

Тема 4.5 Основы программирования на языке Паскаль

1. Языки программирования высокого уровня
2. Структура программы. Описание данных
3. Простейшие операции
4. Операторы ввода-вывода операторы



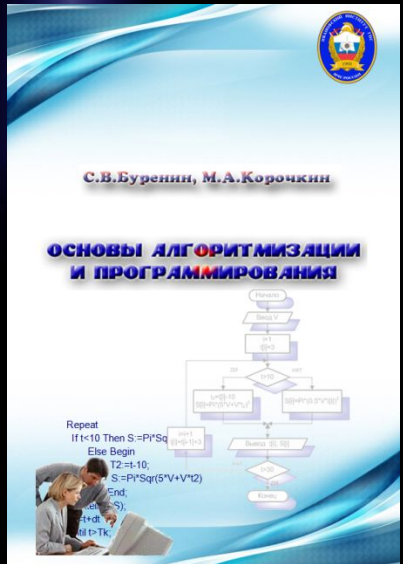


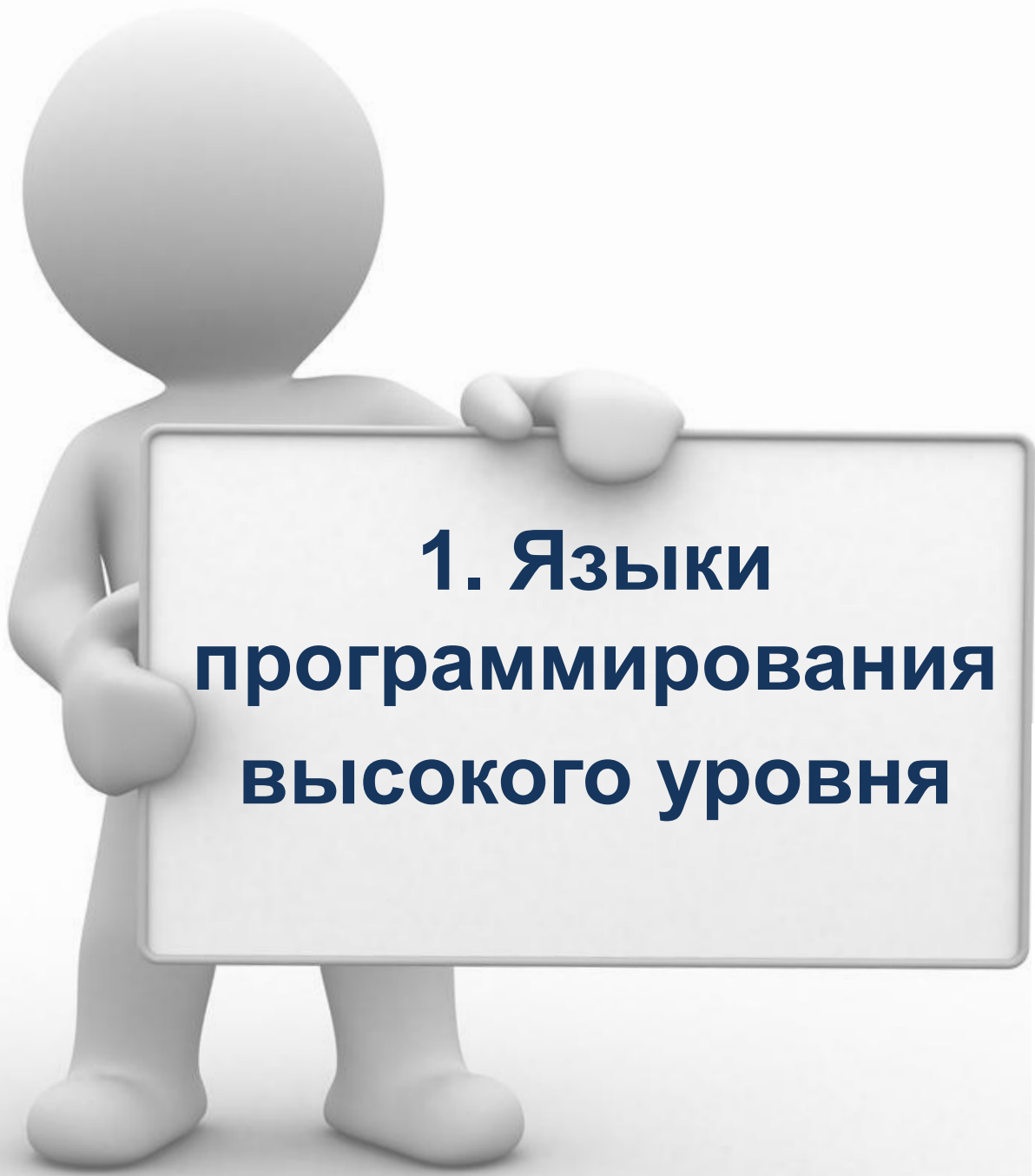
ЛИТЕРАТУРА



Буренин С.В.
Turbo Pascal. Основы
программирования:
Учебное пособие. – 2004

Буренин С.В., Корочкин М.А.
Основы алгоритмизации и
программирования:
учебное пособие. – 2011





**1. Языки
программирования
высокого уровня**



Visual Basic, Simula, Visual C++, HTML, Oberon, APC, Fortran, Basic, Algol, Eiffel, Cobol, PL/I, awk, Lisp, ADA, Pascal, Icon, sed, Prolog, C, Modula-2, Smalltalk, perl, Scheme, Delphi, JAVA

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ



Системное

**операционные системы,
драйверы, утилиты,
сервисные программы**

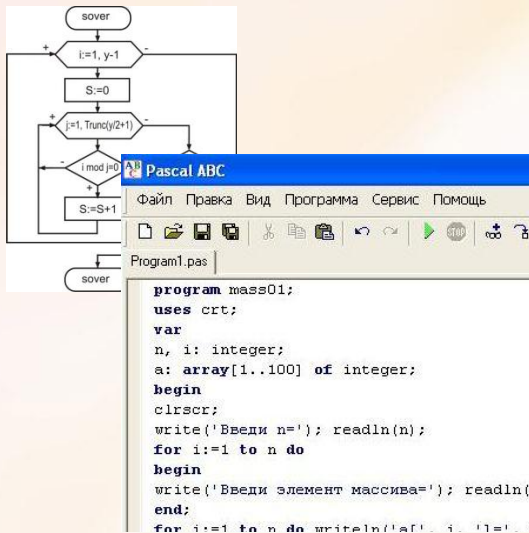
Прикладное

**программы, выполняющие
прикладные задачи**

**Инструментально
е**

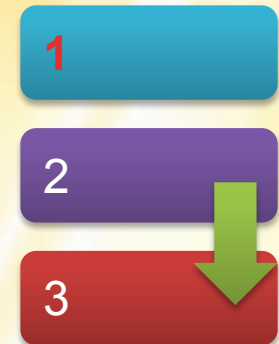
**алгоритмические языки и
системы программирования**

Алгоритмический язык - это инструментальное средство для разработки программ (язык программирования)



Программа - последовательность команд, понятных для ЭВМ и реализующих заданный алгоритм

Команда определяет один шаг процесса выполнения программы



ЯЗЫКИ ПРОГРАММИРОВАНИЯ



машинные

Позволяют разрабатывать программы, выполняемые на конкретной ЭВМ. Определяются системой команд процессора и архитектурой компьютера

ассемблера

высокого уровня

The screenshot shows a CPU simulator window titled "CPU" with a thread ID of "\$00000000". The main window displays assembly code for a program named "test.asm". The code includes instructions like "sahf", "add [eax], eax", "add [eax], al", "add al, \$70", "add [esi], dl", "add [esi], cl", "or al, \$65", "add al, \$70", "add [ebp+\$04], ah", and "in +\$00".

Registers are listed on the right side of the window:

EAX	000004D2	CF	0
EBX	00000000	PF	0
ECX	00000000	AF	0
EDX	00000400	ZF	0
ESI	00000000	SF	0
EDI	00000000	TF	0
EBP	00000000	IF	0
ESP	00000000	DF	0
EIP	00000000	OF	0
EFL	00000000	IOPL	0
CS	0000	MF	0
DS	0000	RF	0
SS	0000	VM	0
ES	0000	AC	0

Below the main window, there is a smaller window titled "D:\EVM\WWW\narodlevm\sym\l\sym.com" which shows a list of memory addresses and their contents, such as "00410000", "00410001", "00410002".

At the bottom, there is a status bar with the text: "ESC=CANCEL / LOAD FILE NAME ??test.asm USE : F1=COMPILE End.Home.FgUp.FgDn.De1.BkSpace F8=De1Str ESC=BACK TO SYMULATOR"

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

Ада – язык программирования для применения в системах реального времени (например, управление процессами и/или устройствами в бортовых ЭВМ – корабельных, авиационных и др.)



Алгол (1958 - 1960) – разработка программ для решения научно-технических задач на ЭВМ



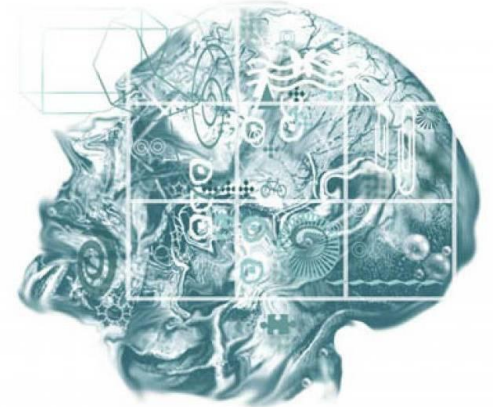
Бейсик (1963-1964) – язык для обучения программированию

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

Кобол (1959) – язык программирования для решения экономических задач (операторы выглядят как обычные английские фразы)



Лисп (1960), **Пролог**(1971) – языки для решения задач, связанных с искусственным интеллектом



Паскаль (1968-1969) – универсальный язык программирования

(обучение программированию в вузах, промышленное программирование, написание больших и сложных программ)

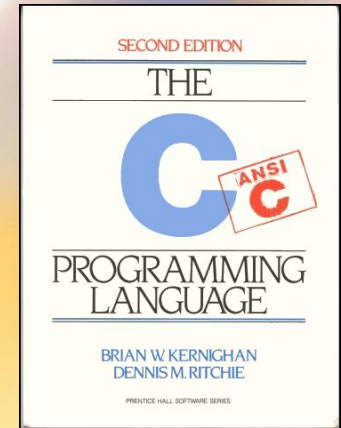
```
program test;
uses crt;
begin
  clr;
  writeln('1111111111');
end.
```

Строка: 5 Столбец: 1

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

СИ (1971) – универсальный язык программирования.

