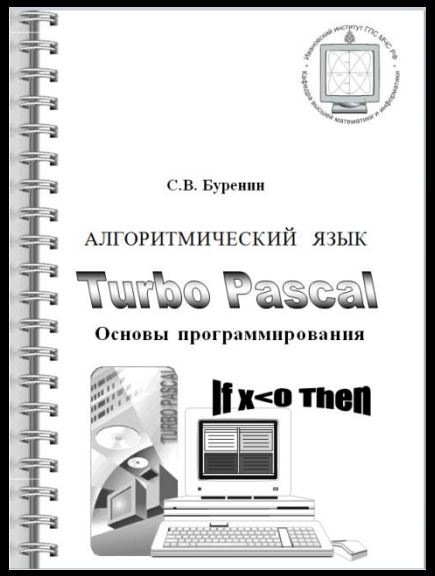
Тема 4.5 Основы программирования на языке Паскаль

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

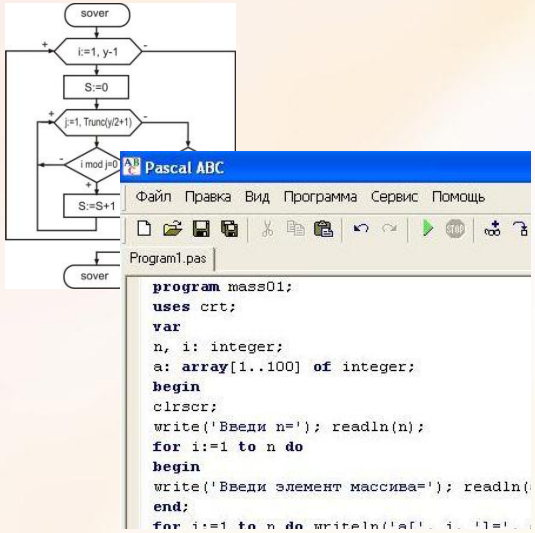
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Системное — операционные системы, драйверы, утилиты, сервисные программы

Прикладное — программы, выполняющие прикладные задачи

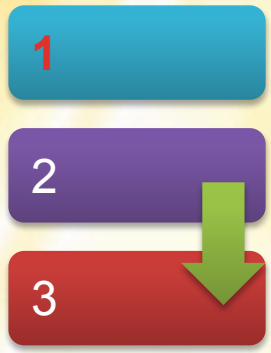
Инструментальное — алгоритмические языки и системы программирования

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



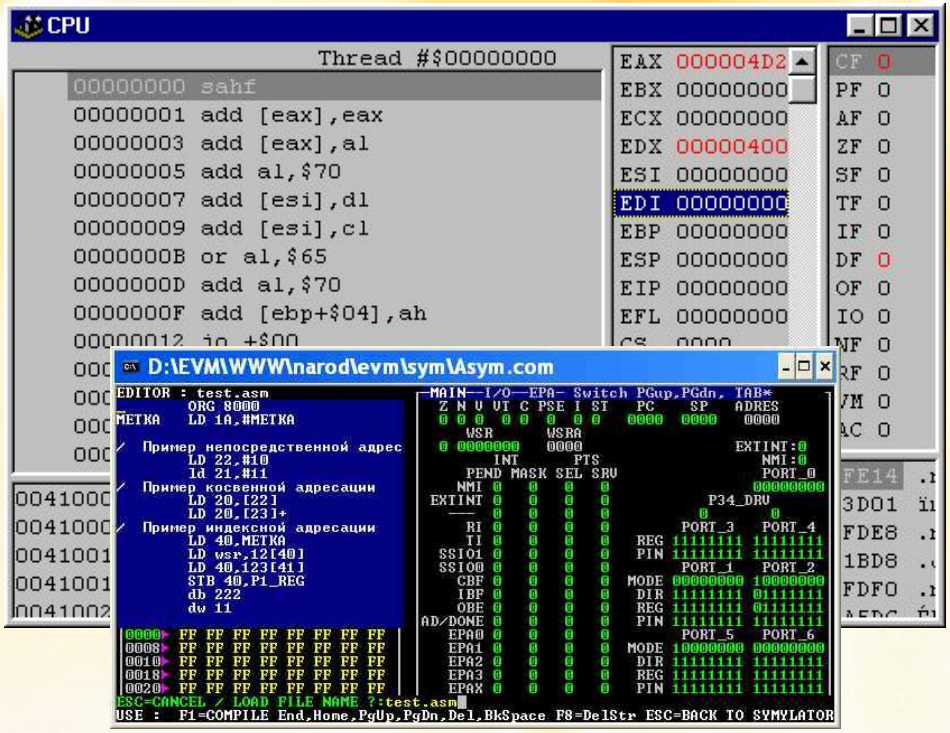
ЯЗЫКИ ПРОГРАММИРОВАНИЯ

машинные

ассемблера

высокого уровня

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



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.)

