


Записи

в языке Pascal

Запись – это структурированный тип данных, состоящий из фиксированного числа компонентов одного или нескольких типов, называемых полями. Идентификатор (имя) поля должен быть уникален только в пределах записи.

Обращение к значению поля осуществляется с помощью **составного имени** - идентификатора переменной (имени записи) и идентификатора поля, разделенных точкой.

ИМЯ ЗАПИСИ . ИМЯ ПОЛЯ



Номер	Марка	ФИО
М234КТ 178	Жигули	Ивашов Иван Петрович
К456МН 178	Ауди	Перова Ольга Осиповна
К456КК 178	Рено	Алексеев Олег Сергеевич

В языке Turbo Pascal различают фиксированные и варианты записи.

- **Фиксированная запись** состоит из одного или нескольких полей, для каждого из которых при объявлении указывается имя (идентификатор) и тип

Описание фиксированной записей

Type **имя типа** = **record**

имя поля 1 : тип;

имя поля 2 : тип;

...

имя поля N: тип

end;

Var **Имя записи** : **имя типа**;

где **record**, **end** – зарезервированные слова (запись, конец);

имя поля1,..имя поля n – список полей;

Пример:

Type Car = **record**

Number: string[10]; {номер}

Marka: string[20]; {марка автомобиля}

FIO: string[50] {фамилия владельца}

end;

Var a,b: Car;

Пример обращения к полям записи:

a. Marka, B.FIO, a.Number

Для упрощения доступа к полям записи используется оператор присоединения **With**

With имя записи **do**
оператор;


где **With, do** – зарезервированные слова (с, делать);
имя записи – имя переменной типа запись, за
которым возможно следует список вложенных полей;
оператор – любой (один) оператор языка Pascal .

Например: **With A do** Marka:='Жигули'

- К записям в целом применим оператор присваивания .

Например $A := B$

- Над полями записей можно выполнять любые операции, присущие данному типу.



Pascal допускает использование
вложенных записей (т.е. поле записи
может быть в свою очередь тоже записью).

Уровень вложения не должен превышать 9.

Например:

Type **zap1** = **record**

Day : byte; {день}

Month : 1..12; {месяц}

Year : word {год}

end;

zap2 = **record**

FIO: string[30]; {фамилия студента}

Gruppa: integer; {номер группы}

Birthday: **zap1** {дата рождения}

end;

Var Spisok: array [1..10] of zap2;

Пример обращения к дню рождения студента:

For i:=1 to 10 do

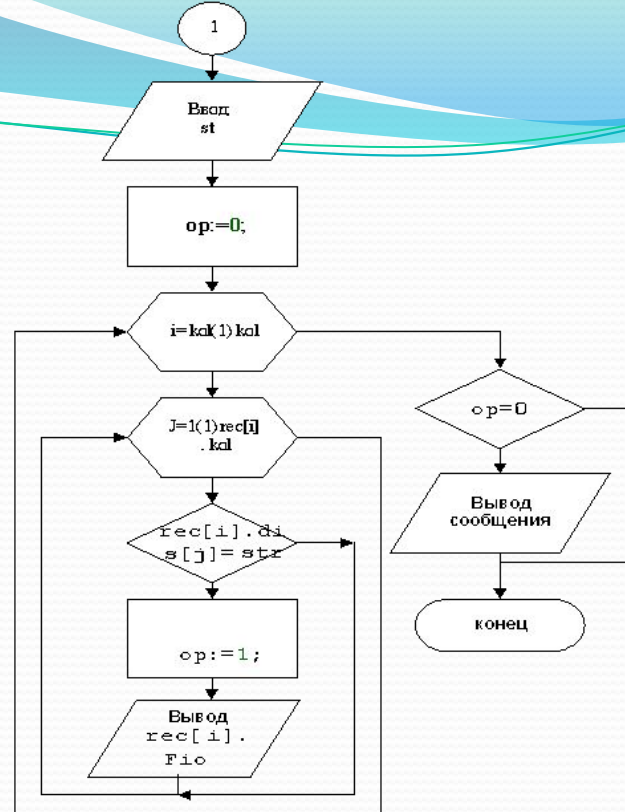
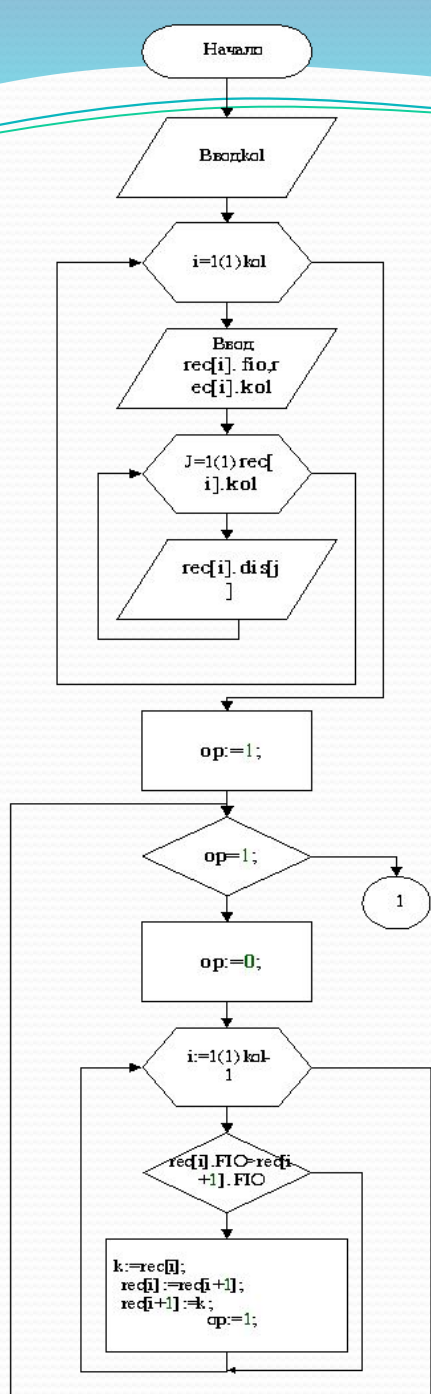
With spisok[i] , birthday do

