

Соловьев С.Ю.  
soloviev@glossary.ru

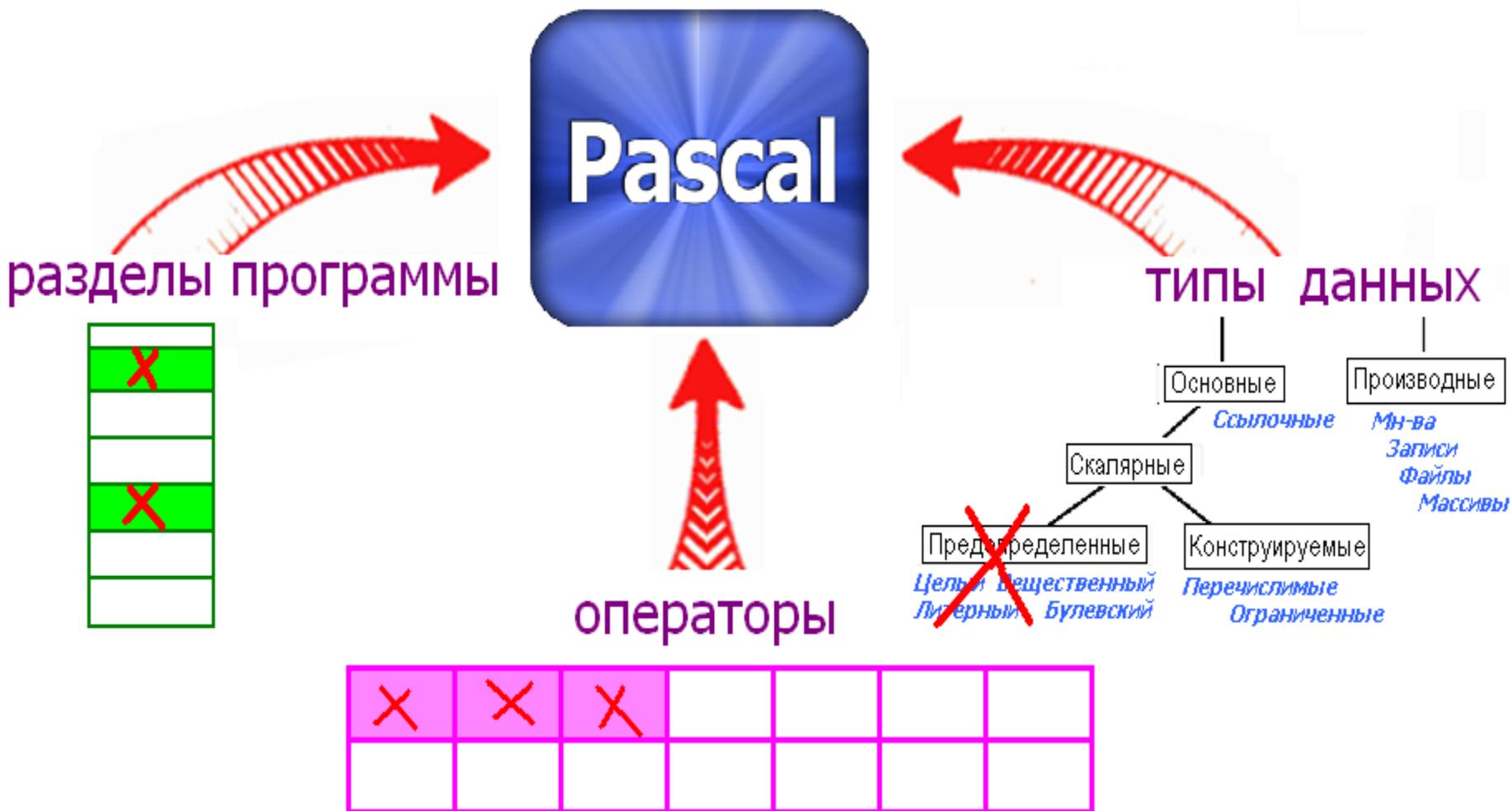
# **А**лгоритмы и **А**лгоритмические языки

[www.park.glossary.ru/pascal/](http://www.park.glossary.ru/pascal/)

*Лекция No.6*

2022

# Напоминание



# Составной оператор +

<составной оператор> ::=

begin <оператор> {; <оператор>} end

Семантика: последовательное выполнение операторов,  
если очередной оператор не изменяет посл-ть.