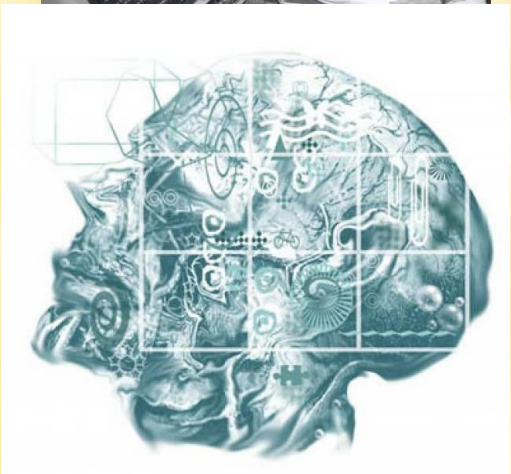
Алгол (1958 - 960) **– разработка программ для решения научно-технических задач на ЭВМ**

Бейсик (1963-1964) **– язык для обучения программированию**

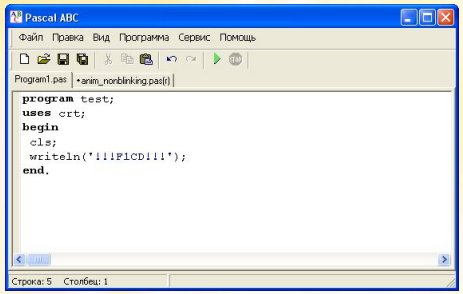
ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

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**

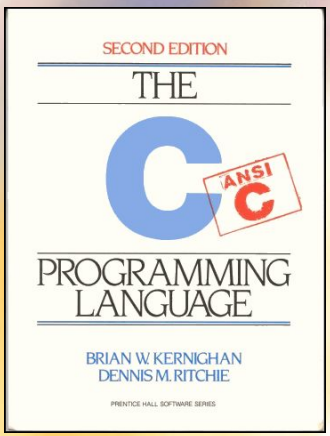


Паскаль (1968-1969) **– универсальный язык программирования**
(обучение программированию в вузах, промышленное программирование, написание больших и сложных программ)



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**

```
      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
             WRITE(6,9) I,Y
 9           FORMAT(I10. F12.6)
         ELSE
             WRITE (6,5) I
 5           FORMAT(I10,' TOO LARGE'
         ENDIF
10    CONTINUE
      END
```

Фортран (**For**mula **Tran**slation, 1957) **– язык для решения математических задач**

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

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

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

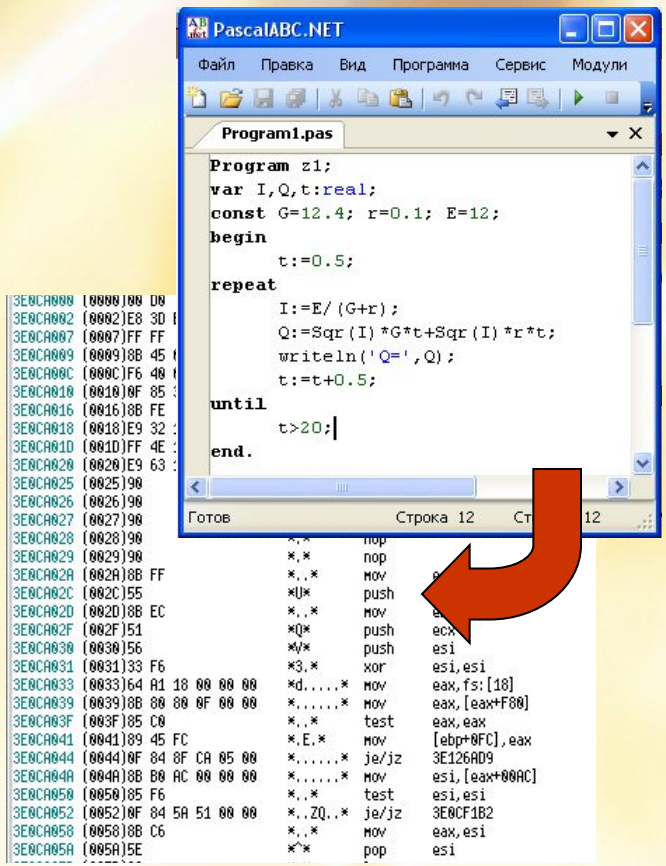
VBA – языки, применяемые в различных офисных программах

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

Трансляторы реализуются в виде компиляторов или интерпретаторов.