(Широко используется для разработки операционных систем, трансляторов, баз данных и других системных и прикладных программ)



Снобол (1962 -1967) – язык для обработки текстовой информации

Фортран (Formula Translation, 1957) – язык для решения математических задач

```
PROGRAM TPK
C   THE TPK ALGORITHM
C   FORTRAN 77 STYLE
REAL A(0:10)
READ (5,*) A
DO 10 I = 10, 0, -1
  Y = FUN(A(I))
  IF (Y .LT. 400) THEN
9     WRITE(6,9) I, Y
      FORMAT(I10, F12.6)
  ELSE
5     WRITE (6,5) I
      FORMAT(I10, ' TOO LARGE'
10    ENDIF
CONTINUE
END
```

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

Java, Perl, PHP – языки, ориентированные на создание серверных приложений в Интернет



HTML (1992) – язык разметки гипер (разработка Web-страниц)

VBA – языки, применяемые в различных офисных программах

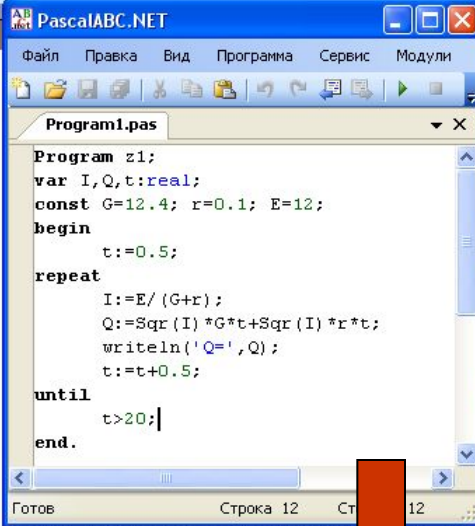


Транслятор переводит программу с языка высокого уровня на язык машины, понятный компьютеру

Трансляторы реализуются в виде компиляторов или интерпретаторов.

Компилятор транслирует всю программу целиком и создает вариант программы на машинном языке.

Интерпретатор переводит и выполняет программу строку за строкой.




The screenshot shows the PascalABC.NET IDE with a Pascal program named 'Program1.pas' and its corresponding assembly code. A red arrow points from the assembly code to the Pascal code.

```
Program z1;
var I,Q,t:real;
const G=12.4; r=0.1; E=12;
begin
  t:=0.5;
repeat
  I:=E/(G+r);
  Q:=Sqr(I)*G*t+Sqr(I)*r*t;
  writeln('Q=',Q);
  t:=t+0.5;
until
  t>20;
end.
```

```
3E0CA000 0000100 08
3E0CA002 0002E8 3D
3E0CA007 0007FF FF
3E0CA009 000938 45
3E0CA00C 000CF6 40
3E0CA010 00108F 85
3E0CA016 00168B FE
3E0CA018 0018E9 32
3E0CA01D 001DFF 4E
3E0CA020 0020E9 63
3E0CA025 002590
3E0CA026 002690
3E0CA027 002790
3E0CA028 002890
3E0CA029 002990
3E0CA02A 002A8B FF
3E0CA02C 002C55
3E0CA02D 002D8B EC
3E0CA02F 002F51
3E0CA030 003056
3E0CA031 003133 F6
3E0CA033 003364 A1 18 00 00
3E0CA039 003988 00 0F 00 00
3E0CA03F 003F85 C0
3E0CA041 004189 45 FC
3E0CA044 00448F 84 8F CA 05 00
3E0CA04A 004A8B B0 AC 00 00 00
3E0CA050 005085 F6
3E0CA052 00528F 84 5A 51 00 00
3E0CA058 00588B C6
3E0CA05A 005A5E
```

```
*,*   pop     esi
*,*   nop
*,*   mov     esi,fs:[18]
*,*   push   eax,[eax+F80]
*,*   mov     eax,eax
*,*   push   esi
*,*   push   ecx
*,*   xor     esi,esi
*,*   mov     eax,fs:[18]
*,*   test   eax,eax
*,*   mov     [ebp+8FC],eax
*,*   je/jz  3E126A09
*,*   mov     esi,[eax+00AC]
*,*   test   esi,esi
*,*   je/jz  3E0CF1B2
*,*   mov     eax,esi
*,*   pop     esi
```



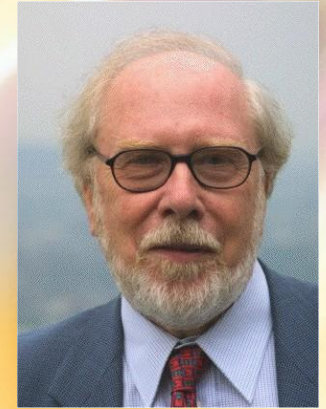
**2. Структура
программы.
Описание данных**



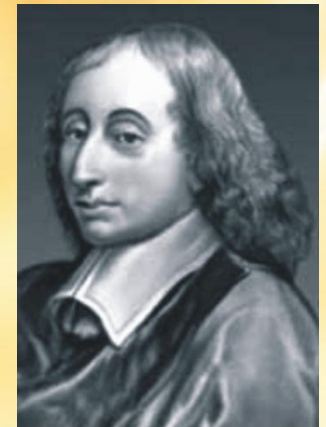
Visual Basic Simula Visual C++
HTML Oberon
APC
Cobol PL/I Eiffel Algol
Fortran Basic
awk
Lisp ADA C Pascal Icon sed
Smalltalk perl
Modula-2
Scheme Delphi JAVA

АЛГОРИТМИЧЕСКИЙ ЯЗЫК ПАСКАЛЬ

Язык разработан профессором Цюрихского технологического института **Никлаусом Виртом** в 1969 - 1971 годах.



Название язык получил в честь великого французского ученого XVII века **Блеза Паскаля**, который изобрел автоматическое устройство для суммирования чисел.



Блез Паскаль

Сначала язык предназначался для обучения студентов программированию. Но уже через 5-6 лет Паскаль приобрел широкую известность и распространился среди профессиональных программистов всего мира, благодаря простоте, хорошему восприятию материала и эффективности реализации большинства задач вычислительного характера, систем управления базами данных, экспертных систем.

Pascal, Borland Pascal, Turbo Pascal, Pascal ABC, Delphi

СТРУКТУРА ПРОГРАММЫ

Программа состоит из строк. В языке Паскаль максимальная длина строки не должна превышать 127 символов.

Program Name;

Блок описания данных

Begin

Тело программы

End.

Заголовок программы состоит из служебного слова **PROGRAM** и произвольного имени программы **Name**, задаваемого пользователем.

Имя может состоять из латинских букв, цифр и знака подчеркивания; начинается должно только с латинской буквы.

Заголовок завершается знаком ;

СТРУКТУРА ПРОГРАММЫ

Программа состоит из строк. В языке Паскаль максимальная длина строки не должна превышать 127 символов.

Program Name;

Блок описания данных

Begin

Тело программы

End.

**Блок описания данных
может включать:**

- описание меток;
- описание констант;
- описание типов данных;
- описание переменных;
- описание процедур и функций.

Любой из этих разделов может отсутствовать. Они могут встречаться в программе любое количество раз и следовать в любом порядке.

СТРУКТУРА ПРОГРАММЫ

Программа состоит из строк. В языке Паскаль максимальная длина строки не должна превышать 127 символов.

Program Name;

Блок описания данных

Begin

Тело программы

End.

Тело программы – это текст основной программы, начинается служебным словом **BEGIN** и заканчивается словом **END**.

В конце программы обязательно ставится **точка**

Знак **;** является разделителем всех операторов и строк в программе.

СТРУКТУРА ПРОГРАММЫ

Комментарии – это пояснительный текст, который можно записать в любом месте программы.

Текст комментария ограничен символами { } или (* *).

Пример: { это пояснительный текст }
Идентификаторы (* – это любые имена,*) задаваемые пользователем в программе для обозначения меток, констант, переменных, процедур и функций.
(а можно и так записать)

Все идентификаторы должны начинаться с буквы или знака подчеркивания. Не допускается использование в именах пробелов, точек и других символов.

Регистр букв (прописные, строчные) в именах и служебных словах значения не имеет.

Пример: Metka1, Blok_38, _Dom - правильно записанные имена
12Gr, Blok 5, Dom.5 - ошибки в именах идентификаторов!



ОПИСАНИЕ ДАННЫХ В ЯЗЫКЕ ПАСКАЛЬ

Описание констант

Константы – это элементы данных, значения которых известны и в процессе выполнения программы не изменяются.

Для описания констант используется служебное слово **Const**

Формат записи:

```
Const Имя константы = значение;
```

Пример: **Const** Max=100;
 A=8.3; B=-5.1;

Для обозначения числа **π** применяется стандартный идентификатор **Pi**, не требующий описания

ОПИСАНИЕ ДАННЫХ В ЯЗЫКЕ ПАСКАЛЬ

Описание переменных

Переменные – это данные, которые могут изменять свои значения в процессе выполнения программы. Каждая встречающаяся в программе переменная должна быть описана до начала программы в блоке описания данных !

Для описания переменных используется служебное слово **Var**

Формат записи:

```
Var Имя переменной : тип  
переменной;
```

Тип переменной – это формат ее значения: число, символ, строка и т.д.

ОПИСАНИЕ ДАННЫХ В ЯЗЫКЕ ПАСКАЛЬ

Описание переменных

Стандартные типы переменных:

1. **Integer** – целые числа в диапазоне от -32768 до +32767.
2. **Real** – вещественные (дробные) числа.
3. **Byte** – целые числа в диапазоне от 0 до 255.
4. **String** – строковый тип, использующий строковые данные.
5. **Char** – символный тип (буквы, цифры, символы и знаки).
6. **Boolean** – логические переменные, принимающие только одно из двух значений: True (истина) или False (ложь).

3. Простейшие операции



ПРОСТЕЙШИЕ ОПЕРАЦИИ

Арифметические операции

Операция	Команда	Пример	Результат
Сложение Вычитание	$+$ $-$		
Умножение Деление	$*$ $/$		
Целочисленное деление (дробная часть отбрасывается)	div	14 div 5 2 div 6	2 0
Деление по модулю (остаток от деления)	mod	11 mod 5 24 mod 5	1 4

ПРОСТЕЙШИЕ ОПЕРАЦИИ

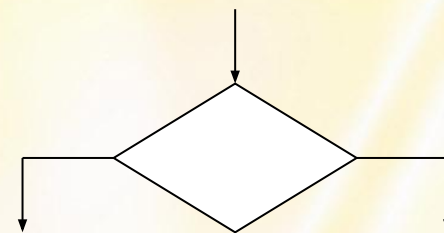
Операции отношения (сравнения)

Операция	Знак
Равно	=
Не равно	<>
Меньше	<
Больше	>
Меньше или равно	<=
Больше или равно	>=

Логические выражения

Операция	Выражение
И	and
ИЛИ	or
отрицание	not

Операции сравнения и логические выражения используются для проверки логических условий, т.е. для описания логических блоков



ПРОСТЕЙШИЕ ОПЕРАЦИИ

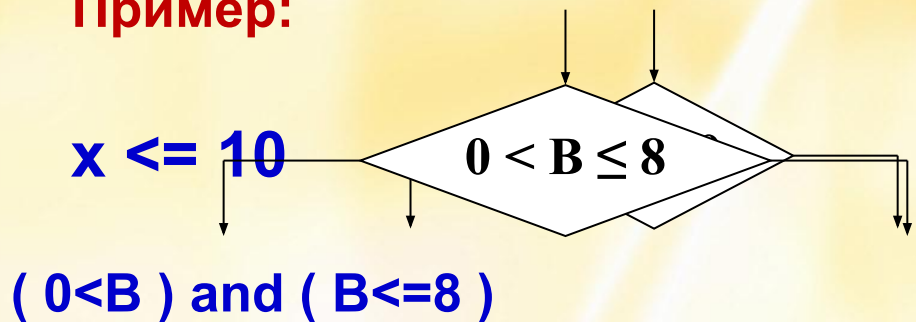
Операции отношения (сравнения)

