

Программирование.  
Объединение  
разнородных  
элементов.

**Записи в Pascal**

Учитель информатики - Комкова М.С.  
Гимназия МИИТ, г. Москва

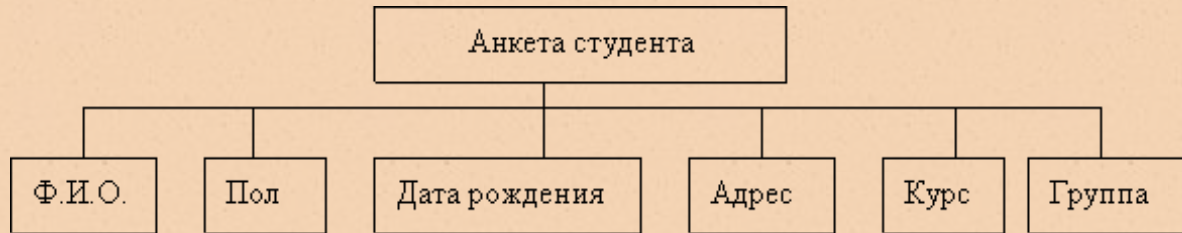
- 0 Массивы объединяют однородные единицы информации – элементы одного и того же типа.
- 0 Многообразие информации нельзя свести только к какому-то одному типу данных.
- 0 Указывая положение точки в пространстве, мы можем воспользоваться одним и тем же типом для указания ее координат, но, описывая человека, мы должны указать его имя, рост, цвет глаз и волос, то есть в одном описании объединим **разнородную** информацию.

- 0 Данные описывающие существенные стороны того или иного объекта путем включения в описание нескольких, часто разнотипных, элементов, называют **записью (record)**.
- 0 В языке **Паскаль запись** определяется путем указания служебного слова **record** и перечисления входящих в запись элементов с указанием типов этих элементов.

# Определение

**Запись – структурированный  
комбинированный тип данных,  
состоящий из фиксированного  
числа компонент (полей)  
разного типа.**

# Пример



**type < имя \_ типа >=record**

**<имя\_поля1>: тип;**

**<имя\_поля2>: тип;**

.....

**<имя\_поля К >: тип**

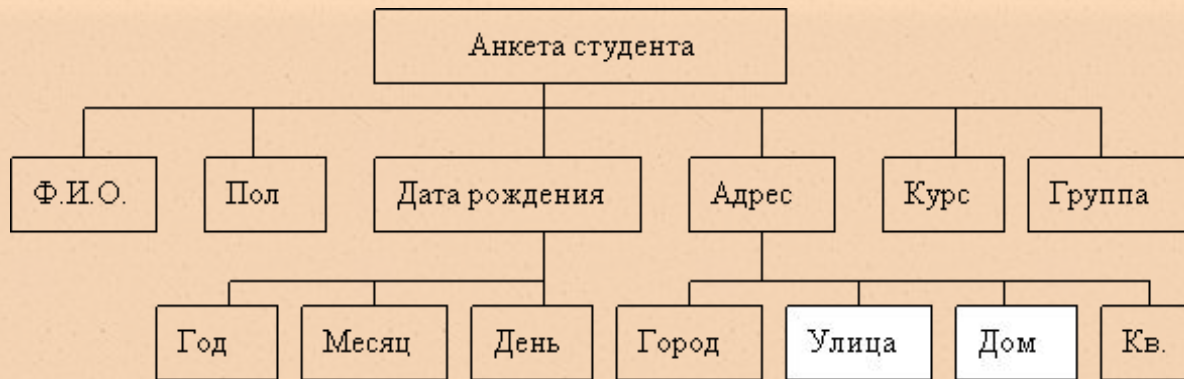
**end ;**

где record – служебное слово, а <имя\_типа>  
и <имя\_поля> - правильные  
идентификаторы языка Паскаль.