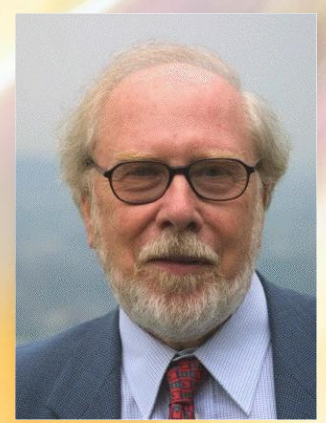
Компилятор транслирует всю программу целиком и создает вариант программы на машинном языке.

Интерпретатор переводит и выполняет программу строку за строкой.



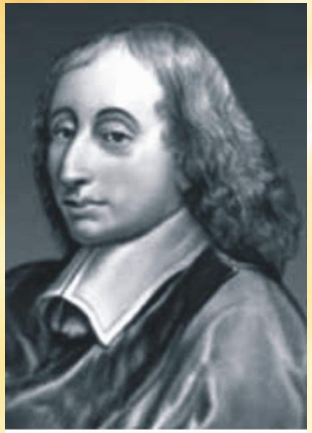
2. Структура программы. Описание данных

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* }
(* *а можно и так записать* *)

Идентификаторы – это любые имена, задаваемые пользователем в программе для обозначения меток, констант, переменных, процедур и функций.

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 – символьный 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

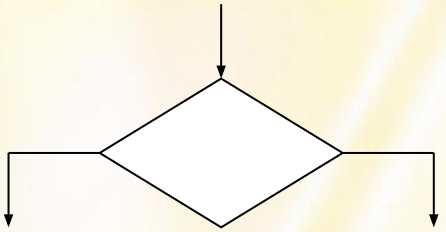
Операции отношения (сравнения)

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)

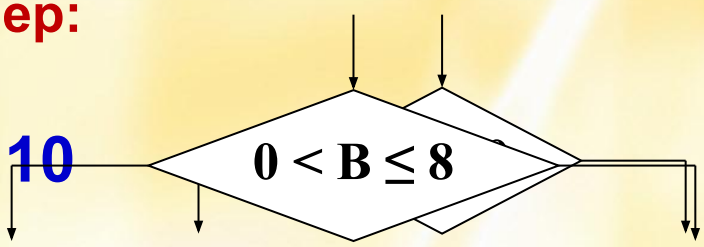
Операция	Знак
Равно	=
Не равно	< >
Меньше	<
Больше	>
Меньше или равно	<=
Больше или равно	>=

Логические выражения

Операция	Выражение
И	and
ИЛИ	or
отрицание	not

Пример:

x <= 10



(0<B) and (B<=8)

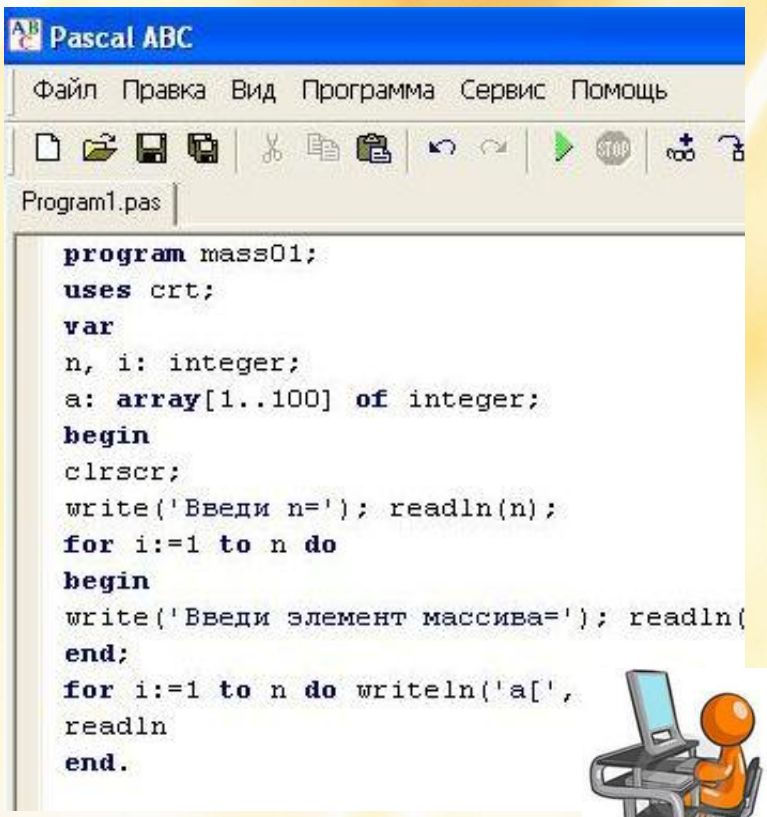
4. Операторы ввода-вывода, структурные операторы