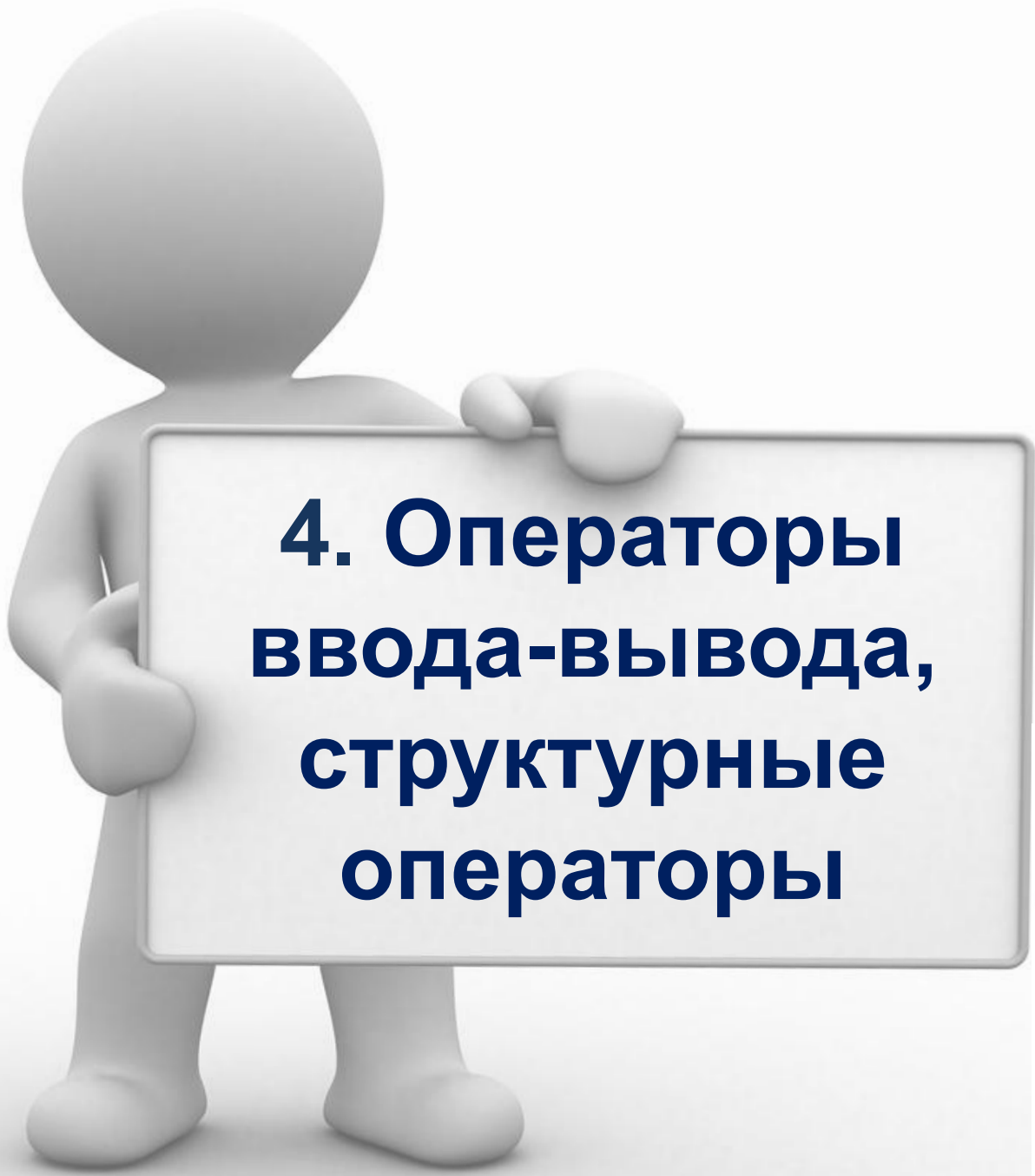
Операция	Знак
Равно	=
Не равно	<>
Меньше	<
Больше	>
Меньше или равно	<=
Больше или равно	>=

Логические выражения

Операция	Выражение
И	and
ИЛИ	or
отрицание	not

Пример:





**4. Операторы
ввода-вывода,
структурные
операторы**



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

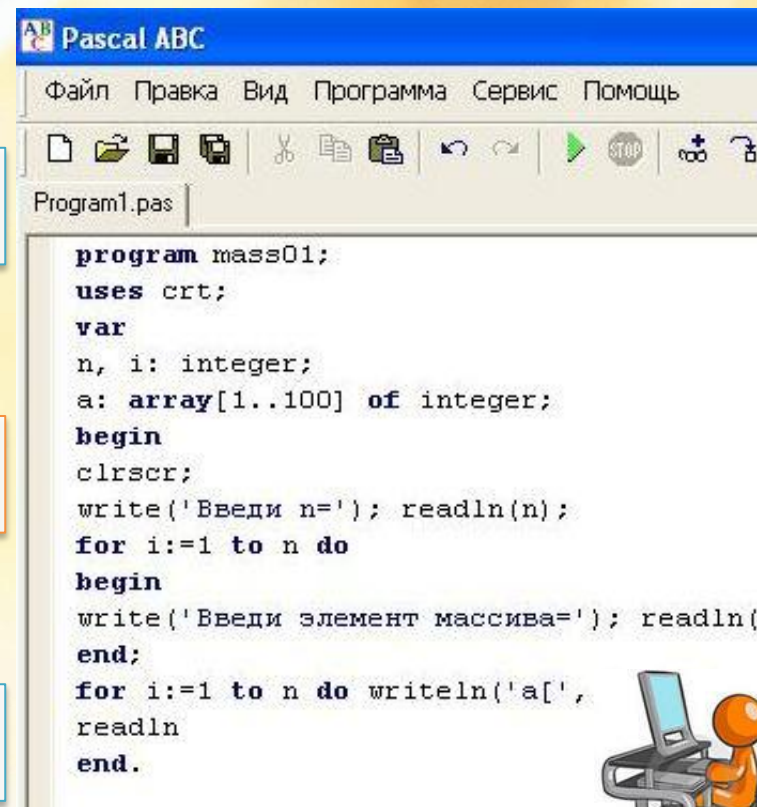
Программа состоит из последовательности операторов, выполняемых строго друг за другом в порядке их описания слева направо и сверху вниз.

Операторы :

Простые

Ввода-вывода

Структурные



```
program mass01;
uses crt;
var
n, i: integer;
a: array[1..100] of integer;
begin
clrscr;
write('Введи n='); readln(n);
for i:=1 to n do
begin
write('Введи элемент массива='); readln(i);
end;
for i:=1 to n do writeln('a[', i, ']=', a[i]);
end.
```



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

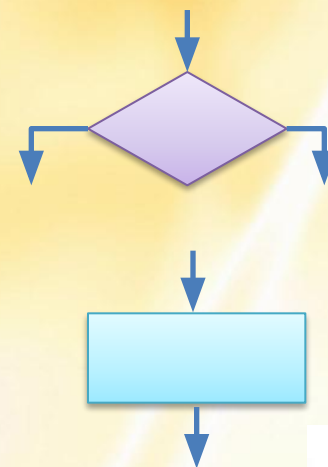
Простые операторы

1. **Оператор присваивания** **:=** вычисляет выражение справа от него и присваивает результат переменной, стоящей слева.

Пример: $A:=1;$ $B:=\sin(C)+\cos(D);$
 $N:=N+1;$ $S:=S+y;$

Обычный знак равенства **=** используется для сравнения данных или переменных.

Во всех арифметических выражениях, где требуются вычисления, используется именно оператор присваивания **:=**



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Простые операторы

1. **Оператор присваивания** `:=` вычисляет выражение справа от него и присваивает результат переменной, стоящей слева.

Пример: `A:=1; B:=sin(C)+cos(D);`
`N:=N+1; S:=S+y;`

2. **Оператор безусловного перехода** **Goto** применяется для перехода на заданную метку из любого места программы.

Метка позволяет выполнить переход к отмеченному оператору или строке из любого места программы.



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Операторы ввода-вывода

1. **Оператор ввода** (чтения) обеспечивает ввод данных с клавиатуры (либо чтение из файла) для их последующей обработки программой.

Описание: **Read**(x_1, x_2, \dots, x_n);
 ReadLn(x_1, x_2, \dots, x_n);

$x_1 \dots x_n$ – переменные, значения которых необходимо задать с клавиатуры

Пример: **Read**(A,B); **ReadLn**(x);



Все переменные, перечисленные в одном операторе ввода можно задавать в одной строке друг за другом через пробел.



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Операторы ввода-вывода

2. **Оператор вывода** (записи) обеспечивает вывод данных на экран монитора (либо запись в файл).

Описание: **Write**(y_1, y_2, \dots, y_n);

WriteLn(y_1, y_2, \dots, y_n);

$y_1 \dots y_n$ – переменные, значения которых необходимо вывести на экран монитора

Пример: **Write**(A+B-2); **WriteLn**(x,y);

Вывод
x,y

Значения переменных, перечисленных в операторе вывода через запятую, выводятся в одной строке.



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Операторы ввода-вывода

2. **Оператор вывода** (записи) обеспечивает вывод данных на экран монитора (либо запись в файл).

Описание: **Write**(y_1, y_2, \dots, y_n);

WriteLn(y_1, y_2, \dots, y_n);

$y_1 \dots y_n$ – переменные, значения которых необходимо вывести на экран монитора

При выводе на экран текстовой информации текст в операторе записывается в одинарных кавычках.

Пример: **WriteLn**('Решения нет');

Write('Ответ:');



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Операторы ввода-вывода

2. **Оператор вывода** (записи) обеспечивает вывод данных на экран монитора (либо запись в файл).

Описание: **Write**(y_1, y_2, \dots, y_n);

WriteLn(y_1, y_2, \dots, y_n);

$y_1 \dots y_n$ – переменные, значения которых необходимо вывести на экран монитора

В операторе вывода одновременно могут использоваться переменные различных типов.

Пример: A:=3; B:=8; C:=A+B;

WriteLn('C=',C);

Результат: C=11



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Операторы ввода-вывода

2. **Оператор вывода** (записи) обеспечивает вывод данных на экран монитора (либо запись в файл).

Пример описания оператора вывода

WriteLn (**X** : **n₁** : **n₂**);

X – переменная, значение которой выводится на экран;

n₁ – число символов на экране для вывода значения переменной **X**;

n₂ – число символов после запятой (для дробных чисел типа **real**).

