



# Алгоритмы и алгоритмические языки

## *Лекция 8*

Структурное программирование.

Перечислимые и ограниченные типы.



# Что делает программа?

```
Program p123(input,output); label 1, 2, 3, 4, 5,  
6, 7, 8, 9; var x,y,z:integer; begin  
8: read(x);if x<=0 then goto 2; 6:read(y); if y<=0 then  
goto 1; 3:if x=y then goto 7;  
9::2: writeln('Error.');goto 8; 1: writeln('Error'); goto  
6; 1: if x=y then goto 7;if x>y then x:=x-y  
else y:=y-x; goto 3;7: z:=x;  
  
end.
```



# Как сделать программу понятной

***1. Комментарии***

**2. Структурная запись программы**

**3. Структурное программирование**



# Структурное программирование

Э.Дейкстра (1930 – 2002)

Алгоритм Дейкстры

Семафор Дейкстры

Структурное программирование



# Структурное программирование

1. Программа считается наглядной, простой для понимания, только если она имеет «хорошую» структуру
2. Программа имеет хорошую структуру, если в ней операторы выполняются в том же порядке, в котором они записаны в тексте программы
3. Чтобы операторы выполнялись в том же порядке, в котором записаны, нужно использовать только такие структуры управления, которые имеют только один вход и только один выход



---

# Структуры управления «хороших» программ

## 1. Следование

`S1 ; S2`

## 2. Разветвление

`if B then S1 else S2`

## 3. Цикл

`while B do S`

Доказано, что любой алгоритм можно реализовать в виде структурированной программы (с помощью трех структур управления: следования, разветвления и цикла)

# Два случая использования goto

## 1. Переход на конец программы

```
begin
  ...{ $\alpha$ }...
  if error then goto 9;
  ...{ $\beta$ }...
9:
end
```

```
begin
  ...{ $\alpha$ }...
  if not error then
    begin
      ...{ $\beta$ }...
    end
  end
end
```

## 2. Выход из цикла с параметром

```
for d:=2 to N do
  if N mod d = 0 then goto 1;
1: write(d)
```

```
exit:=false;
d:=2;
while (d<=N) and not exit do
  if N mod d = 0 then exit:=true
  else d:=d+1;
write(d)
```

# Язык Паскаль. Структура программы

```
program <имя> (<имя файла> {,<имя файла>}); - заголовок программы  
  <раздел меток>  
  <раздел констант>  
  <раздел типов>  
  <раздел переменных>  
  <раздел процедур и функций> } раздел описаний  
                                (описания)  
begin } блок  
  <оператор>;  
  <оператор>;  
  ...  
  <оператор>  
end. } раздел операторов  
                                (тело программы)
```



# Язык Паскаль. Константы

<раздел констант> ::=

const <описание константы> {<описание константы>}

<описание константы> ::= <имя> = <константа> ;

<константа> ::=

<целое> | <вещественное> | ' <символ>' | <строка> |  
<знак><имя>

<знак> ::= | + | -

## Примеры:

```
const k=-200;      pi=3.1415926;  
      star='*';   err='An error occured!';  
      mpi=-pi;    f=false;
```

# Язык Паскаль. Константы

Пример.

{Вычисление площади правильного 20-угольника, вписанного в окружность заданного радиуса}

```
program square(input, output);
  const N=20; pi=3.1415826;
  var R, fi, S: real;
begin
  write('радиус='); read(R);
  fi:=2*pi/N;    {угол}
  S:=N*(0.5*sqr(R)*sin(fi));
  writeln('площадь=', S:15:4)
end.
```



# Язык Паскаль. Раздел типов

<раздел типов> ::= type <описание типа> {<описание типа>}

<описание типа> ::= <имя> = <тип> ;

<тип> ::= <имя> | <конструктор типа>

<конструктор типа> ::= <простой тип> | <составной тип>  
| <ссылочный тип>

## Назначение:

### 1. Синонимы существующих типов

```
type int=integer;
```

### 2. Создание новых типов

справа от = пишем конструктор типа, который указывает множество величин, входящих в новый тип.

