



# Алгоритмы и алгоритмические языки

## *Лекция 15*

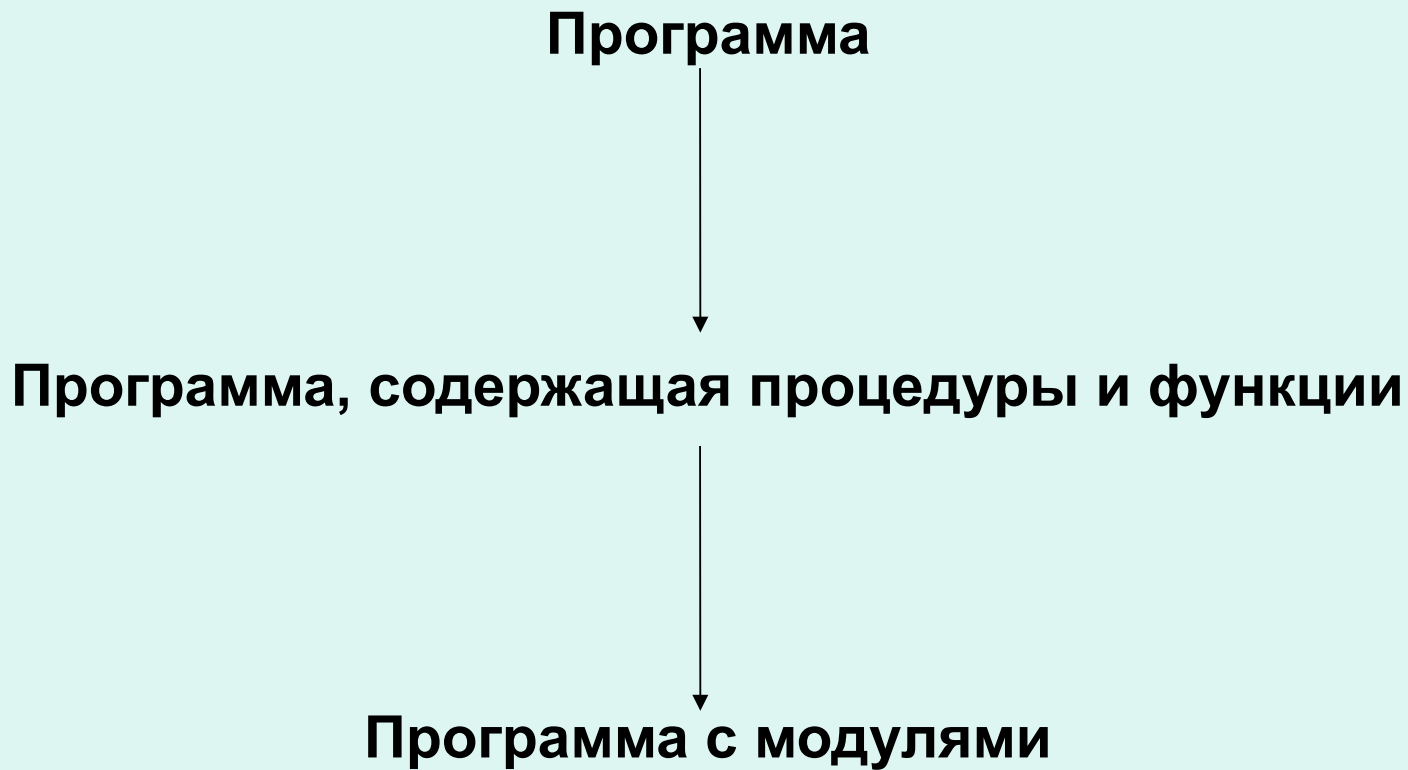
Модули (Турбо Паскаль).