Пример:

x:=33.5391;

WriteLn(x);

WriteLn(x:10);

WriteLn(x:5:2);

Результат:

3.3539100000E+01

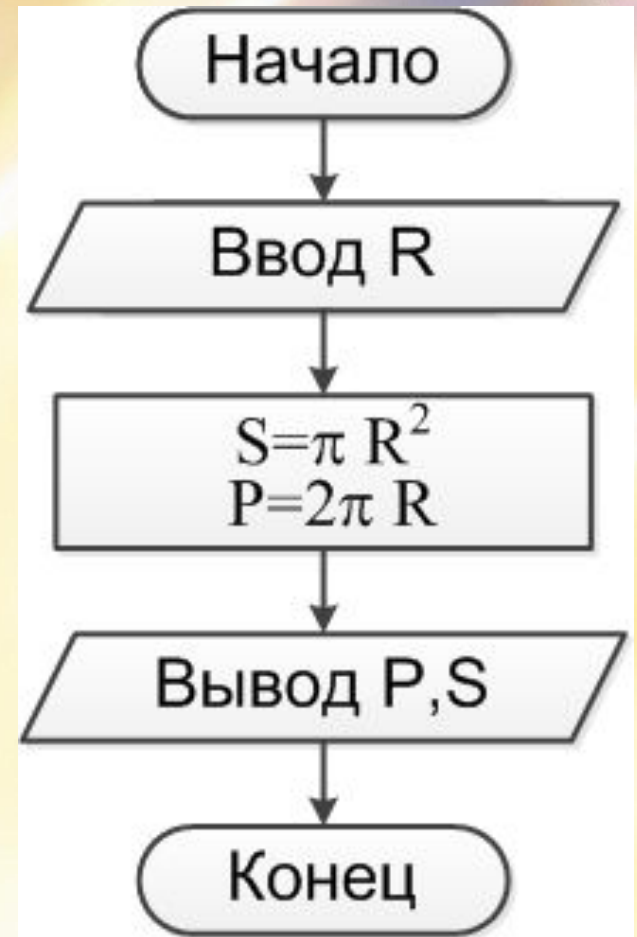
3.3539E+01

33.54



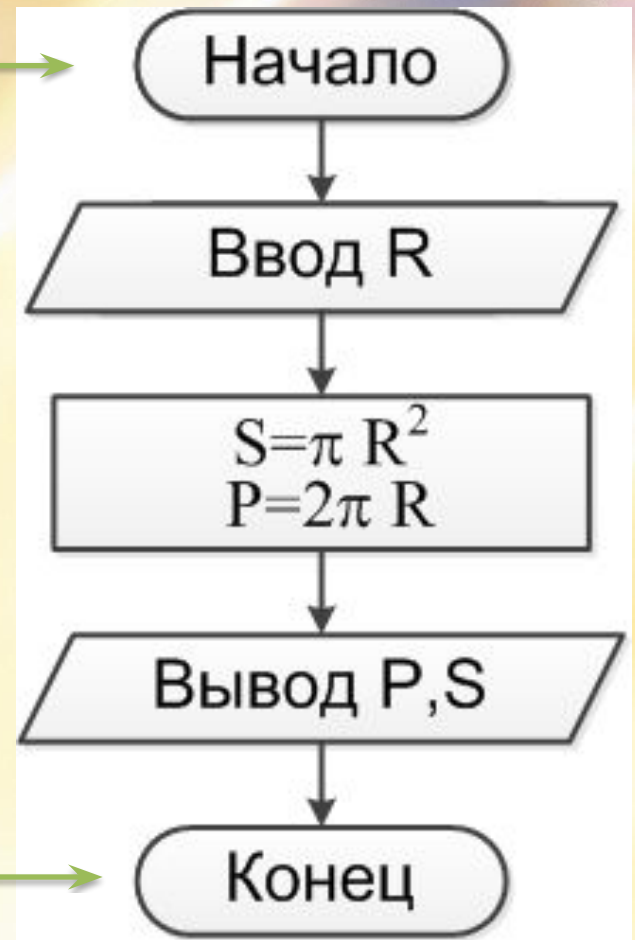
ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Задача 1: Вычислить площадь и периметр окружности заданного радиуса R .



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

```
Program Z1;  
Var R,S,P : Real;  
Begin  
  Readln(R);  
  P:=2*Pi*R;  
  S:=Pi*R*R;  
  Writeln(P,S);  
End.
```

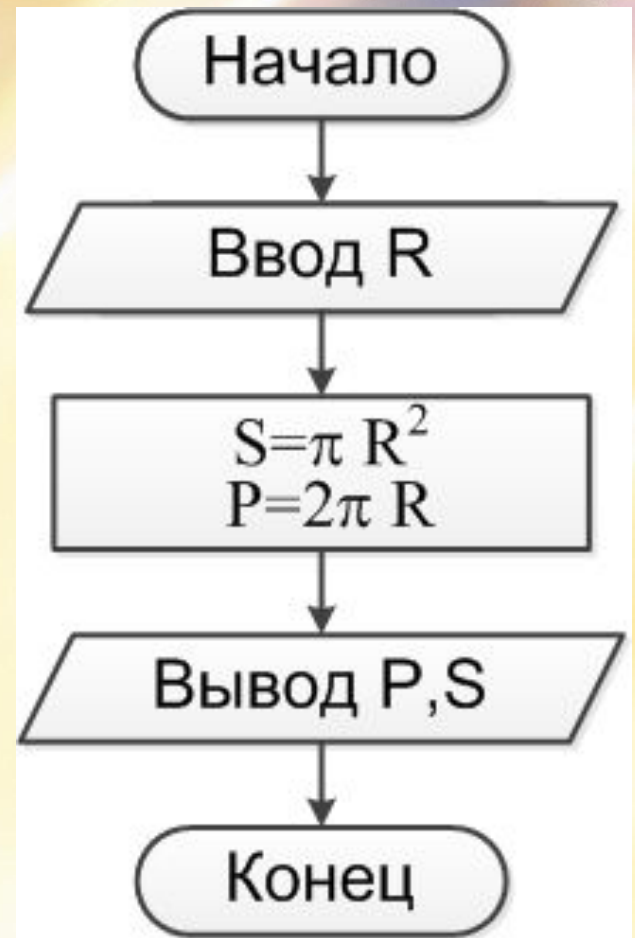


ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

```
Program Z1;  
Var R,S,P : Real;  
Begin  
    Readln(R);  
    P:=2*Pi*R;  
    S:=Pi*R*R;  
    Writeln(P,S);  
End.
```

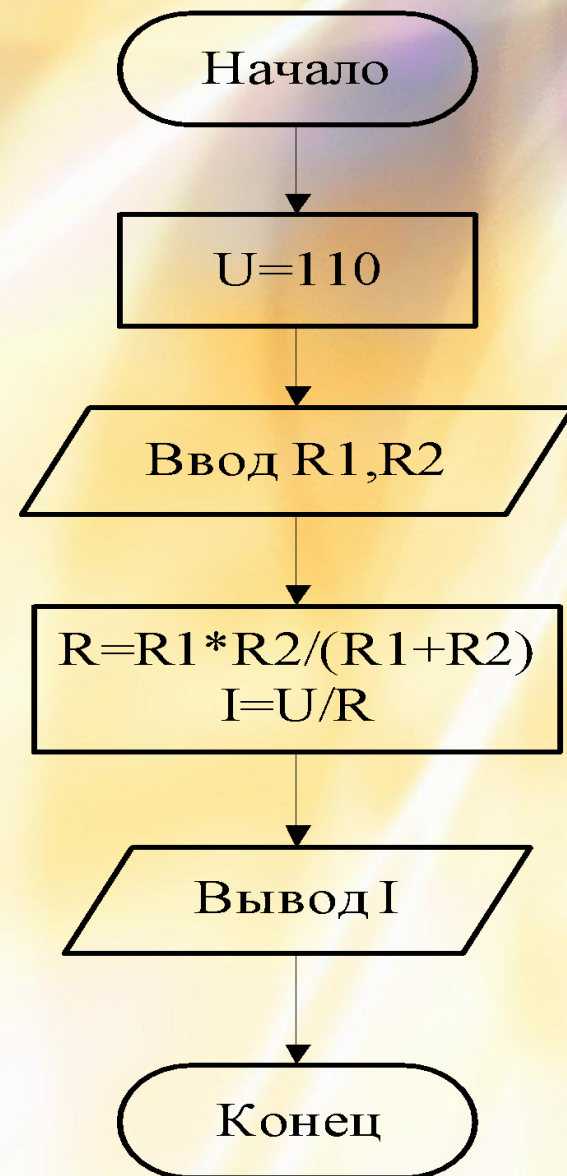
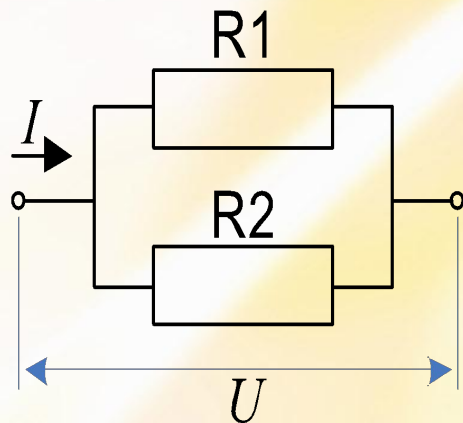
Заголовок
Блок описания
данных

Тело
программы



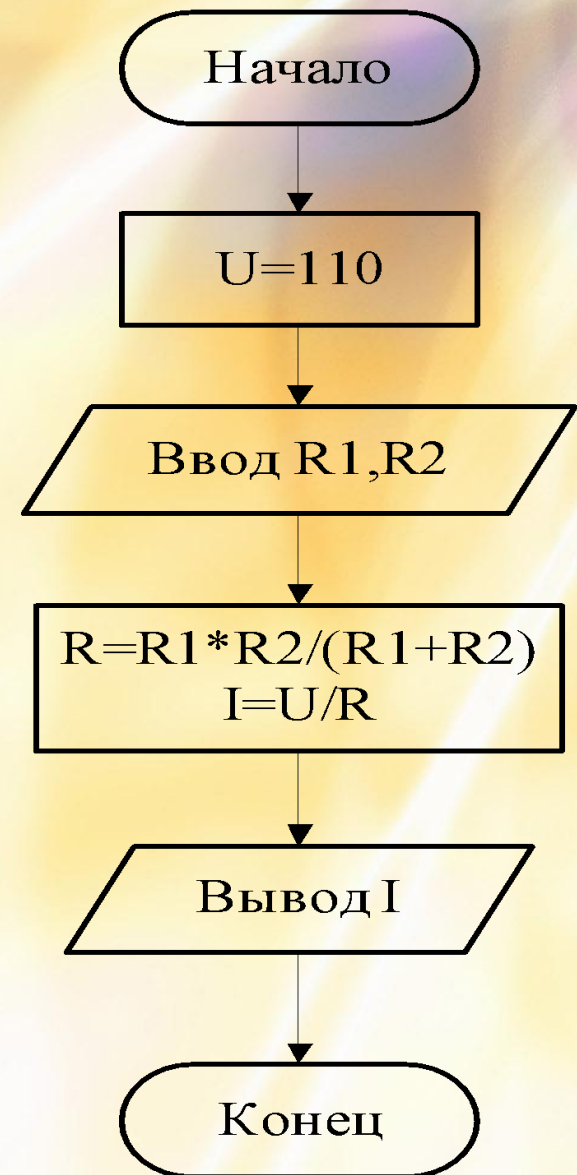
ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

Задача 2: Дана электрическая схема, в которой $U=110$ В. Для произвольно заданных значений сопротивлений $R1$ и $R2$ вычислить ток I , проходящий через цепь.