day := 23;



Задача:

Из массива записей, содержащего информацию о преподавателях (фамилия_имя_отчество, читаемые дисциплины) вывести в алфавитном порядке фамилии преподавателей, читающих заданную дисциплину.



```
Program zapis;  
type  
Mas=array[1..10] of string;  
  TRec = Record  
    FIO: String[40]; {ФИО преподавателя}  
    KOL: integer;    {количество читаемых  
                     дисциплин}  
    DIS: Mas;        {перечень читаемых  
                     дисциплин}  
  end;  
  zap=array[1..15] of Trec;  
var rec:zap;  
k:trec;  
op,kol,i,j,n:integer;  
st:string;
```

begin

writeln('Введите количество преподавателей ');
readln(kol);

For i:=1 **to** kol **do**

begin

Write('Введите ФИО -');

readln(rec[i].fio);

write('Введите количество дисциплин -');

readln(rec[i].kol);

For j:=1 **to** rec[i].kol **do**


begin

Write('Введите ', j, '-ую дисциплину-');

readln(rec[i].dis[j]);

end;

end;



```
op:=1;
```

```
{сортировка массива записей в алфавитном  
порядке фамилий}
```

```
while op=1 do {op=1 - сортировка закончена}
```

```
begin
```

```
op:=0;
```

```
for i:=1 to kol-1 do
```

```
if rec[i].FIO>rec[i+1].FIO then
```

```
begin
```

```
    k:=rec[i];
```

```
    rec[i]:=rec[i+1];
```

```
    rec[i+1]:=k;
```

```
    op:=1;
```

```
    end;
```


```
end;
```



```
Writeln('Введите название дисциплины для  
поиска преподавателя');  
readln(st);  
op:=0; {op- признак наличия  
преподавателей, читающих заданную  
дисциплину}  
For i:=1 to kol do  
For j:=1 to rec[i].kol do  
if rec[i].dis[j]=st then begin  
    op:=1;  
    writeln(rec[i].Fio);  
end;  
if op=0 then  
    writeln ('нет преподавателей, читающих  
дисциплину -', st);  
end.
```

В языке Pascal имеется возможность задать тип записи, содержащий произвольное число вариантов структуры. Такие записи называются ***записями с вариантами***. Записи с вариантами обеспечивают средства объединения записей, которые похожи, но не идентичны. Они состоят из фиксированной и вариантной частей.

Вариантная часть формируется с помощью оператора **Case**. Он задает *поле признака*, которое определяет, какой из вариантов в данный момент будет активизирован. Значением признака в каждый текущий момент выполнения программы должна быть одна из расположенных далее констант. Константа, служащая признаком, задает вариант записи и называется *константой выбора*.



Количество полей каждого из вариантов неограниченно. **Объем памяти**, необходимый для записи с вариантами, складывается из объемов полей фиксированной части и максимального по объему поля переменной (вариантной) части.