Язык Паскаль: комбинированные типы,  
множества



---

# Модули (Турбо Паскаль)





# Структура модуля

**unit** <имя модуля> ;

**interface**

<экспортируемые описания>

**implementation**

<

описания экспортируемых процедур и функций,  
описание вспомогательных типов, констант, переменных  
процедур

>

**end.**

*Имя файла, в котором хранится модуль должно совпадать с  
именем модуля (указанным, после unit)*



# Пример описания модуля

**unit** Mod1;

*{работа с текстом, который требуется хранить в массиве}*

**interface**

**const** space = ' '; textSize=256;

**type** text = packed array [1..textSize] of char;

**procedure** textInput; *{считывание текста в массив}*

*..... {заголовки остальных функций}*

**implementation**

var t:text;

**procedure** textInput;

  var i:integer;

**begin**  i:=1; read(t[i]);

    writeln('Please, input text: ');

**while** (i<textSize) **and** (t[i]<>'.') **begin** i:=i+1; read(t[i]) **end**;

**if** (t[i]<>'.') **then** writeln('The text is too long');

**end**;

*..... {реализация остальных функций}*

**end.**



---

# Структура Паскаль-программы с модулями

**program** <имя программы>;

**uses** <имена модулей через запятую>

<описания меток, констант, переменных, процедур и функций>

**begin**

<операторы программы>

**end.**

# Язык Паскаль. Типы данных





---

# Язык Паскаль. Комбинированные типы.

Объекты комбинированных типов называются *записями*.

## Примеры:

<i>today</i>	<i>студ92</i>
day: 23	ФИО: Печкин Игорь Иванович
month: oct	пол: муж
year: 2012	группа: 109
	оценки: 4, 5, 5, 4

**Запись** — составной объект из фиксированного числа элементов (полей), в котором:

- 1) типы полей могут быть различными (не обязательно);
- 2) каждое поле имеет индивидуальное имя;
- 3) доступ к полям осуществляется по их именам:

today.day  
today.month  
today.year



# Язык Паскаль. Комбинированные типы.

*today*

```
day: 23
```

```
month: oct
```

```
year: 2012
```

```
type date = record
```

```
    day: 1..31;
```

```
    month: (jan, feb, mar, apr, may, jun, jul, aug, sep, oct, dec);
```

```
    year: integer
```

```
end;
```

Поля записей могут иметь любой тип, в том числе и сложный.





# Язык Паскаль. Комбинированные типы.

*студ92*

ФИО: Печкин Игорь Иванович

пол: муж

группа: 109

оценки: 4, 5, 5, 4

```
type строка = packed array[1..10] of char;
```

```
студент = record
```

```
  ФИО: record фамилия, имя, отчество: строка end;
```

```
  пол: (муж,жен);
```

```
  группа: 101..518;
```

```
  оценки: array[1..4] of 2..5
```

```
    end;
```

