

# Работа с данными типа запись

**Лабораторная работа**

# Цель работы

Решение задач на ЯП Паскаль по теме «Комбинированный тип данных (Запись)»

## Ход занятия:

- Повторение теоретического материала по теме «Записи в Паскаль»
- Решение Задачи1
- Решение Задачи2
- Самостоятельная работа. Задача3

# Оценка лабораторной работы

Каждый этап занятия оценивается в баллах. По набранному количеству баллов выставляется оценка.

- Работа на уроке 1 балл
- Задача1 1 балл
- Задача2 2 балла
- Задача3 3 балла
- Отчет 1 балл
- 8-7 баллов «5»
- 6-5 баллов «4»
- 4-3 балла «3»

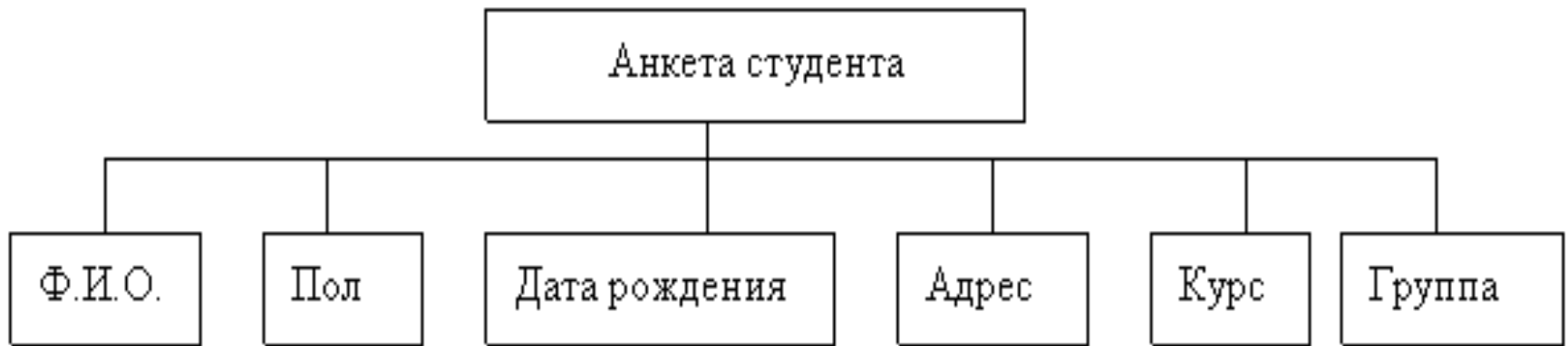
# Записи (Record)

**Дайте определение типу данных Запись**

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

# Записи (Record)

Например, **анкетные данные о студенте** могут быть представлены в виде информационной структуры



Такая структура называется **двухуровневым** деревом. В Паскале эта информация может храниться в одной переменной типа **record** (запись).

# Записи (Record)

Задается тип **record** следующим образом:

```
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> - идентификаторы языка  
Паскаль.

# Записи (Record)

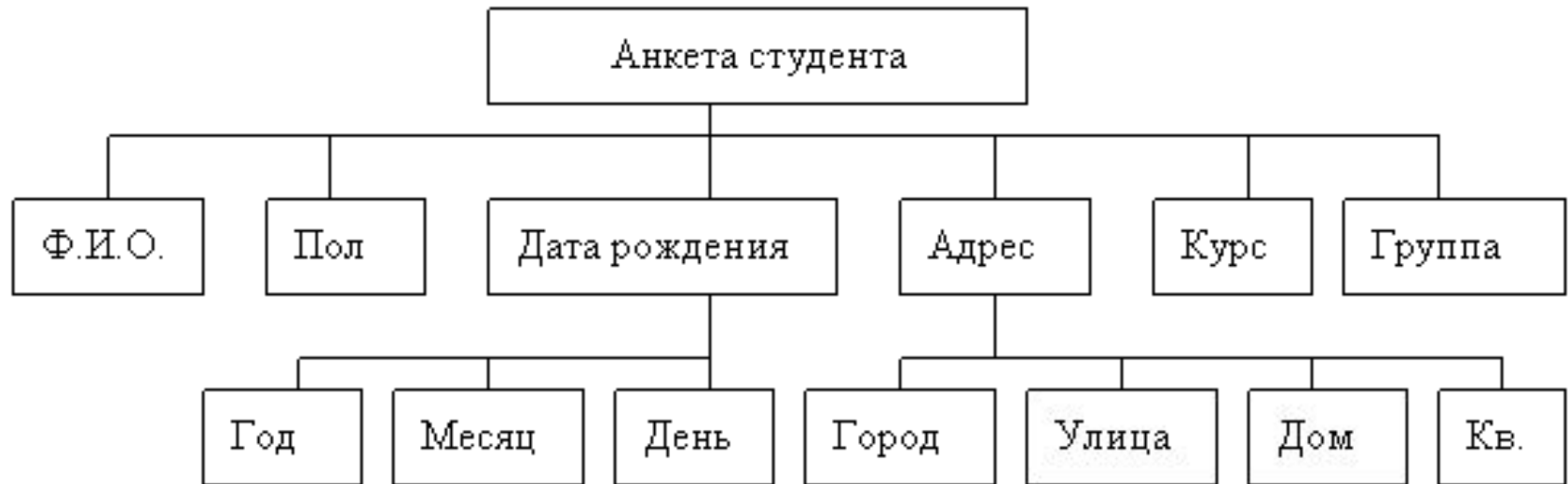
Описание анкеты студента на ЯП Паскаль будет выглядеть так:

```
Type anketa=record
  fio: string[45];
  pol: char;
  dat_r: string[8];
  adres: string[50];
  curs: 1..4;
  grupp: string[4];
end;
```

Такая запись на ЯП Паскаль, так же как и соответствующее ей дерево, называется двухуровневой.

# Записи (Record)

Поля записи Паскаля могут иметь любой тип, в частности сами могут быть **записями**. Такая возможность используется в том случае, когда требуется представить многоуровневое дерево (**более 2 уровней**). Например, те же сведения о студентах можно отобразить **трехуровневым** деревом.



Такая организация данных позволит, например, делать выборки **по году рождения или по городу**, где живут студенты.



# Записи (Record)

В этом случае описание соответствующей записи на ЯП

Паскаль будет выглядеть так:

```
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, kv: integer;
      end;
    curs: 1..4;
  grupp: string [4];
end;
```

# Записи (Record)

## Объявление переменных типа Запись:

var

student: anketa;

student 1: anketa 1;

## Обращение к элементам записи

<имя\_записи>.<имя\_поля>

# Записи (Record)

## Задача 1

Заполнить сведения о студентах (Ф.И.О., дата рождения, адрес, курс и группа), а затем вывести эти сведения на экран.

```
program zadacha1;  
type anketa=record  
    fio: string[45];  
    dat_r: string[8];  
    adres: string[50];  
    curs: 1..4;  
    grupp: string[4]  
end;  
var student: array[1..25] of  
    anketa;  
i,n:integer;
```

# Записи (Record)

```
begin
writeln ('введите количество студентов в группе');
readln (n);
for i:=1 to n do begin
{обратите внимание, ввод каждого поля осуществляется отдельно}
  writeln ('введите сведения о студенте');
  writeln ('введите фамилию, имя и отчество');
  readln (student[i].fio);
  writeln ('введите дату рождения');
  readln (student[i].dat_r);
  writeln ('введите адрес');
  readln (student[i].adres);
  writeln ('введите курс');
  readln (student[i].curs);
  writeln ('введите группу');
  readln (student[i].grupp);
  writeln ('ввод закончен');
  writeln;
end;
```

# Записи (Record)

```
for i:=1 to n do begin
```

```
{обратите внимание, что вывод записи осуществляется по полям}
```

```
  writeln ('фамилия студента: ', student[i].fio );
```

```
  writeln (' дата рождения : ', student [i].dat_r);
```

```
  writeln (' адрес : ', student [i].adres);
```

```
  writeln (' курс : ', student [i].curs);
```

```
  writeln (' группа : ', student [i].grupp);
```

```
end;
```

```
end.
```

# Записи (Record)

## 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** ).

# Записи (Record)

Например, фрагмент программы с использованием оператора присоединения будет выглядеть так:

Пример фрагмента программы с записью и префиксом ЯП

Паскаль

```
for l:=1 to n do
  with student[i] do
  begin
    writeln ('введите сведения о', i , '-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;
```

# Записи (Record)

**Задача 2.** Заполнить данные об автомобиле (марка, модель, год выпуска). Вывести на экран информацию об автомобилях выпущенные после года введенного пользователем.

Фильтрация данных

```
For i:=1 to n do  
  If mas_avto[i].god_v > user_god then  
    writeln(mas_avto[i].marka, ' ', mas_avto[i].model, ' ',  
    mas_avto[i].god_v);
```



# Записи (Record)

**Задача 3.** Заполнить данные о книгах в библиотеке (не менее пяти книг) в соответствии со схемой. Вывести на экран информацию:

1. Книги не старше пяти лет.
2. Книги определённого автора, отсортировав список по названию.

КНИГА

АВТОР

НАЗВАНИЕ

ИЗДАТЕЛЬСТВО

ГОД ВЫПУСКА

КОЛИЧЕСТВО  
ЭКЗЕМПЛЯРОВ

# Оформление отчета

Баллы за отчет начисляются если он оформлен правильно.

Отчет должен содержать ФИО студента, формулировку задачи, код программы (не скрин), скрин результата (видны входные и выходные данные)