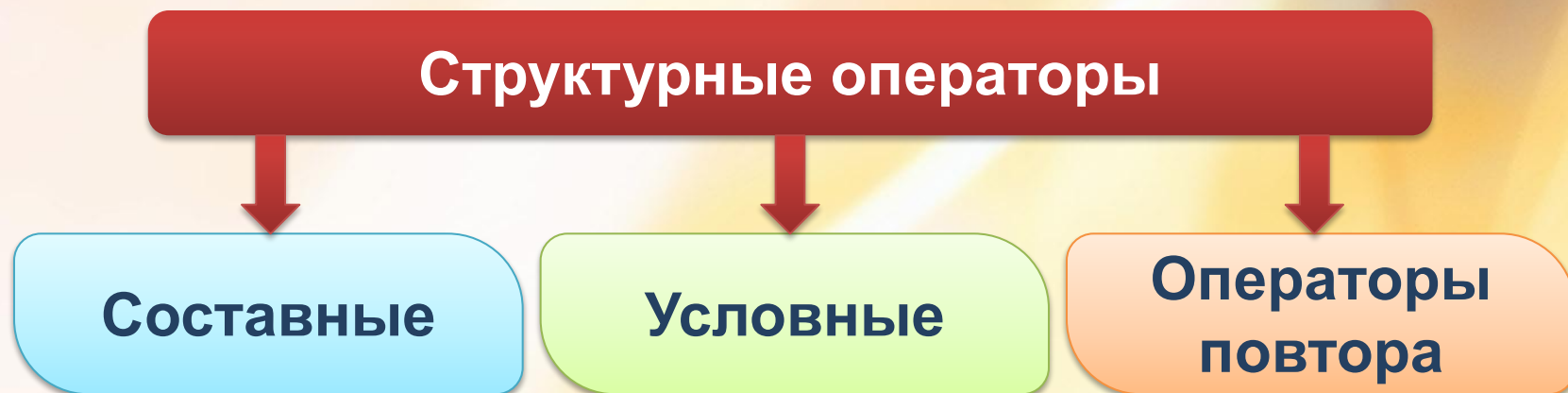
ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

```
Program Z2;  
Const U=110;  
Var R1, R2, R, I : Real;  
Begin  
  Readln(R1, R2);  
  R:=R1*R2/(R1+R2);  
  I:=U/R;  
  Writeln(I);  
End.
```



СТРУКТУРНЫЕ ОПЕРАТОРЫ

Структурные операторы - это структуры, построенные из других операторов по определенным правилам.



Составные операторы – это любая группа операторов в теле программы, ограниченная словами **Begin** и **End**.

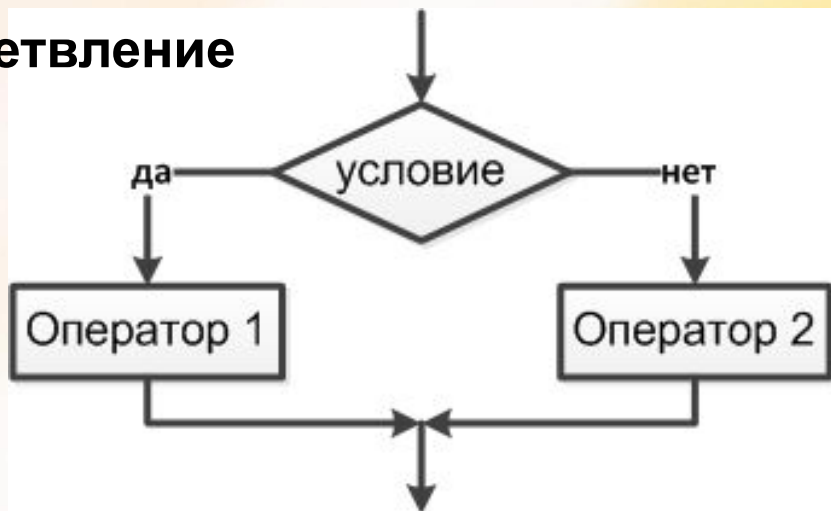
Пример: `begin`
 `y:=Sin(x);`
 `Writeln(x,y);`
`end;` } *составной оператор*

СТРУКТУРНЫЕ ОПЕРАТОРЫ

Условный оператор **if** обеспечивает выполнение оператора или группы операторов в зависимости от заданных условий.

Варианты записи условного оператора if

а) ветвление



условие **then оператор1**
else оператор2;

Если **условие** выполняется, то работает **оператор1**, в противном случае работает **оператор2**



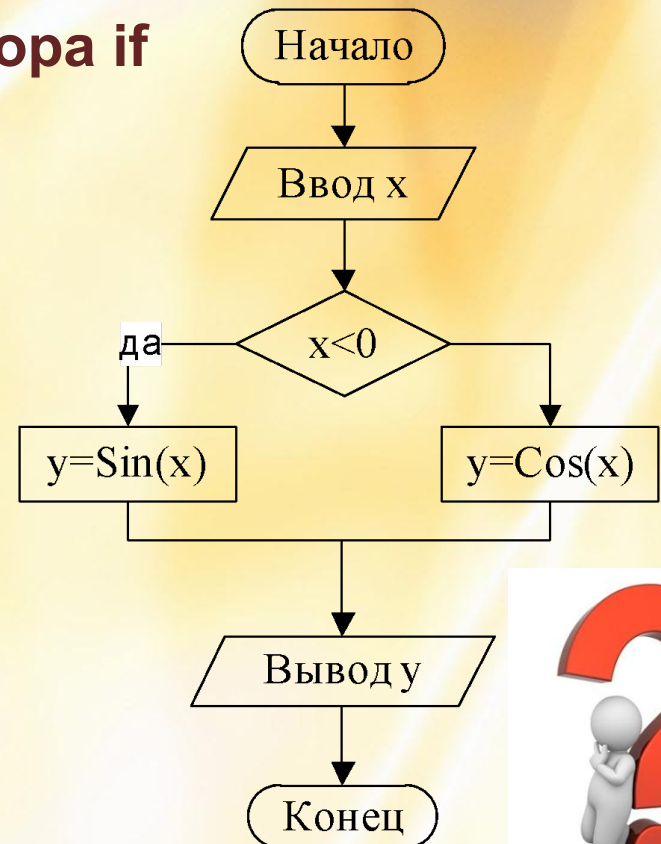
СТРУКТУРНЫЕ ОПЕРАТОРЫ

Условный оператор if обеспечивает выполнение оператора или группы операторов в зависимости от заданных условий.

Варианты записи условного оператора if

Задача 3: Для заданного значения переменной **x** вычислить

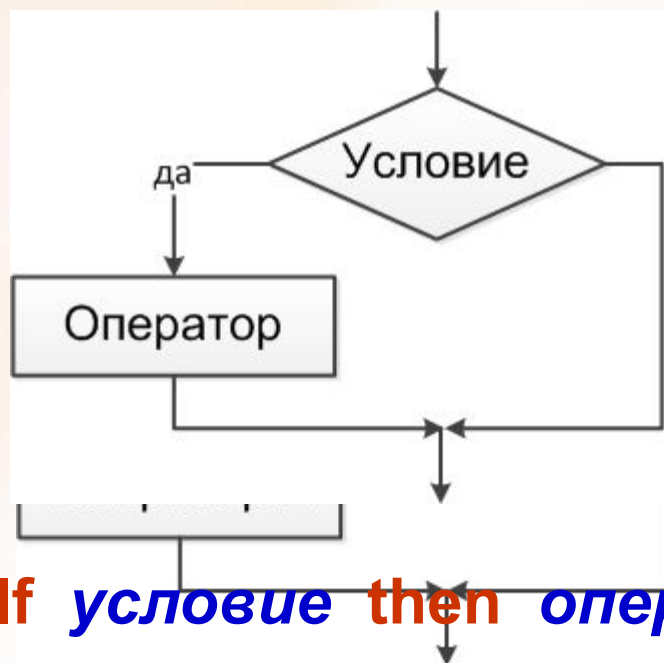
$$y = \begin{cases} \sin(x), & \text{если } x < 0 \\ \cos(x), & \text{если } x \geq 0 \end{cases}$$



СТРУКТУРНЫЕ ОПЕРАТОРЫ

Условный оператор **if** обеспечивает выполнение оператора или группы операторов в зависимости от заданных условий.

Варианты записи условного оператора if



If *условие* **then**
Begin
оператор1;
оператор2
End;

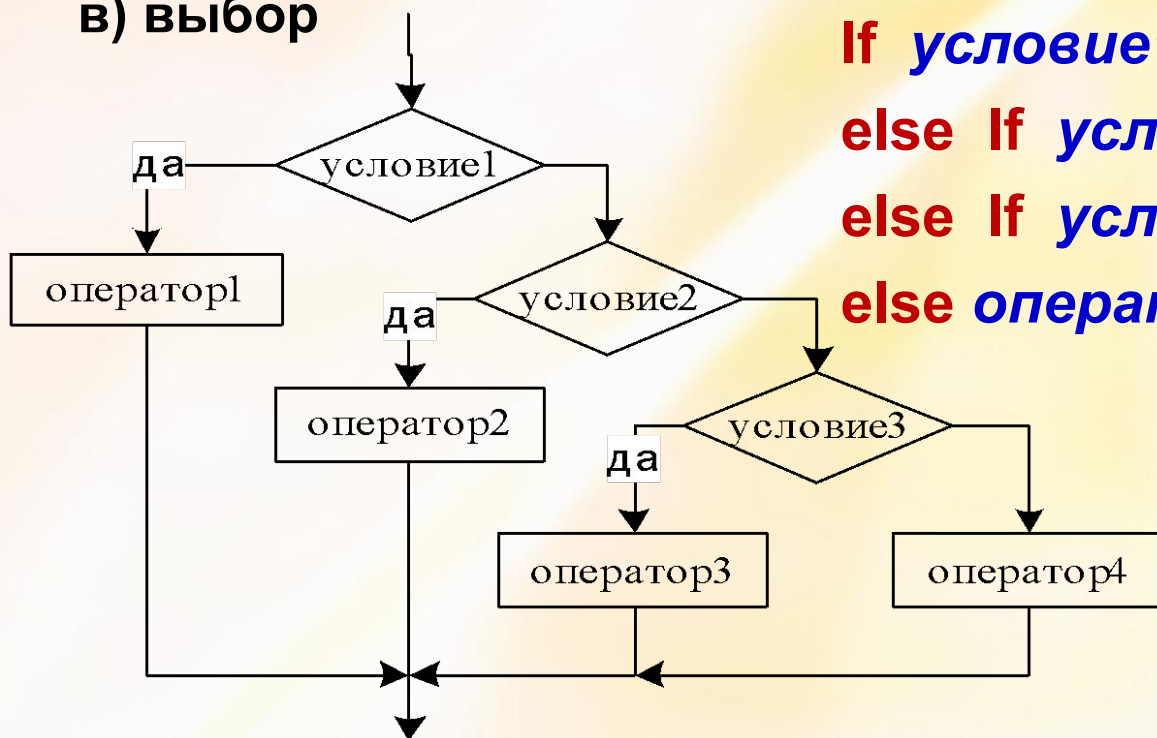


СТРУКТУРНЫЕ ОПЕРАТОРЫ

Условный оператор **if** обеспечивает выполнение оператора или группы операторов в зависимости от заданных условий.

Варианты записи условного оператора if

в) выбор



If *условие1* **then** *оператор1*
else If *условие2* **then** *оператор2*
else If *условие3* **then** *оператор3*
else *оператор4*;

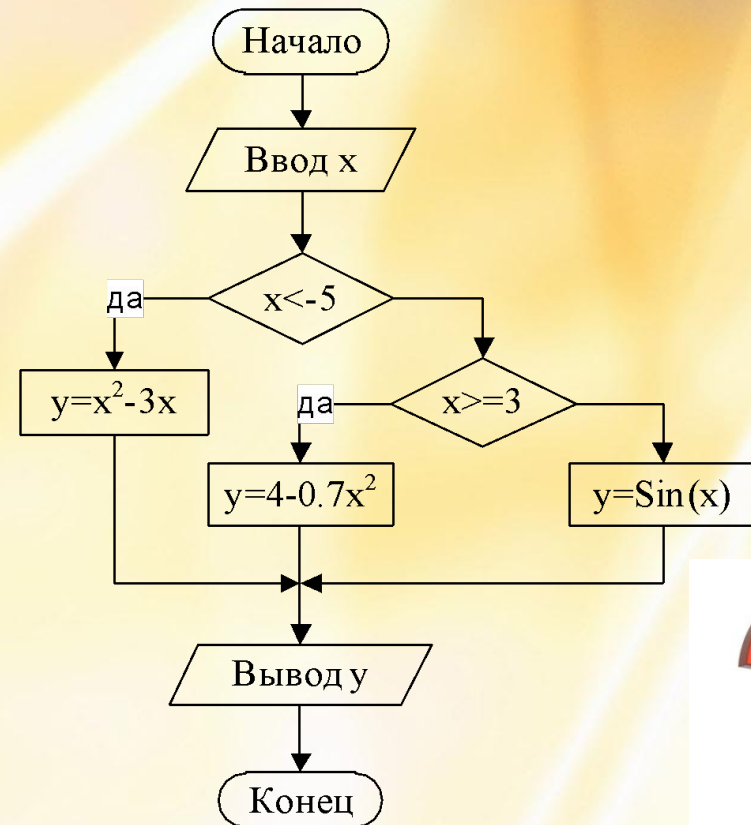


СТРУКТУРНЫЕ ОПЕРАТОРЫ

Условный оператор **if** обеспечивает выполнение оператора или группы операторов в зависимости от заданных условий.