Прагматика: посл-ть операторов = один оператор.

```
begin I:=I+1; writeln(sin(I):8:4) end;
```

<пустой оператор> ::=

```
begin  
  I:=I+1;  
  writeln(sin(I):8:4);  
end; - три оператора
```



<раздел операторов> ::=  
<составной оператор>

Раздел  
операторов

# Условный оператор

<условный оператор> ::=

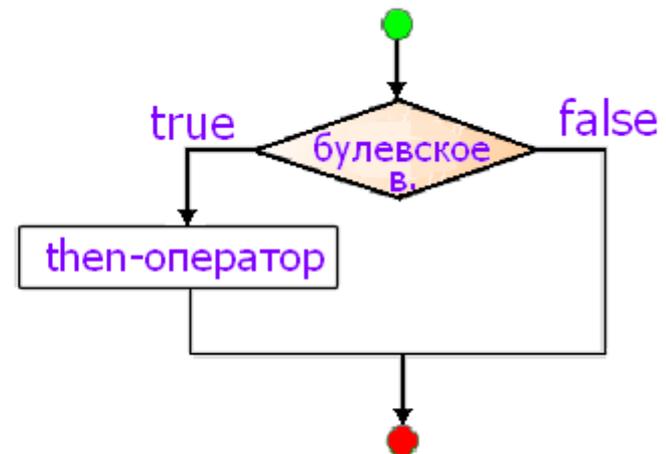
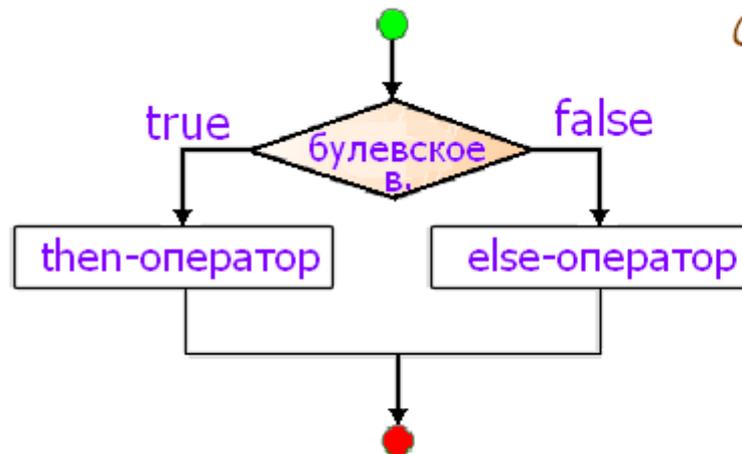
<полный условный оп-р> | <сокращенный условный оп-р>

<полный условный оп-р> ::=

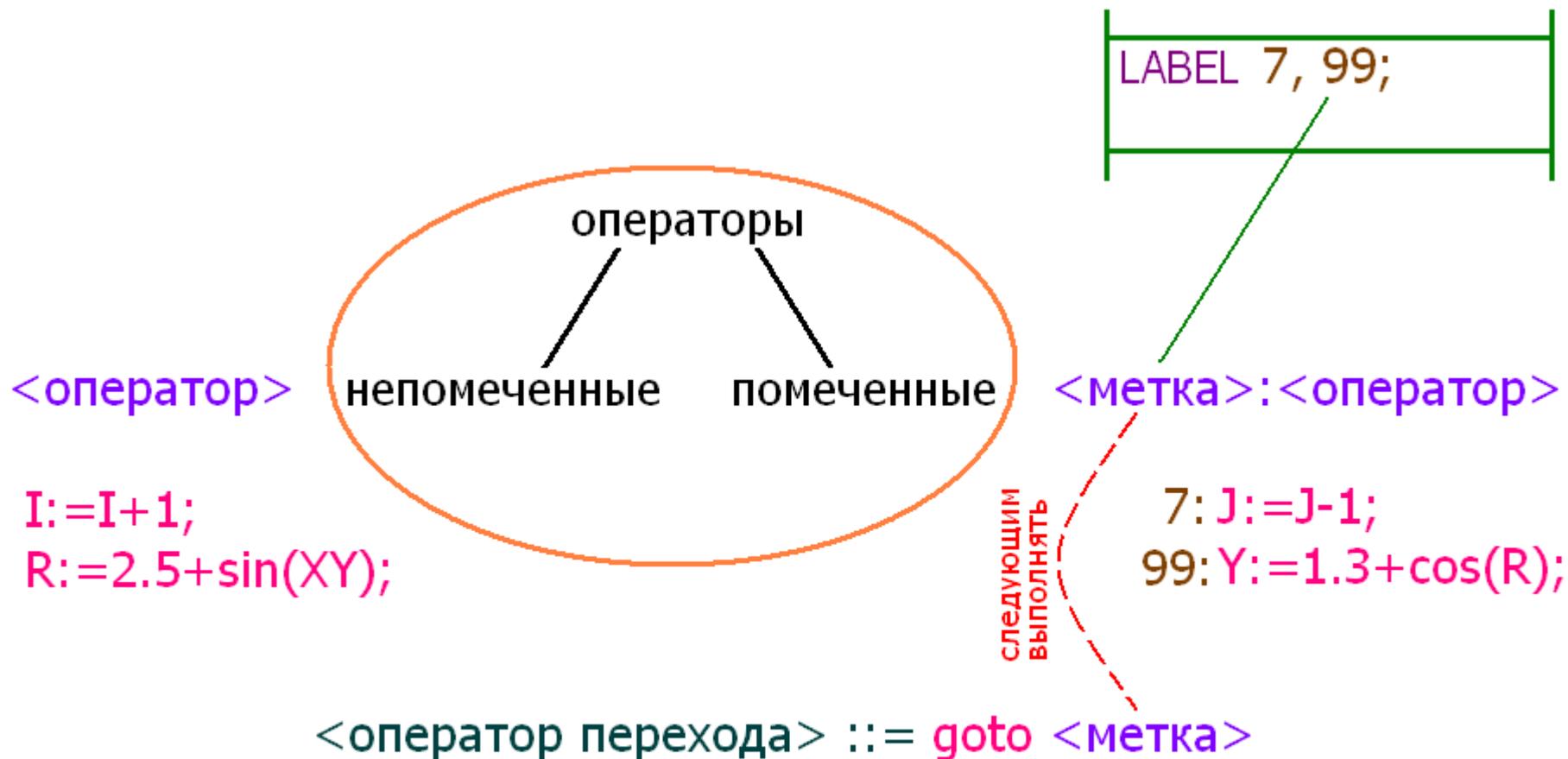
**if** <булевское выр.> **then** <оператор> **else** <оператор>

