
);

```

Второй констант соответствен должен иметь такой вид:

```

const
cCaseRec2 : TCaseRec =
(
  fTitle : 'False - вариант';
  f2      : 00000

```

```
);
```

Еще один пример констант для нулевой и второй вариантной записи с именем селектора **Num**:

```
type TCaseRec = record
  fTitle : string[32];
  case Num : byte of
    0: (f1 : byte);
    1: (f2 : word);
    2: (f3 : integer);
end;

const
cCaseRec0 : TCaseRec =
  (
    fTitle : '0 - вариант записи';
    Num      : 0;
    f1       : 255
  );

const
cCaseRec2 : TCaseRec =
  (
    fTitle : '2 - вариант записи';
    Num      : 2;
    f3       : -1
  );
```