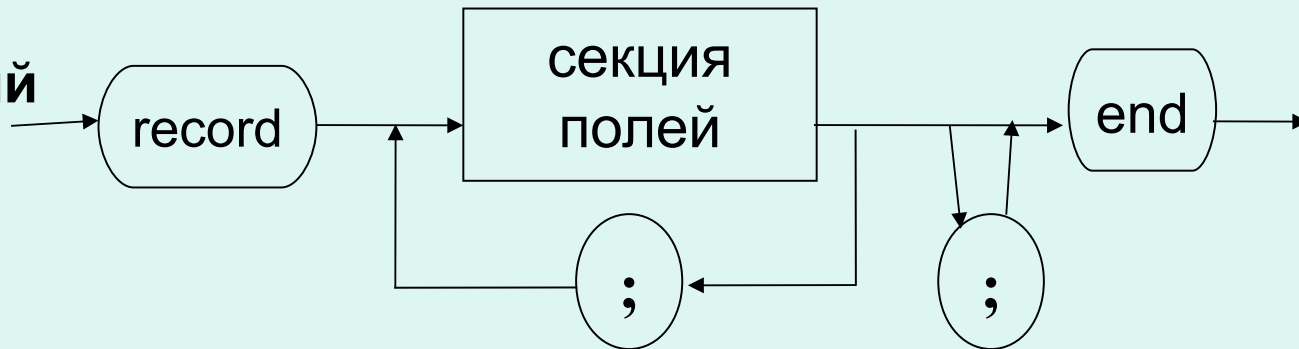
```
var студ92: студент;
```



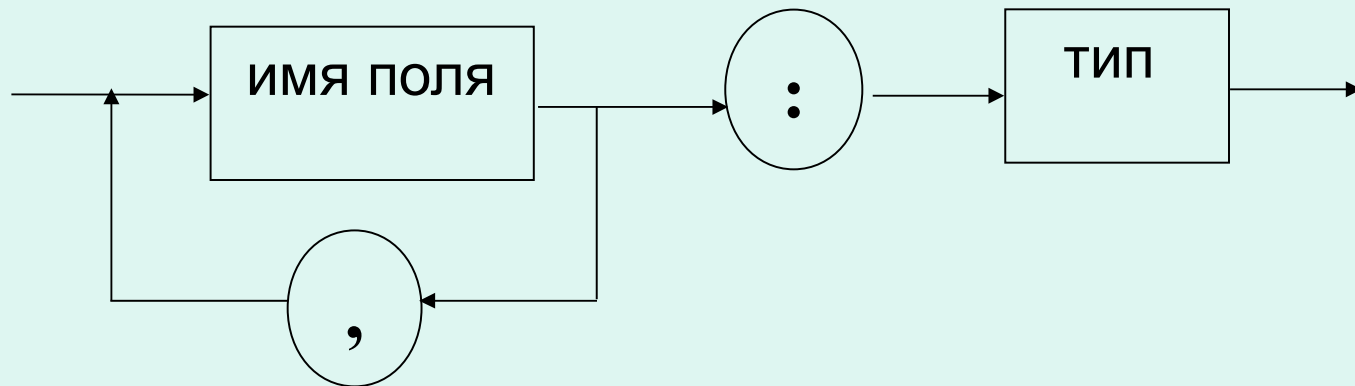
# Язык Паскаль. Комбинированные типы.

## Синтаксис

комбинированный  
тип



секция полей





# Действия с записями и их полями

```
type date = record
    day: 1..31;
    month: (jan, feb, mar, apr, may, jun, jul, aug, sep, oct, dec);
    year: integer
end;
```

```
var today, birthday: date;
```

```
    birthday := today;
```

```
today.day := 23;
today.month := oct;
today.year := 2012
```

# Работа с записями. Пример

```
function RightDate (var d:дата):boolean;  
    var last:28..31;  
begin {определение последнего числа месяца}  
    case d.month of  
jan, mar, may, jul, aug, oct, dec: last:=31;  
apr, jun, sep, nov: last:=30;  
feb: if (d.year mod 400=0) or (d.year mod 4=0) and  
    (d.year mod 100<>0)  
        then last:=29  
        else last:=28  
    end;  
    RightDate:=(d.year<>0) and (d.day<=last)  
end;
```



## Работа с записями. Пример

```
function RightDate (var d:дата) :boolean;  
    ...
```

### Использование функции RightDate:

```
var MozartBirthday:data;
```

```
MozartBirthday.day := 27;
```

```
MozartBirthday.month := jan;
```

```
MozartBirthday.year := 1756;
```

```
RightDate (MozartBirthday) ;
```



---

## Работа с записями. Пример

```
function RightDate (var d:дата) :boolean;  
    ...
```

### Использование функции RightDate:

```
var MozartBirthday:data;
```

```
MozartBirthday.day := 27;
```

```
MozartBirthday.month := jan;
```

```
MozartBirthday.year := 1756;
```

```
RightDate (MozartBirthday) ;
```

```
with MozartBirthday do
```

```
    begin day:=27;
```

```
        month:=jan;
```

```
        year:= 1756
```

```
    end;
```