<сокращенный условный оп-р> ::=

**if** <булевское выр.> **then** <оператор>



# Оператор перехода



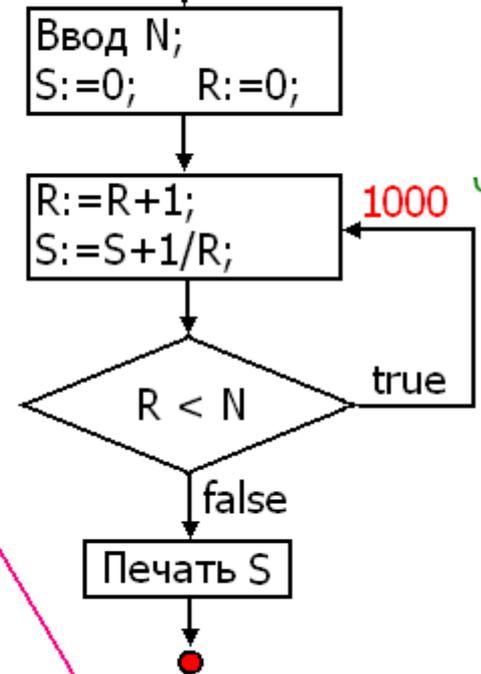
примеры: **goto 7;**  
**goto 99;**

# Пример

```
program Sep02;  
label 1000;  
var S,R : real;  
    N : integer;  
begin  write('=>'); readln(N);  
       R:=0;  
       S:=0;  
1000:  R:=R+1;  
       S:=S+1/R;  
       if round(R) < N then goto 1000;  
       writeln(S);  
end.
```

$$S = \sum_{i=1}^n \frac{1}{i}$$

идея:

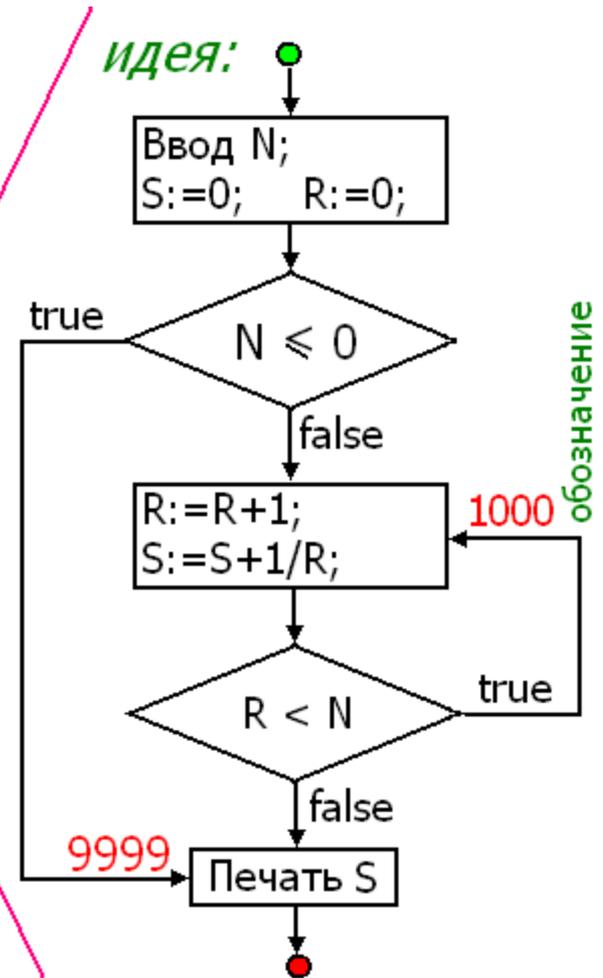


Если  $N \leq 0$  ?