Задача 4: Для заданного значения переменной **x** **вычислить**

$$y = \begin{cases} x^2 - 3x, & \text{если } x < -5 \\ \sin x, & \text{если } -5 \leq x < 3 \\ 4 - 0.7x^2, & \text{если } x \geq 3 \end{cases}$$



СТАНДАРТНЫЕ ФУНКЦИИ ЯЗЫКА ПАСКАЛЬ

Функция	Назначение	Пример	Результат
Abs	вычисление модуля	<code>y:=abs(-5.6)</code>	5.6
ArcTan	вычисление арктангенса	<code>y:=arctan(x)</code>	
Cos, Sin	синус, косинус числа	<code>z:=sin(x)</code>	
Exp	экспонента числа e^x	<code>y:=exp(x)</code>	
Ln	натуральный логарифм	<code>a:=ln(x)</code>	
Frac	вычисление дробной части	<code>d:=frac(7.24)</code> <code>x:=frac(-3.18)</code>	0.24 -0.18
Int	вычисление целой части	<code>y:=int(3.89)</code>	3

СТАНДАРТНЫЕ ФУНКЦИИ ЯЗЫКА ПАСКАЛЬ

Функция	Назначение	Пример	Результат
Sqr	возведение в квадрат	<code>x:=sqr(6)</code>	36
Sqrt	извлечение корня квадр.	<code>y:=sqrt(81)</code>	9
Random(K)	Выбор случайного числа от 0 до K	<code>x:=Random(10)</code>	0..10
Round	Округление до ближайшего целого числа	<code>y:=round(7.9)</code> <code>x:=round(-2.1)</code>	8 -2

Возведение в произвольную степень $y=x^a$

`y:=exp(a*ln(x))`

Вычисление логарифма

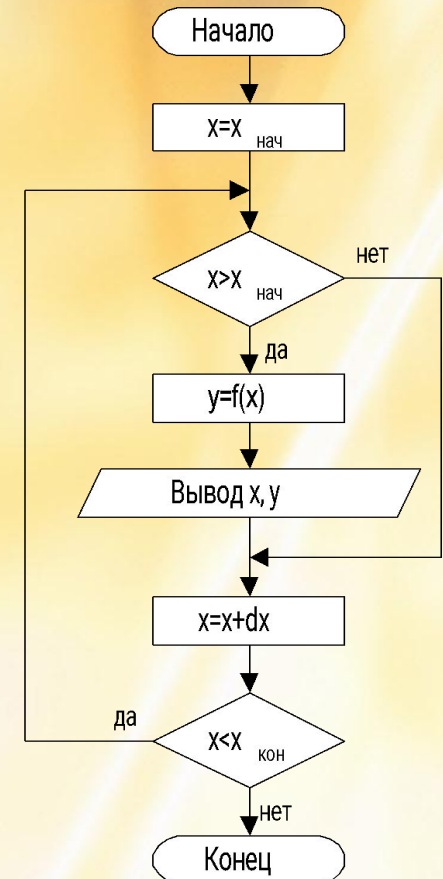
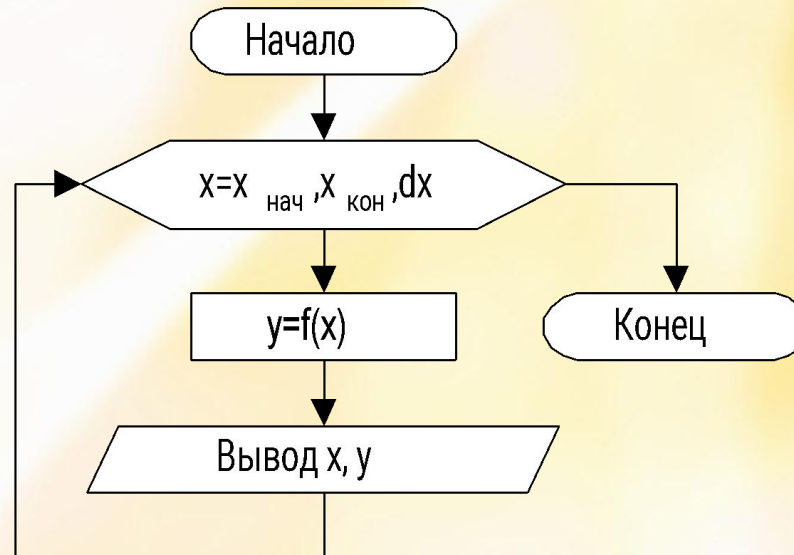
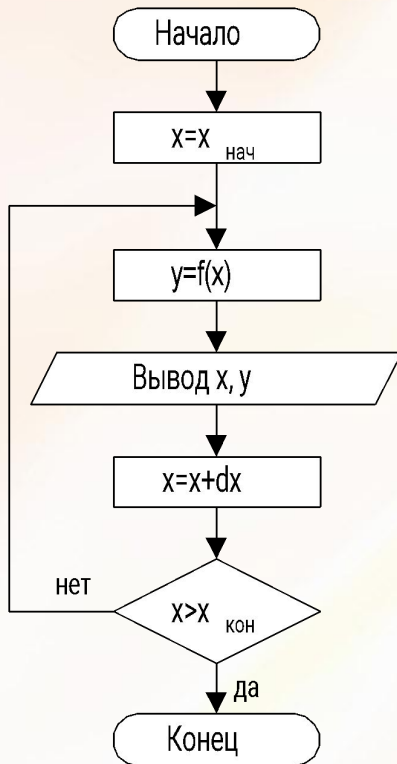
$y=\log_a x$

`y:=ln(x)/ln(a)`

ОПЕРАТОРЫ ПОВТОРА

Операторы повтора используются для описания циклических структур.

Цикл - это последовательность операторов, которая может повторяться более 1 раза



ОПЕРАТОРЫ ПОВТОРА

1) **Оператор повтора For** используется в циклах с шагом изменяемой переменной **+1** или **-1**.

В операторе For не допускается изменение параметра цикла на величину, отличную от ± 1 .

Описание:

For $x := x_{нач}$ **to** $x_{кон}$ **do оператор** (для шага $\Delta x = 1$)

For $x := x_{нач}$ **downto** $x_{кон}$ **do оператор** (для шага $\Delta x = -1$)

x – изменяемая переменная цикла (обязательно тип **Integer**);

$x_{нач}$, $x_{кон}$ – начальное и конечное значения изменяемой переменной (тип **Integer**).



ОПЕРАТОРЫ ПОВТОРА

1) **Оператор повтора For** используется в циклах с шагом изменяемой переменной **+1** или **-1**.

В операторе For не допускается изменение параметра цикла на величину, отличную от ± 1 .

Описание:

For $x := x_{нач}$ **to** $x_{кон}$ **do**

begin

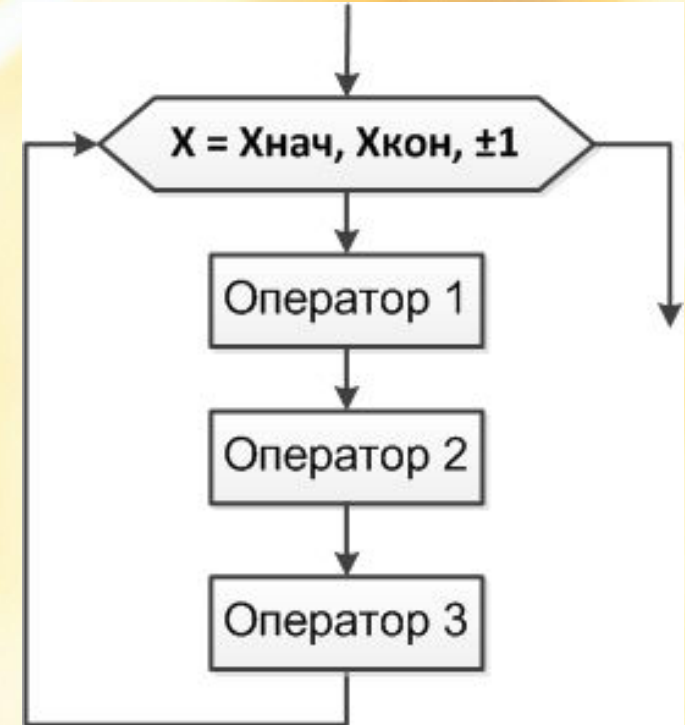
оператор1;

оператор2;

оператор3

end;

