Тип данных запись

Лекция №9

Пример. Необходимо обработать результаты зимней сессии студенческой группы:

	ФИО	**№ зачетки**	**Оценки**	**Стипендия**
1.	Иванов И.И.	12345678	2 4 4 3	0
2.	Петров В.К.	12345679	4 4 5 4	4500.00
3.	Сидоров П.И.	12345680	4 4 4 3	4000.00
4.	...			

В таких случаях используют тип запись. Записи используются для представления **разнотипной**, но логически связанной между собой информации.

Например, для обработки информации об абонентах телефонной связи, информации о библиотечных книгах, информации о товарообороте газетного киоска и т.д.

Структура записи

Компоненты (элементы) записи определяются не индексами, как в массиве, а полями. Каждое поле имеет свое имя, которое должно нести информацию о содержании поля и чаще всего совпадает с названием столбца таблицы.

Имя поля

ФИО	***№ зачетки***	***Оценки***	***Стипендия***
Иванов И.И.	12345678	2 4 4 3	0

значение поля

Записи с фиксированными полями

Запись – это структура данных, состоящая из фиксированного числа **разнотипных** компонентов, определяемых **полями**.

При описании структуры записи указывается список полей. Для каждого поля указывается тип данных, которые будут храниться в данном поле.

```
Type <имя типа>=record
        <имя поля1>: <тип>;
        <имя поля2>: <тип>;
        .....
        end;
```

```
Type Student=record
        Fio:string[50];
        N_zach:string[8];
        Mark:array[1..4]of byte;
        Stipend:real;
      End;
Var St1, St2: Student;
```

В качестве базового типа для полей записи можно использовать любой простой или структурный тип, в том числе и тип запись, как определенные ранее, так и определяемые внутри записи.

```
TYPE
{запись ФИО}
FIO= record
        Fam, name, otch: string[15];
         End;

{запись о студенте}
Stud_session= record
      Name:FIO;
      Ocenka: record
                Mat: 1..5;
                      Alg: 1..5;
                Ist:1..5;
                      Inf:boolean;
              end;
      End;
```

Типизированные константы

Константа типа запись также как и константа типа массив, задается списком компонентов, но при этом указываются поля записи.

Пример

CONST

ST:Student= (Fio:’Иванов И.И.’; N_zach:’ 34507891’; Mark: (4,4,5,5); Stipend:3500.0);

Доступ к полям записи

Доступ к полям записи осуществляется указанием так называемого составного имени: имени переменной (записи) и имени поля.

Пример

Var St1: Student;

Имя переменной St1 и соответствующего поля разделяются точкой.

St1. Fio:='Иванов И.И.';
St1. N_zach:=' 34507891';
St1. Mark[1]:=4;
St1. Mark[2]:=4
St1. Mark[3]:=5;
St1. Mark[4]:=5;
St1. stipend:= 3500;

В случае более сложной структуры, такой как типа Stud_session, составное имя удлиняется.

H.Name.Fam:='Параськин';

H.Ocenka.Alg:=4;

В том случае, если доступ к полям одной и той же записи осуществляется многократно, целесообразно обращаться к полям записи с использованием оператора присоединения **With**, который присоединяет все указанные поля к одной записи.

Пример

```
With St1 do
    begin
     Fio:='Иванов И.И.';
     N_zach:=' 34507891';
     Mark[1]:=4;
     Mark[2]:=4
     Mark[3]:=5;
     Mark[4]:=5;
     stipend:= 3500;
    end;
```

Присваивание записей

Присваивание записей также как и у массивов возможно только при совпадении типов, например, для переменных:

St1:=St2;

Ввод/вывод записей

Ввод/вывод записей как и в массиве осуществляется поэлементно, т.е. каждое поле вводится/выводится отдельно в соответствии с правилами соответствующего типа.

Пример

```
Write('введите ФИО:');
Readln(St2.fio);
Write('Введите № зачетки:');
Readln(St2.N_zach);
Write('Введите оценки сессии:');
For i:=1 to 4 do
  Readln(St2.mark[i]);
Write ('Введите начисление стипендии');
Readln (St2.stipend);
```

Задача

1) Опишите массив, элементами которого являются записи с полями ФИО и дата рождения.
2) Дан массив, содержащий информацию об успеваемости учащихся группы по дисциплине: ФИО, оценки за семестр. Написать функцию, вычисляющую средний балл группы.
3) Вывести список студентов группы по убыванию среднего балла.