# Пример/М

```
program Sep02m;  
label 1000, 9999;  
var S,R : real;  
    N : integer;  
begin write('=>'); readln(N);  
    R:=0;  
    S:=0;  
    if N <= 0 then goto 9999;  
1000: R:=R+1;  
    S:=S+1/R;  
    if round(R) < N then goto 1000;  
9999: writeln(S);  
end.
```

$$S = \sum_{i=1}^n \frac{1}{i}$$



- Два оператора перехода.
- Повторяющиеся последовательности операторов.

# Операторы цикла

*Прагматика: для программирования повторяющихся [последовательностей] операторов.*

◆ Оператор цикла с предусловием

**while B do S**

◆ Оператор цикла с постусловием

**repeat S<sub>1</sub>; ... S<sub>n</sub> until B**

◆ Оператор цикла с параметром

вариант 1. **for V:=E1 to E2 do S**

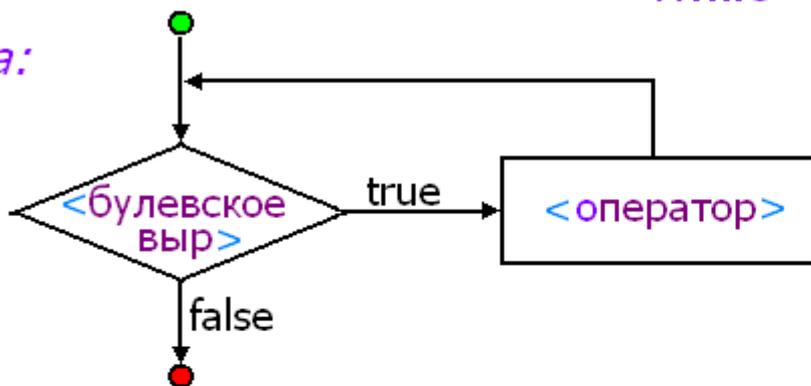
вариант 2. **for V:=E1 downto E2 do S**

# ◆ Оператор цикла с предусловием

<оператор цикла с предусловием> ::=

**while** <булевское выр> **do** <оператор>

Семантика:



Задача. Вычислить  $\min \{ k \mid k^2 > M \}$ ,  $M$  — задано.

Алгоритм.

```
K:=0;  
while sqr(K) <= M do K:=K+1;  
writeln('K=',K);
```

$M = 10$ :

K	sqr(K) <= M	
0	0 <= 10	true
1	1 <= 10	true
2	4 <= 10	true
3	9 <= 10	true
4	16 <= 10	false

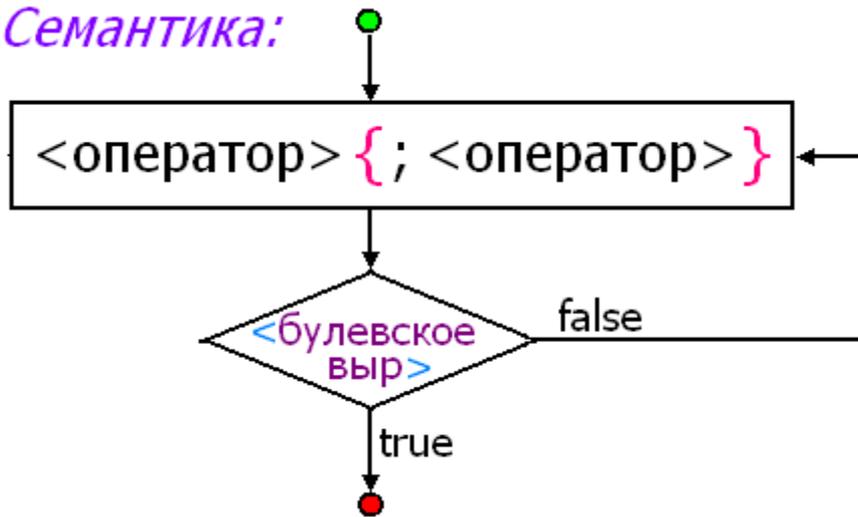
⇒ K=4

# ◆ Оператор цикла с постусловием

<оператор цикла с постусловием> ::=

repeat <оператор> { ; <оператор> } until <булевское выр.>

Семантика:



Задача. К-во подряд нажатых клавиш с нечетными значениями ord.

```
program Sep09;
  var N : integer;
      A : char;
begin
  N:=0;
  write('>');
  repeat
    read(A);
    if odd(ord(A)) then N:=N+1;
  until not odd(ord(A));
  readln;
  writeln(N);
end.
```

# ◆ Оператор цикла с параметром

<оператор цикла с параметром> ::=

for <параметр> := <выражение> to <выражение> do <оператор> |

for <параметр> := <выражение> downto <выражение> do <оператор>

## Семантика

for V:=E1 to E2 do S

```
temp1:=E1;
temp2:=E2;
if temp1 <= temp2 then begin
  V:=temp1;
  S;
  while V <> temp2 do begin
    V:=succ(V);
    S
  end
end;
```

for V:=E1 downto E2 do S

```
temp1:=E1;
temp2:=E2;
if temp1 >= temp2 then begin
  V:=temp1;
  S;
  while V <> temp2 do begin
    V:=pred(V);
    S
  end
end;
```

## ◆ Оператор цикла с параметром – 2

for V:=E1 to E2 do S

for V:=E1 downto E2 do S

*Требование.* V, E1 и E2 – должны быть одного типа  $\neq$  real

*Условие.* После выполнения оператора цикла значение V неизвестно.

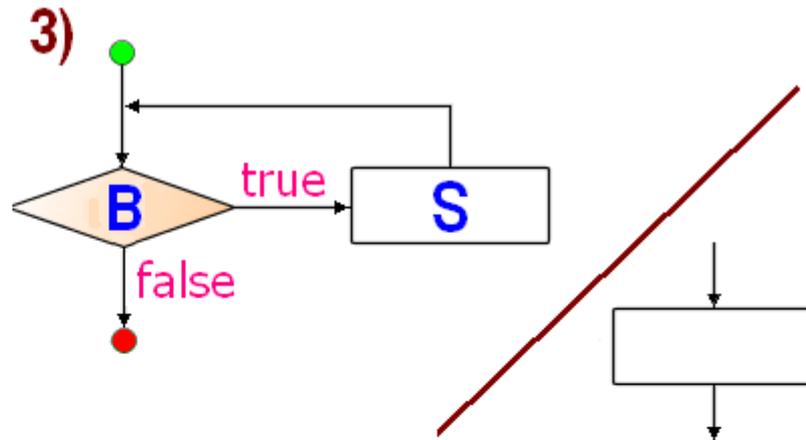
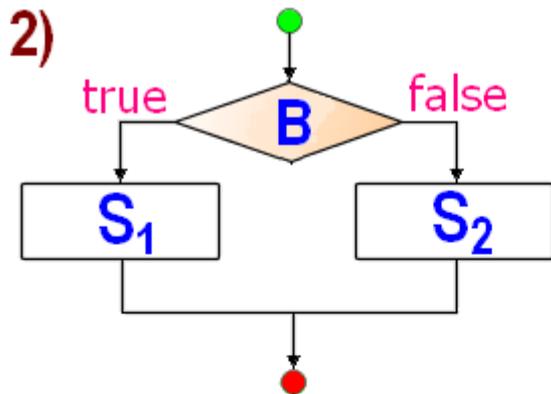
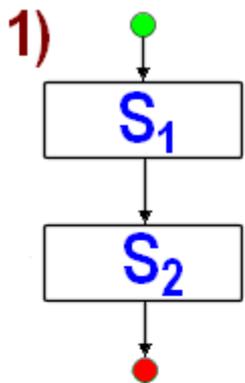
$$S = \sum_{i=1}^n \frac{1}{i}$$

*Пусть:* var I, N : integer;  
S : real;

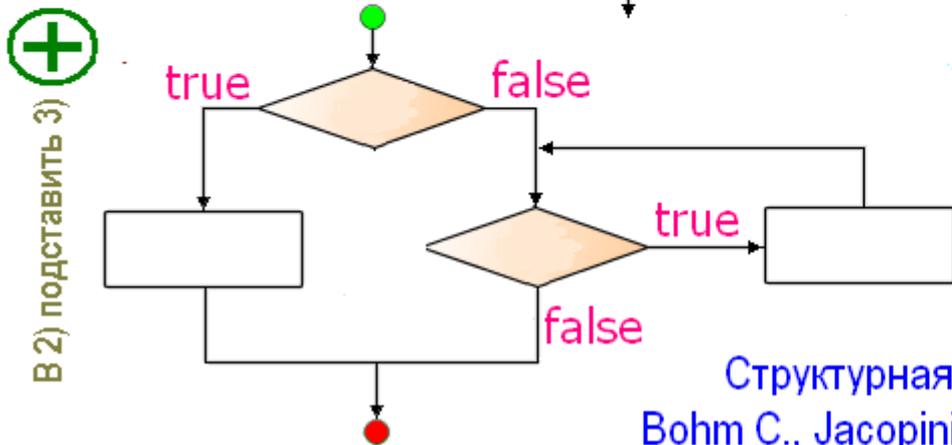
-----  
S:=0;  
for I:=1 to N do S:=S+1/I;