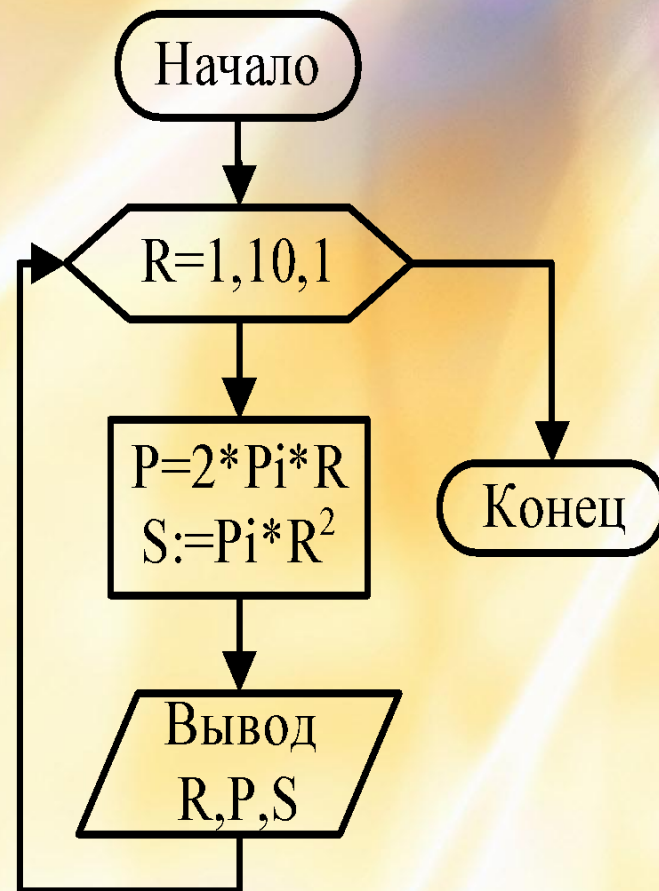
Составной оператор



ОПЕРАТОРЫ ПОВТОРА

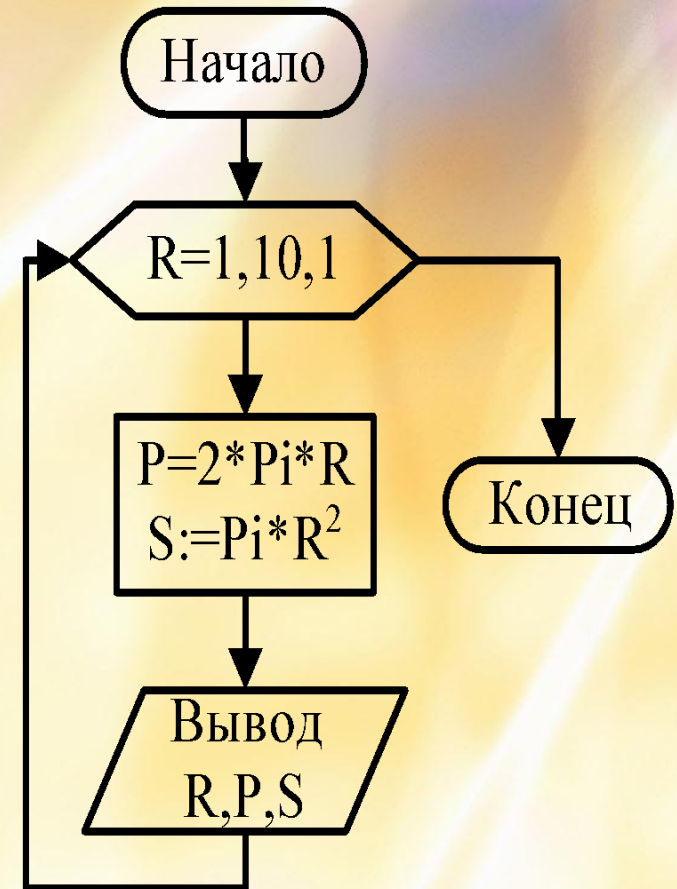
Задача 1.

Вычислить периметр и площадь круга при изменении радиуса от 1 до 10 см с шагом 1 см.



ОПЕРАТОРЫ ПОВТОРА

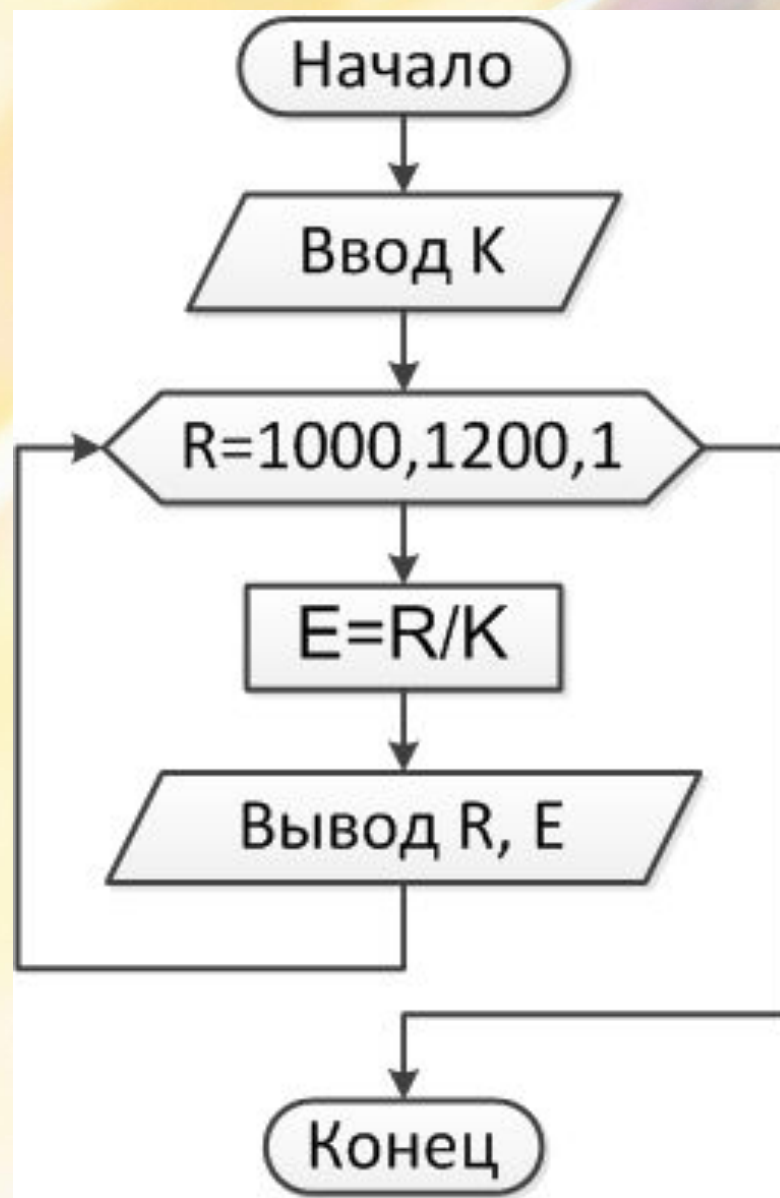
```
Program Krug;  
Var R : Integer;  
      P, S : Real;  
Begin  
  For R:=1 to 10 do  
    Begin  
      P:=2*Pi*R;  
      S:=Pi*Sqr(R);  
      Writeln(R, P, S);  
    End;  
End.
```



ОПЕРАТОРЫ ПОВТОРА

Задача 2.

Составить таблицу перевода из рублей в евро от 1000 до 1200 рублей с шагом в 1 рубль. Курс валюты задать.



ОПЕРАТОРЫ ПОВТОРА

2) **Оператор повтора Repeat** состоит из заголовка (**Repeat**), тела цикла и условия окончания цикла (**until**).

Применяется для циклических структур с произвольным шагом изменяемой переменной.

Описание:

$x := x_{нач}$;

Repeat

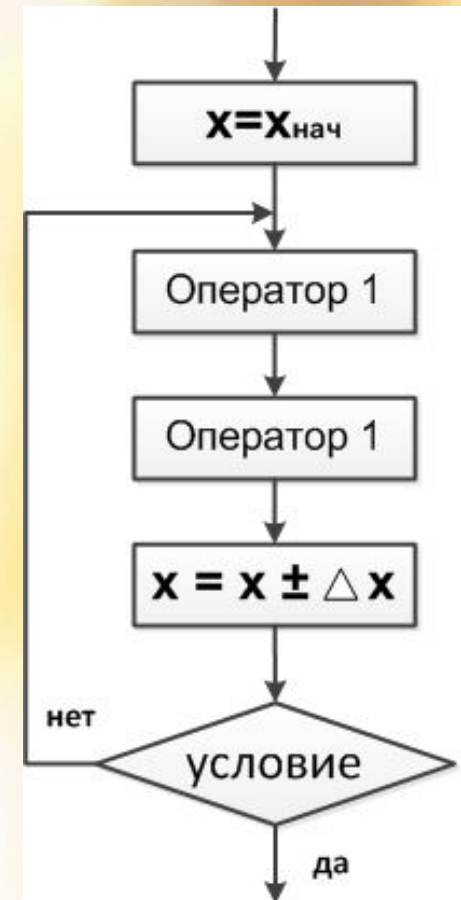
оператор1;

оператор2;

...

$x := x \pm \Delta x$

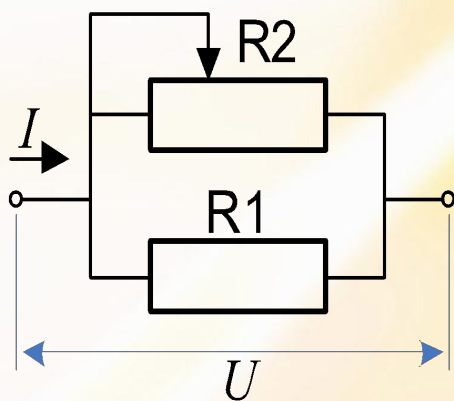
Until условие выхода из цикла;



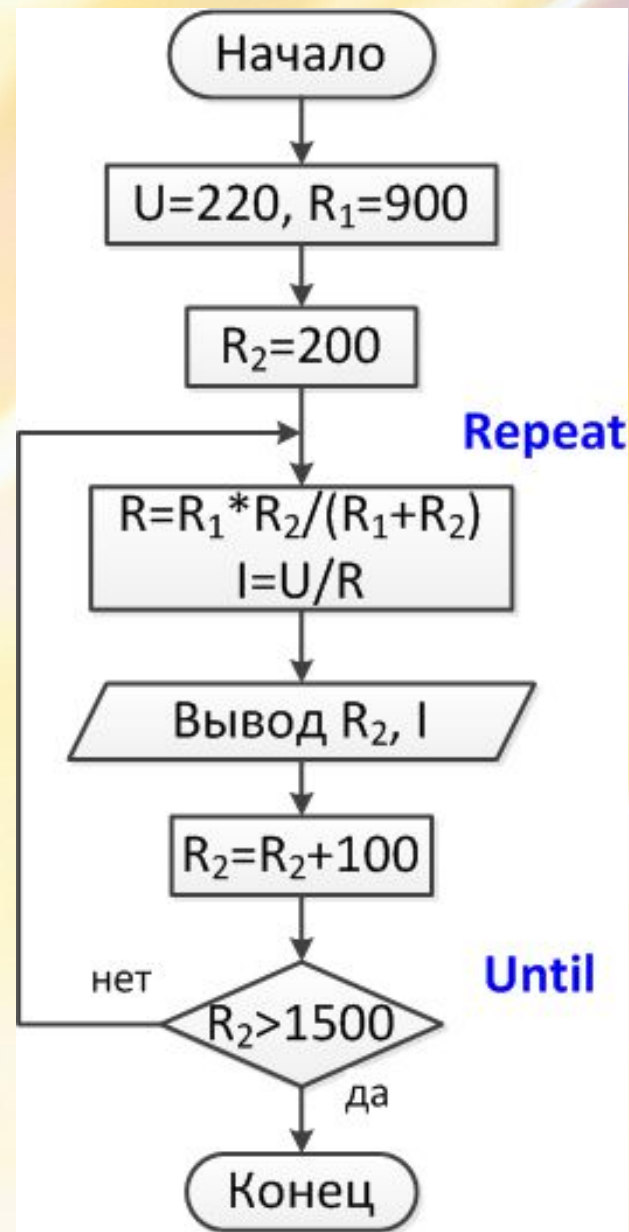
ОПЕРАТОРЫ ПОВТОРА

Задача 3:

Дана электрическая схема, в которой $U=220$ В, $R_1=900$ Ом. Рассчитать значения тока в цепи при изменении сопротивления R_2 от 200 до 1500 Ом с шагом 100 Ом.