Запись может иметь **только одну** вариантную часть и она должна размещаться в конце записи.

Описание вариантной записи

Type имя типа = record

имя поля 1 : тип;

...

имя поля N: тип;

Case поле признака : тип **of**

константа выбора 1 : (поле, ... : тип);

...

константа выбора n : (поле, ... : тип)

end;

Var Имя записи : имя типа;

Например:

```
Type Tmark1=record           {экзамены 1 семестра}
  Matem1:byte;               {математика}
  LinAlg:byte;               {линейная алгебра}
  Programir: byte;          {программирование}
end;
```

```
Tmark2=record               {экзамены 2 семестра}
  Matem2: byte;             {математика}
  Electron: byte;          {электроника}
  Fisika: byte;             {физика}
  DigAutom: byte;          {цифровые автоматы}
end;
```

```
Tstudent= record      {информация о студенте}
FIO:string[40];        {фамилия,имя,отчество}
Gruppa:string[4];      {группа}
Year: integer;         {год рождения}
{вариантная часть}
Case semester: byte of      {выбор семестра}
1: (mark1: Tmark1);
2: (mark2: Tmark2)
    end;
```

```
Var  gr921: array[1..15] of Tstudent;
```

```
Program Katalog; {каталог лит-ры в библ}
Type Publ=(Book, Journal, NewPaper);
Liter=record
    title: string[50];    {название}
    autor: string[50];    {автор}
Case V: Publ of {начало вариантной части}
    Book: (YearB: integer);
    Journal: (Num: 1..12;
                YearJ: 1900..2012);
    NewPaper: (Day:1..31;
                Month: 1..12;
                YearN: integer);
end;
```



```
Const count=10;
```

```
Var Katalog: array[1..count] of liter;  
    i: 1..count;
```

```
    Yes_liter:boolean; {поиск заданной лит-ры}
```

```
    Vibor: byte; {выбор вида издания}
```

```
    E: Puble; {для задания искомого типа}
```

```
//процедура ввода данных
```

```
Procedure Input_data;
```

```
Writeln('Введите кол-во литературы', i);
```

```
Writeln('введите число, указывающее  
номер издания:');
```

```
Readln(Vibor); {задание полю признака конкретного  
значения}
```

Case Vibor of

1: Katalog[i].v:=Book;

2: Katalog[i].v:=Journal;

3: Katalog[i].v:=NewPaper;

End;

With Katalog[i] **do**

Begin

Write('фамилия автора');

Readln (autor);

Write('название книги');

Readln(title);

Case V of {начало ввода данных в вариант. часть записи}

```
Book: begin Write('год издания');  
        Readln(YearB);  
        end;
```

```
Journal: begin Write('номер');  
        Readln(Num);  
        Write('год издания');  
        Readln(YearJ);  
        end;
```

```
NewPaper: begin Write('день');  
        Readln(Day);  
        Write('месяц');  
        Readln(Month);  
        Write('год издания');  
        Readln(YearN);  
        end;
```

```
End  
End;
```

```
//процедура вывода данных
Procedure Vivod_data;
Begin Write('литература №');
    With Katalog[i] do
        begin Write('название', title);
            Write('фамилия автора', autor);
            Case V of {начало вывода на экран
вариантной части}
Book: Write('год издания', YearB);

Journal: begin Write('номер', Num );
            Write('год издания', YearJ);
            end;
NewPaper: begin Write('день', Day );
            Write('месяц', Month);
            Write('год издания', YearN );
            end;
            end;
        end;
End;
```

// поиск литературы по типу издания

Procedure Find_liter;

Write('введите число, указывающее тип
издания');

Readln(Vibor);

Case Vibor **of**

1: E:=Book;

2: E:=Journal;

3: E:= Newspaper

end;

Yes_liter:=false; {не найден ни один экземпляр
издания}

Kol:=0;

```
For i:=1 to count do  
If Katalog[i].V=E then  
    begin  
        kol:=kol+1;  
        Vivod_data; {ВЫВОД}  
        Yes_liter:=true  
    end;  
If Yes_liter=false then Write('в  
    библиотеке нет такой литературы')  
Else Write('всего в библиотеке таких  
    изданий', kol);  
end;
```



```
//ГОЛОВНАЯ
```

```
begin
```

```
For i:=1 to count do
```

```
Input_data;      {ВВОД ДАННЫХ}
```

```
Find_data;       {ПОИС ЛИТЕРАТУРЫ}
```

```
end.
```