# Структурное программирование

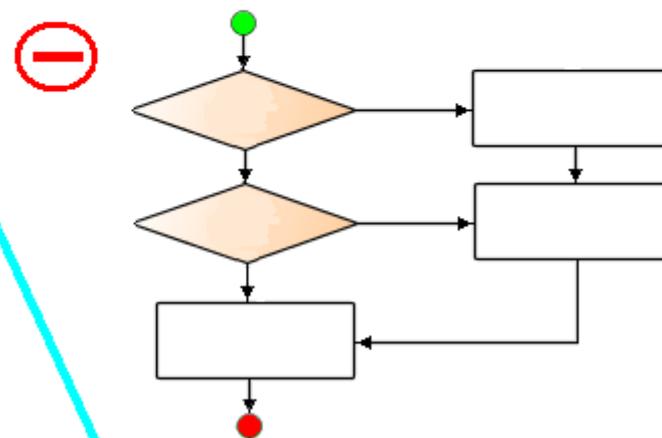
дисциплина записи программ с использованием трех конструкций с одним входом и одним выходом.



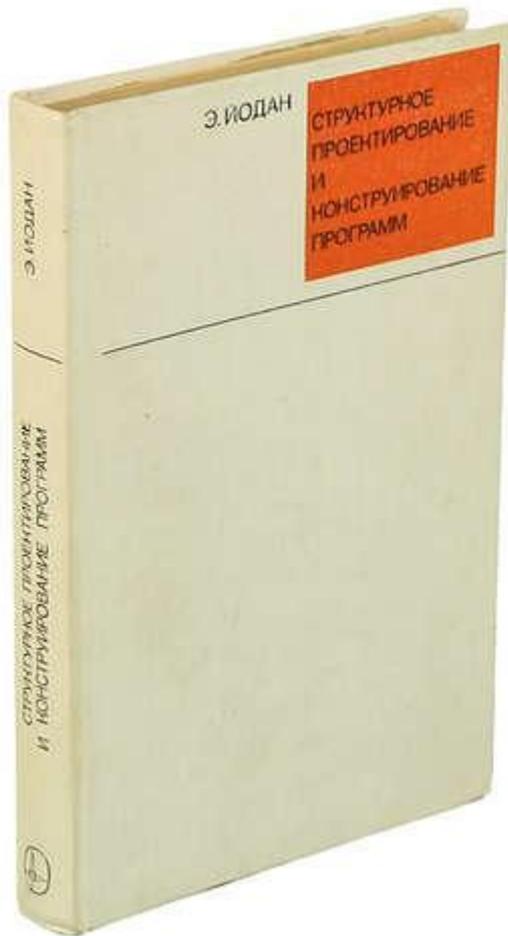
Суперпозиция: вместо  можно подставлять 1), 2) и 3)



Структурная теорема  
Bohm C., Jacopini G. (1966)