# Пример фрагмента программы описания записи Паскаля

```
Type anketa=record  
  fio: string[45];  
  pol: char;  
  dat_r: string[8];  
  adres: string[50];  
  curs: 1..5;  
  grupp: string[3];  
end;
```

Такая запись называется двухуровневой



## Пример фрагмента программы описания записи Паскаля

```
Type anketa1=record
```

```
  fio: string[45];
```

```
  pol: char;
```

```
  dat_r: record;
```

```
  god: integer;
```

```
  mes: string[10];
```

```
  den: 1..31;
```

```
end;
```

```
  adres: record
```

```
    gorod: string[25];
```

```
    ulica: string [20];
```

```
    dom: integer;
```

# Задача 1.

Пусть нам необходимо заполнить сведения о студенте (Ф.И.О., дата рождения, адрес, курс и группа), а затем вывести эти сведения на экран.

```
program primer1;
type anketa=record
  fio: string[45];
  dat_r: string[8];
  adres: string[50];
  curs: 1..5;
  grupp: string[3]
end;
var student: anketa;
begin
  writeln ('введите сведения о студенте');
  {обратите внимание, ввод каждого поля осуществляется отдельно}
  writeln ('введите фамилию, имя и отчество');
  readln (student.fio);
  writeln ('введите дату рождения');
  readln (student.dat_r);

  writeln ('введите адрес');
  readln(student.adres);
  writeln ('введите курс');
  readln(student.curs);
  writeln ('введите группу');
  readln (student.grupp);
```



# Усложним задачу

Пусть нам необходимо иметь сведения о многих студентах, например, нашего факультета. Следовательно, необходимо организовать массив записей Паскаля. А затем из общего списка вывести фамилии студентов 2-го курса.

```
program primer 2 ;
type anketa=record
  fio: string[45];
  dat_r: string[8];
  adres: string[50];
  curs: 1..5;
  grupp: string[3]
end;
var student: array [1..100] of anketa;
  l: integer;
begin
  {последовательно вводим каждую запись}
  for l:=1 to 100 do
  begin
    writeln ('введите сведения о', l, '-м студенте');
    writeln ('введите фамилию, имя и отчество');
    readln (student[l].fio);
    writeln ('введите дату рождения');
    readln (student[l].dat_r);
    writeln ('введите адрес');
    readln(student[l].adres);
    writeln ('введите курс');
    readln(student[l].curs);
```

# Оператор присоединения или как избавиться от префикса

**Префикс** – обязательная предшествующая часть составного идентификатора для имен полей в структуре типа запись Паскаля. Очень часто у программиста возникает желание не указывать префикс в имени полей, например, когда идет постоянное использование одних и тех же записей. В языке Паскаль предусмотрена такая возможность, реализуемая при помощи оператора присоединения, который в общем виде выглядит так:

**with** <имя\_записи> **do** <действие с полем записи>;

Следует обратить внимание на то, что после служебного слова **do** может стоять только один оператор, но он может быть составным (любая последовательность операторов, заключенная в операторные скобки **begin end**).

# Пример фрагмента программы с записью и префиксом Паскаля

```
for l:=1 to 100 do
  with student[l] do
  begin
    writeln ('введите сведения о', l, '-м студенте');
    writeln ('введите фамилию, имя и отчество');
    readln (fio);
    writeln ('введите дату рождения');
    readln (dat_r);
    writeln ('введите адрес');
    readln(adres);
    writeln ('введите курс');
    readln(curs);
    writeln ('введите группу');
    readln (grupp);
  end;
```

# Источники:

- 0 <http://pas1.ru/recordtask>
- 0 [http://life-prog.ru/view\\_zam.php?id=30](http://life-prog.ru/view_zam.php?id=30)
- 0 <http://www.cyberforum.ru/post5537228.html>
- 0 <http://www.cyberforum.ru/pascalabc/thread60683.html>
- 0 <http://www.programmersforum.ru/index.php?s=c092477c928639cd83c4384dc65a70ca>
- 0 <http://pascalabc.net>
- 0 <http://www.tehnari.ru/f41/t54258/>