Вариантные записи

Иногда необходимо менять список полей записи внутри одной записи. В таких случаях используют вариантные записи, которые в зависимости от какого-либо признака позволяют выбрать список полей.

Например, для представления данных о геометрических фигурах необходима такая структура, которая

- для окружности содержала бы поля координат центра окружности и поле радиуса,
- для прямоугольника – поле координат левого верхнего угла, поля значений длины и ширины,
- для квадрата – поле координат левого верхнего угла, поле значения стороны.
- Для параллелепипеда - поле координат левого верхнего угла, поле длины, поле ширины, поле высоты.

Для реализации такой структуры запись кроме фиксированного списка полей может иметь еще и вариантную часть.

Выбор вариантной части осуществляется с помощью специального поля – *поля признака*.

TYPE

<имя записи>=record

{фиксированная часть}

<имя поля1>: <тип>;

<имя поля2>: <тип>;

.....

{вариантная часть}

case **<имя поля признака>: <тип>** of

<конст1>:(<имя поля1>: <тип>; <имя поля2>: <тип>;…);

<конст2>:(<имя поля1>: <тип>; <имя поля2>: <тип>; ...);

.....

end;

Пример вариантной записи:

```
TYPE Figure=(Circle, Rectange, Square);
    Param_F=record
     Name: string[10];
     pixc:record
          x,y:integer;
               end;
     Case Pr: Figure of
          Circle: (radius: real);
          Rectange: (a,b: real);
          Square: (d:real);
     End;
```

```
Var i, j:Byte;
   Fig:Array[1..10]Of Figura;
Procedure Figur_Meny;
  Begin
    Writeln('1-Окружность');
    Writeln('2-Прямоугольник');
    Writeln('3-Квадрат');
  End;

Procedure Zapol;
  Begin
    For j:=1 To 3 Do
      With Fig[j] Do
        Begin
          Writeln ('Введите название Фигуры: ');
          Figur_Meny;
          Readln(i);
```

```
Case i Of
   1:Begin
       Name:='Окружность';
       Pr:=Circle;
       Write('Введите коор. центра (X,Y):');
       Readln(X,Y);
       Write('Введите радиус: ');
       Readln(R);
     End;
   2:Begin
       Name:='Прямоугольник';
       Pr:=Rectange;
       Write('Введите стор. прямоуг-ка:');
       Readln(A, B);
     End;
   3:Begin
        Name:='Квадрат';
        Pr:=Square;
        Write('Введ. значение стороны: ');
        readln(z);
      end;
  end;
 end;
end;
```

```
procedure vivod;
  begin
     for j:=1 to 3 do
     with fig[j] do
       begin
          case pr of
           circle:writeln(name:10, x:10:2, y:10:2, r:10:2);
           rectange:writeln(name:10, a:10:2, b:10:2);
           square: writeln(name:10, z:10);
          end;
       end;
  end;

BEGIN
clrscr;
Zapol;
writeln('********************');
vivod;
END.
```

Преобразование типов с помощью вариантной записи

Для размещения переменной типа запись всегда отводится фиксированный объем памяти в соответствии с объемом, занимаемым самым большим из вариантов, т.е. различные варианты размещаются на одном участке памяти, как бы «накладываясь» друг на друга. Именно эта особенность записей может использоваться для неявного преобразования типов данных. Поскольку различные варианты ссылаются на один и тот же участок памяти, можно обращаться к содержимому памяти поочередно, то как к переменной одного типа, то как к переменной другого типа.

Пример

```
var z:record case byte of
      1:(a:integer);
      2:(b:real);
      3:(ch:char);
    end;
begin
     readln(z.a);
     writeln(z.a);  {вывод значения в формате целого числа}
     writeln(z.b:10:10); {вывод значения в формате вещест. числа}
     writeln(z.ch); {вывод значения в формате ASCII-кодов}
readln
end.
```