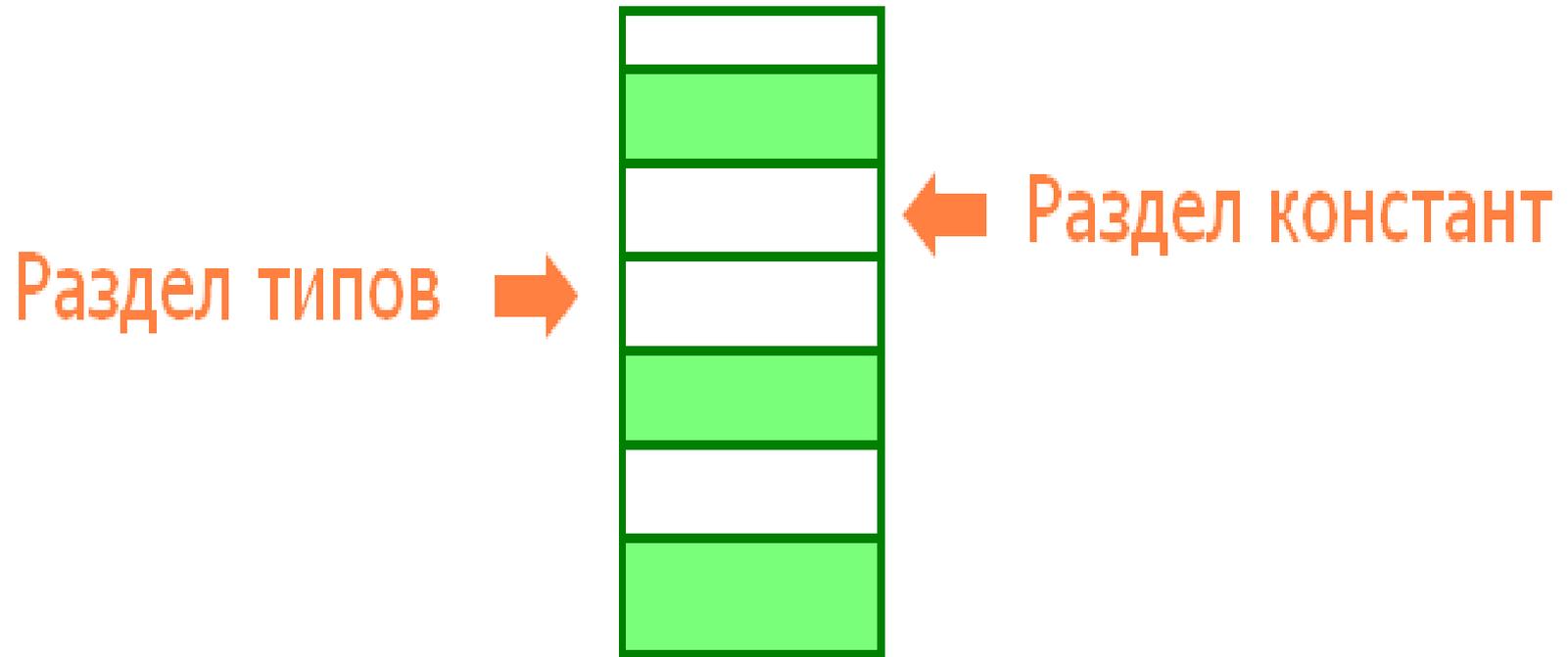


# Структурное программирование



Йодан Э.  
Структурное проектирование  
и конструирование программ.  
– М.: Мир, 1979.

# Программа



# Раздел констант (прагматика)

## Физические постоянные

- Больцмана  $1.380622 \cdot 10^{-23}$
- универсальная газовая  $8.31441$
- ускорение св.падения  $9.80665$

## Математические константы

$$\pi = 3.14159265$$

$$e = 2.718$$

$$\text{постоянная Эйлера } \gamma = 0.5772$$

Требование: **1.** Численно определять постоянные величины.

**2.** Использовать символические имена постоянных.

Константа – в языке Паскаль – конкретное значение некоторого типа, которое

- зафиксировано в программе; и
- не может изменяться в процессе вычислений.

# Программа :: Раздел констант

<раздел констант> ::= <пусто> |

**const** <описание константы>; { <описание константы>; }

<описание константы> ::=

<ИМЯ КОНСТАНТЫ> = <значение константы>

```
program Sep09;  
  const PIr = 3.14159265;  
  var R : real;  
begin  write('Радиус='); Readln(R);  
       writeln('Длина=', 2*PIr*R:9:3);  
       writeln('Площадь=', PIr*sqr(R):7:3);  
end.
```

Радиус=3

Длина= 18.850

Площадь= 28.274

# Программа :: Раздел типов

*Прагматика.* Потребность в нестандартных типах данных.

количество использований

с присваиванием имени

без имени

<описание типа> ::=

<имя типа> = <задание типа> | <имя типа> = <имя типа>

конструктор типа

именная идентичность

<раздел типов> ::=

<пусто> | type <описание типа>; {<описание типа>;}

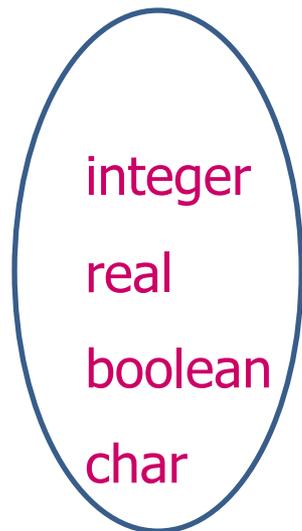
*Правило.* Каждое новое описание вводит новый тип, тип, отличающийся от других типов.

type specint = integer;

# Другими словами

Единственное число

целый тип  
вещественный тип  
булевский тип  
символьный тип

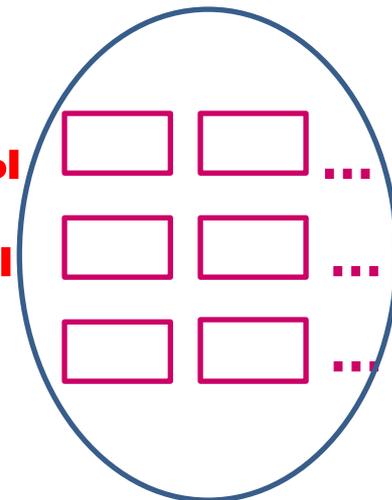


**Н.Вирт**

**vs.**

Множественное число

перечислимые типы  
ограниченные типы  
регулярные типы  
...