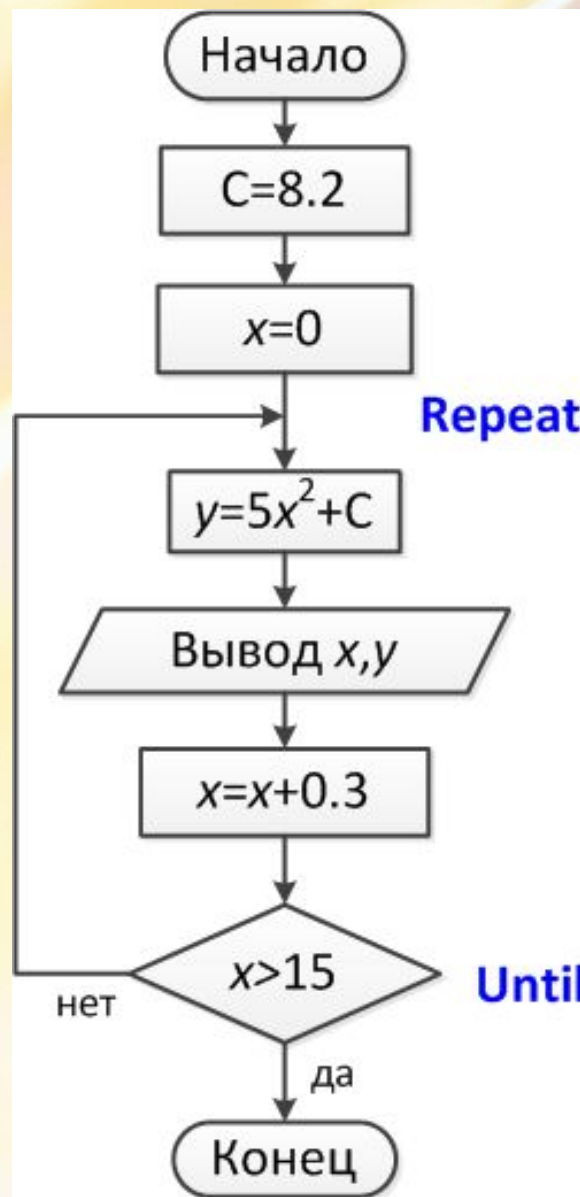
$$200 \leq R_2 \leq 1500, \quad \Delta R_2 = 100$$



ОПЕРАТОРЫ ПОВТОРА

Задача 4:

Вычислить $y=5x^2+C$ для значений $0 \leq x \leq 15$, $\Delta x=0.3$, $C=8.2$.



ОПЕРАТОРЫ ПОВТОРА

3) Оператор повтора While проводит проверку условия в начале, до выполнения тела цикла.

Оператор соответствует циклу с предусловием (цикл ПОКА).

Описание:

$x := x_{нач}$;

While условие выполнения цикла **do**

Begin

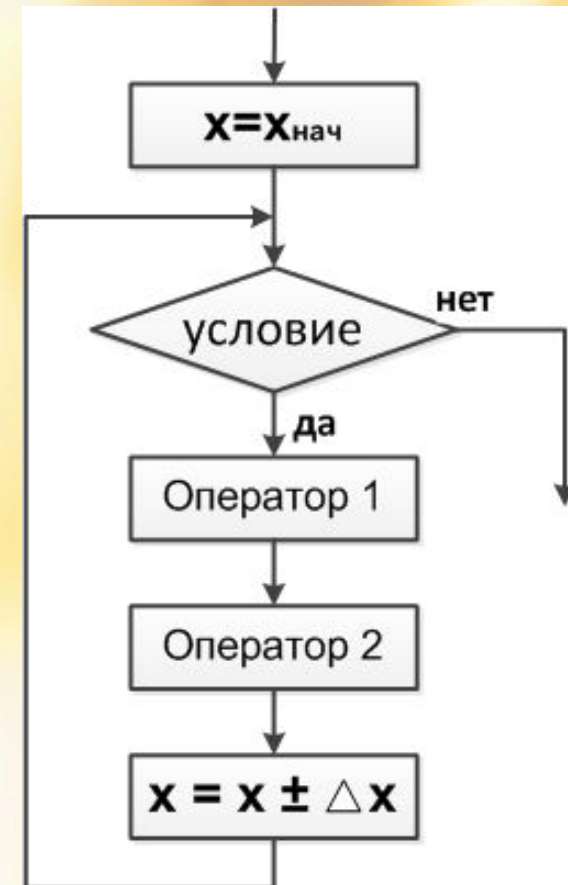
оператор1;

оператор2;

...

$x := x \pm \Delta x$

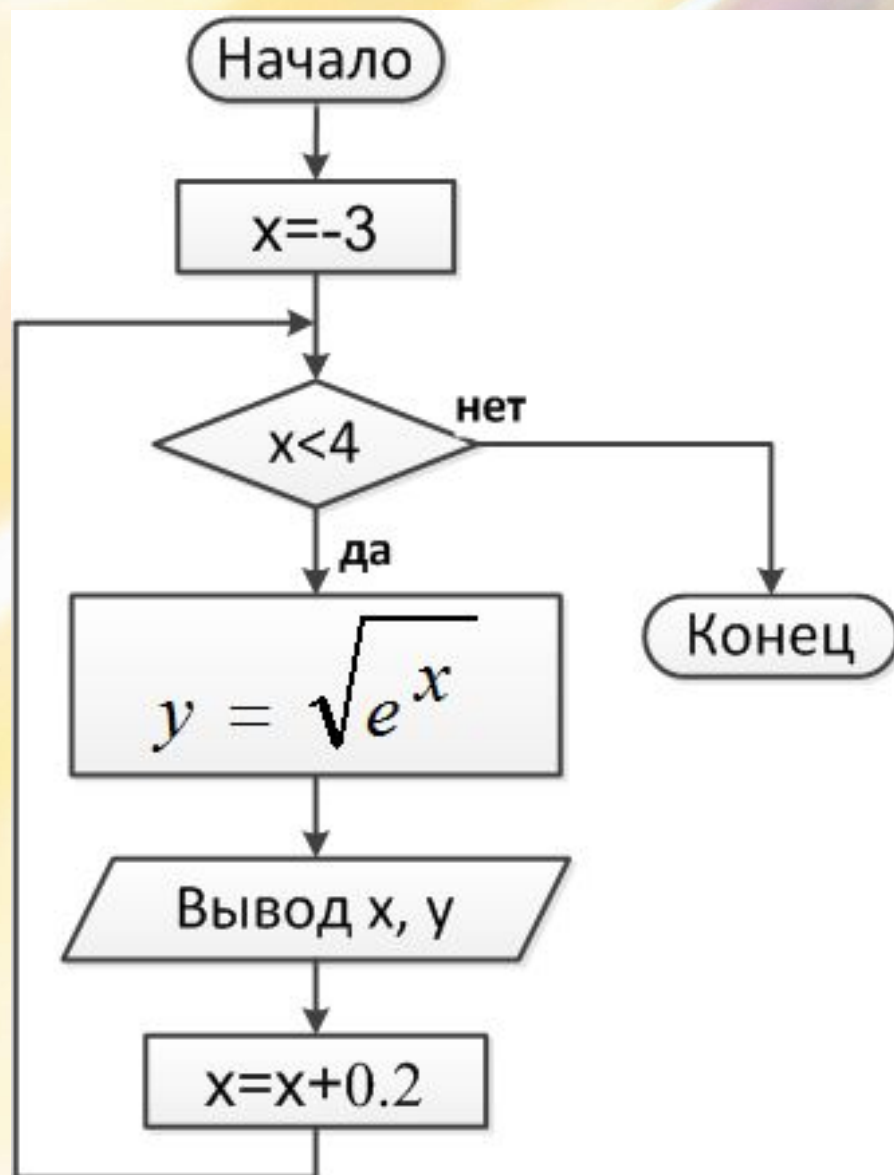
End;



ОПЕРАТОРЫ ПОВТОРА

Задача 5:

Вычислить $y = \sqrt{e^x}$
для значений $-3 \leq x < 4$
с шагом $\Delta x = 0.2$



ОПЕРАТОРЫ ПОВТОРА

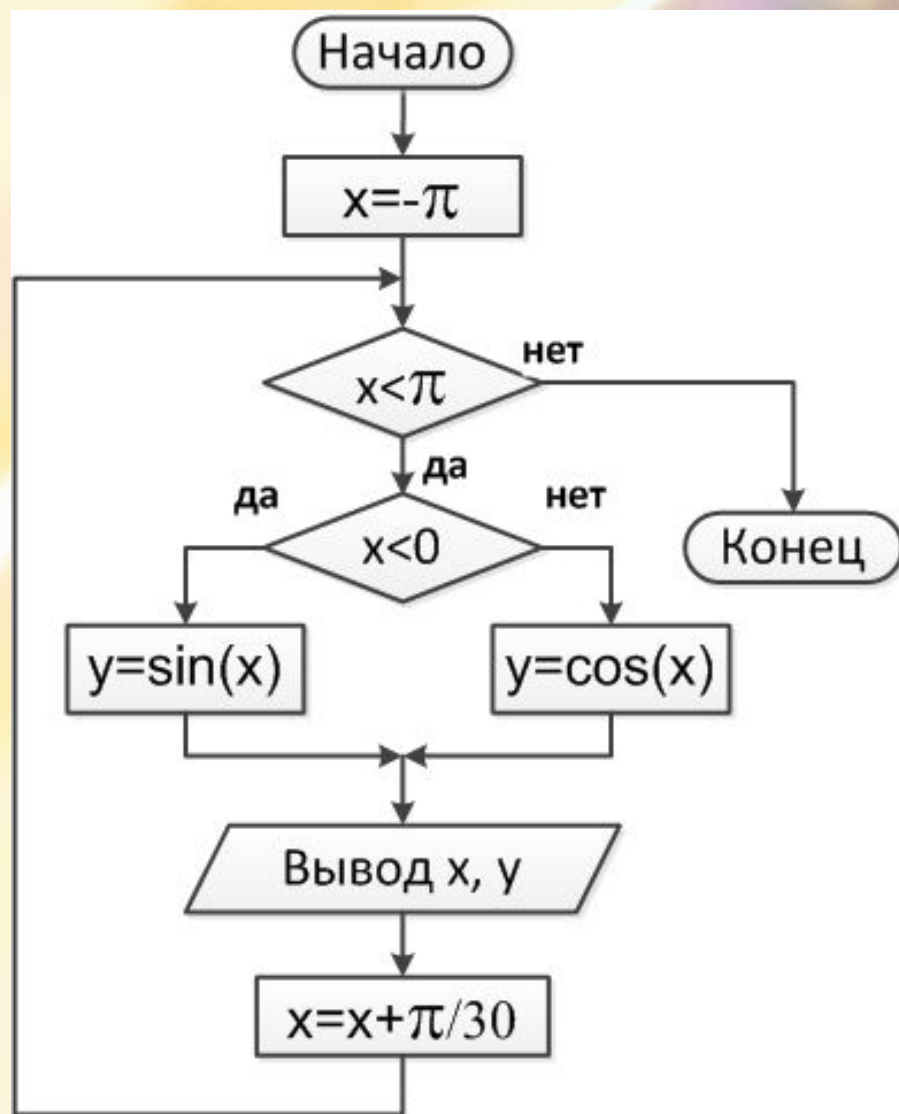
Задача 6:

Вычислить

$$y = \begin{cases} \sin(x), & \text{если } x < 0 \\ \cos(x), & \text{если } x \geq 0 \end{cases}$$

для значений $-\pi \leq x < \pi$

с шагом $\Delta x = \pi / 30$



ОПЕРАТОРЫ ПОВТОРА

Задание.

1. Вычислить сумму $S = \sum_{k=1}^{10} \sin(k \cdot x)$

k изменяется от 1 до 10 с шагом 1

2. Вычислить площадь пожара при изменении времени от 0 до 90 минуты с шагом в 5 минут. Скорость развития пожара $V=2$ м/мин

$$S = \begin{cases} \pi \cdot (0.5 \cdot V \cdot t)^2 & \text{если } t \leq 10 \\ \pi \cdot (5 \cdot V + V \cdot t_2)^2 & \text{если } t > 10 \end{cases} \quad t_2 = t - 10$$