# Оператор присоединения

**with** R1, R2, ..., Rn **do** S

ЭКВИВАЛЕНТНО

**with** R1 **do**

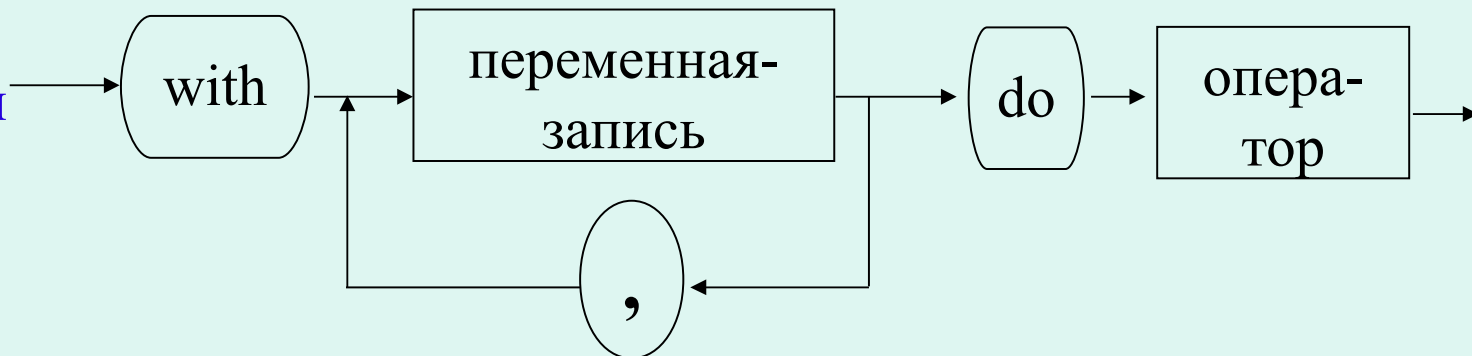
**with** R2 **do**

  ...

**with** Rn **do**

  S

оператор  
присоединения



## Пример 1

```
with today, yesterday do  
  year := 1980;
```

## Пример 2

```
var year : integer;  
with today do year:=1970;
```



# Язык Паскаль. Множественный тип

Объектами множественного типа являются **множества**

В Паскале

- множества конечные
- все элементы множества - одного типа

Тип элементов — базовый тип множества (любой простой тип, кроме real)

## Конструктор множества

$[a_1, a_2, \dots, a_n]$   $n \geq 0$

$a_i$  — элемент базового типа

или диапазон значений базового типа  $h..k$



# Описание множественного типа

<множественный тип> ::= set of <базовый тип>

## Описание типа

```
type M = set of 1..3;
```

[ ], [1], [2], [3], [1,2], [1,3], [2,3], [1,2,3]

## Описание переменной

```
var x: set of (left, right);
```

## Действия с множествами

1. Присваивание
2. Отношения:  $A=B$ ,  $A \neq B$ ,  $A \leq B$ ,  $A \geq B$ ,  $x \in A$
3. Пересечение ( $A * B$ ), объединение ( $A + B$ ), разность ( $A - B$ )



# Выражения множественного типа

выражения, в результате вычисления которого получается множество

$$M = [2, 3, 5, 7]$$


## Примеры

$$M * [1..3]$$

$$M + [4, 5, 19]$$

$$M - [1, 3..5]$$

$$M * [3..6] + ([7, 9] - [7, 8])$$



## Пример использования множеств

Дана непустая последовательность слов из малых лат. букв, слова разделяются запятыми, за последним словом точка. Напечатать те буквы, которые встречаются в каждом из слов.

Пример:

Входная последовательность  
acza, bca, zcascbb.

Результат  
a, c  
(или c, a)

## Пример использования множеств

```
program Lets(input, output);  
var M, L: set of 'a'..'z';  
      c: char;  
begin  
  L:=['a'..'z'];  
  repeat    {цикл по словам}  
    {строим мн-во M из букв очередного слова}  
    M:=[];read(c);  
    while (c<>',' ) and (c<>'.' ) do  
      begin M:=M+[c];  
            read(c)  
      end;  
    L:=L*M;    {учет M в L}  
  until c='.';  {пока не конец последнего слова}  
  {вывод букв из L}  
  for c:= 'a' to 'z' do if c in L then write(c);  
    writeln  
end.
```