**Программист**



**O** <задание типа>

**H** <описание типа>

# Перечислимые типы

*Названия месяцев, дней недели и т.д.*

$\text{pred('A')}$  <задание перечислимого типа> ::= (<имя> {,<имя>})

```
type week = (pon, vto, sre, che, pia, sub, vos) ;  
нумерация:      0      1      2      3      4      5      6
```

(А) Переменные перечислимого типа

```
var Y : week ;  
var X, D : (pon, vto, sre, che, pia, sub, vos) ;
```

(Б) Значения: **pon, vto, sre, che, pia, sub, vos.**

(В) Операции над переменными/значениями перечислимого типа

- |   |                        |
|---|------------------------|
| 1. <b>ord</b> : week $\rightarrow$ { 0,1,2,3,4,5,6 } ; ord(vto) = 1 | 4. Присваивание:       |
| 2. <b>succ</b> : week $\rightarrow$ week; succ(pia) = sub           | Y:=pred(Y); X:=D;      |
| 3. <b>pred</b> : week $\rightarrow$ week; pred(sre) = vto           | 5. <, <=, >, >=, =, <> |
| 6. Переменная цикла: <b>for</b> Y:=pon <b>to</b> pia <b>do</b> S;   |                        |

# Оператор варианта (прагматика)

(-)

```
if (D = pon) or (D = sre) then writeln('лекции')
else if (D = vto) or (D = che) then writeln('семинары')
else if D = pia then writeln('заседания')
else if (D = sub) or (D = vos) then writeln('выходные');
```

## Предпочтительно:

```
case D of
  pon, sre : writeln('лекции');
  vto, che : writeln('семинары');
  pia : writeln('заседания') ;
  sub, vos : writeln('выходные');
end;
```

# Оператор варианта (синтаксис и семантика)

<оператор варианта> ::= **case** <селектор оператора> **of**  
    <элемент списка варианта>  
    { ; <элемент списка варианта> } **end**

---

<селектор оператора> ::= <выражение>

---

<элемент списка варианта> ::=  
    <список меток варианта> : <оператор>

---

<список меток варианта> ::=  
    <метка варианта> { , <метка варианта> }

---

<метка варианта> ::= <константа> того же типа, что и селектор

---

*Семантика.* 1. **Вычислить** <селектор оператора>  
2. **Найти** <элемент списка варианта>  
3. **Вычислить** <оператор>

# Оператор варианта (пример)

Дано:  $Y, M$  – месяц года; пара целых.

Вычислить  $D$  – количество дней в месяце.

```
case M of
  1, 3, 5, 7, 8, 10, 12 : D:=31;
  4, 6, 9, 11          : D:=30;
  2 : if Y mod 400 = 0 then D:=29
      else if Y mod 100 = 0 then D:=28
      else if Y mod 4 = 0 then D:=29
      else D:=28
end;
```

# Ограниченные типы

Если  $M$  – номер месяца, то  $M : \text{integer} \oplus (1 \leq M) \text{ and } (M \leq 12)$

Ограничения на базовый тип

$\text{pred('A')}$  <задание ограниченного типа> ::=  
<константа1>..**<константа2>**

```
type month = 1..12;  
digit = '0'..'9' ;  
week = (pon, vto, sre, che, pia, sub, vos) ;
```

(A) Переменные ограниченного типа

```
var M : month;  
W : pon..pia;
```

(Б) Значения: из диапазона

(В) Операции базового типа  
(в пределах диапазона)

ЕСЛИ

```
var M : month;
```

ТО

```
M:=100 ;      { - }
```

```
M:=5 ;       { + }
```

```
M:=M+2 ;    { + }
```

## Обзор лекции No.6

Составной оператор  
Пустой оператор  
Раздел типов  
Условный оператор  
Оператор перехода  
Оператор цикла с предусловием  
Оператор цикла с постусловием  
Операторы цикла с параметрами  
Структурное программирование  
Раздел констант  
Раздел типов  
Перечислимые типы данных  
Оператор варианта  
Ограниченные типы

**--- Конец лекции No. 6 ---**