*Каждый конструктор типа вводит новый тип, отличный от всех других типов*



# Язык Паскаль. Перечислимый тип

```
type week=(mon, tue, wed, thu, fri, sat, sun);  
      TrafficLight=(red, yellow, green);
```

*В списке имен важен порядок перечисления*

*<перечислимый тип> ::= (<имя> { , <имя> } )*

## Описание переменных

```
type direction = (left, right);  
var p,q: direction;
```

```
var p,q : (left,right); {использован безымянный тип}
```

```
var p: (left, right); {ОШИБКА}  
    q: (left, right);
```

**Функции и операции для перечислимых типов:**  
*ord, succ, pred, операции сравнения, присваивание*<sup>12</sup>



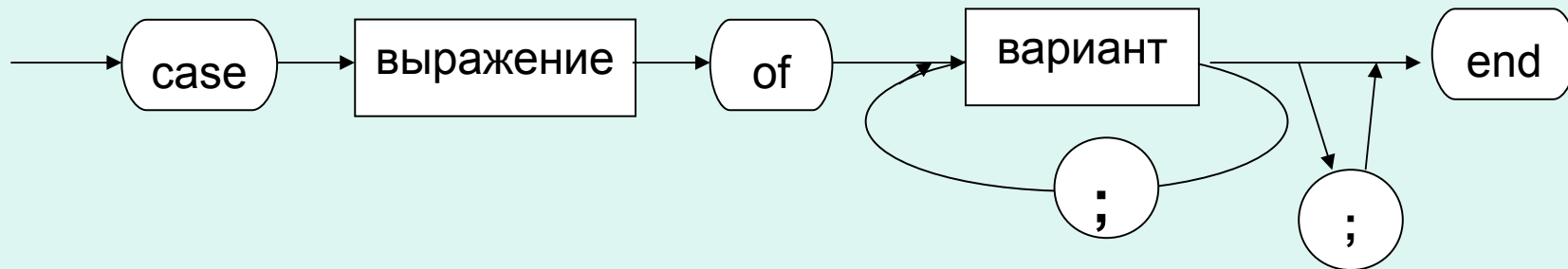
# Язык Паскаль. Оператор варианта

**type** month=(jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec);

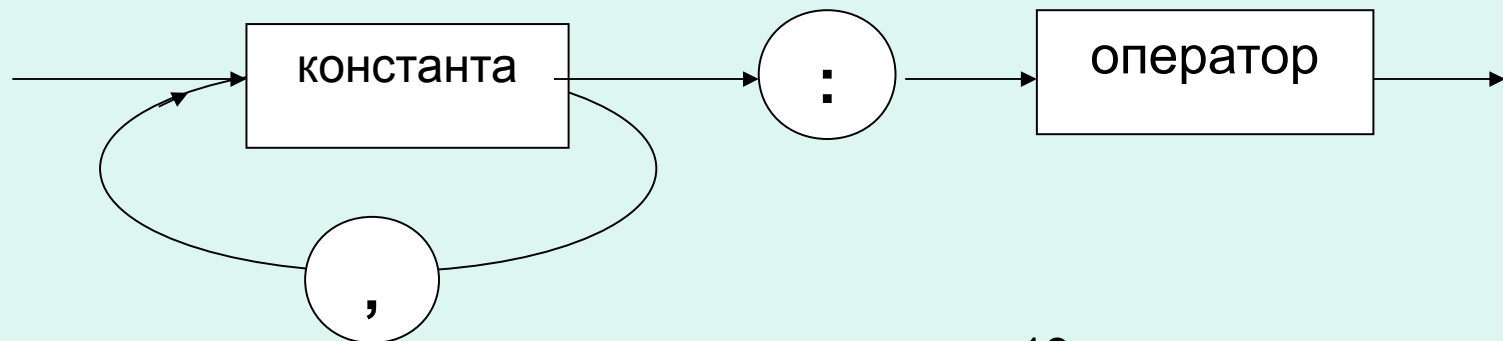
**Задача:** var k: integer; m:month;

Присвоить переменной k количество дней в месяце m

Оператор  
варианта



вариант





---

# Язык Паскаль. Оператор варианта. Пример

```
case n mod 5 of
  0: ;
  1,4: begin x:=-x; k:=1 end;
  2: k:=0
end
```

# Язык Паскаль. Ограниченные типы

```
type day=1..31;
```

```
var d: day;
```

*Переменной ограниченного типа нельзя присваивать значения, выходящие за диапазон, указанный в определении этого типа.*

**<ограниченный тип> ::= <константа>..<константа>**

Константы в диапазоне должны быть одного типа

Тип констант – любой простой, кроме real

Левая константа <= правая константа

*Пример:*

```
type digit = '0'..'9';      {char}
      natural = 1..maxint;   {integer}
      summer = jun..aug;    {month}
      bad = false..false;   {boolean}
```

```
var x: digit;
```

```
small: -9..9;
```

```
little: -9..9;
```