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**



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

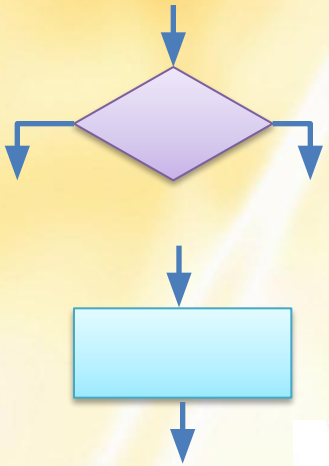
Простые операторы

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,…,x_n);**

ReadLn(x_1,x_2,…,x_n);

x_1…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,…,y_n);**

WriteLn(y_1,y_2,…,y_n);

y_1…y_n – переменные, значения которых необходимо вывести на экран монитора

Пример: **Write(A+B-2); Writeln(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,…,y_n);**

WriteLn(y_1,y_2,…,y_n);

y_1…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,…,y_n);**

WriteLn(y_1,y_2,…,y_n);

y_1…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 : n_1 : n_2);

X – переменная, значение которой выводится на экран;

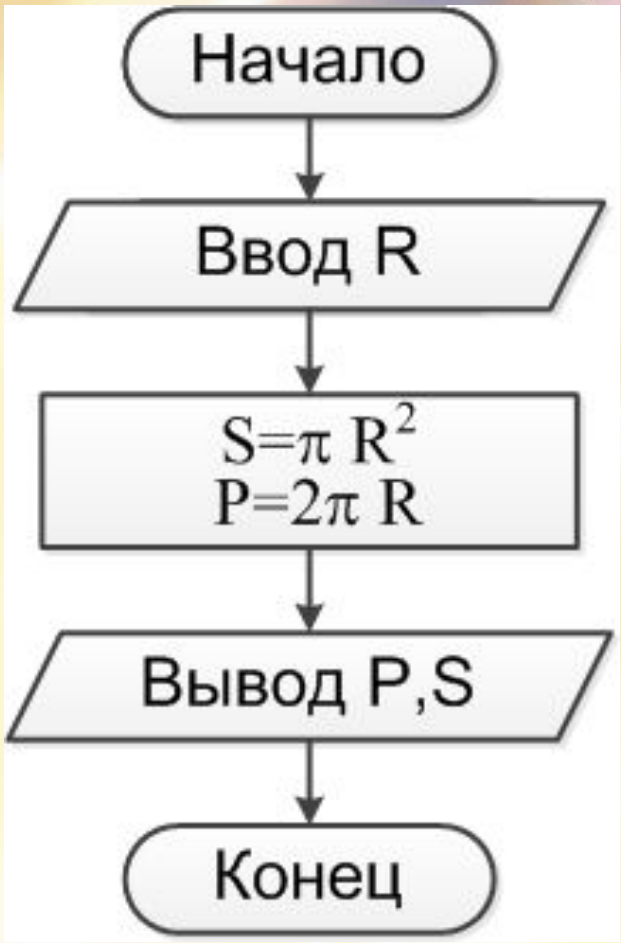
n_1 – число символов на экране для вывода значения переменной X;

n_2 – число символов после запятой (для дробных чисел типа **real**).

Пример:	x:=33.5391;	**Результат:**
	WriteLn(x);	3.3539100000E+01
	WriteLn(x:10);	3.3539E+01
	WriteLn(x:5:2);	33.54

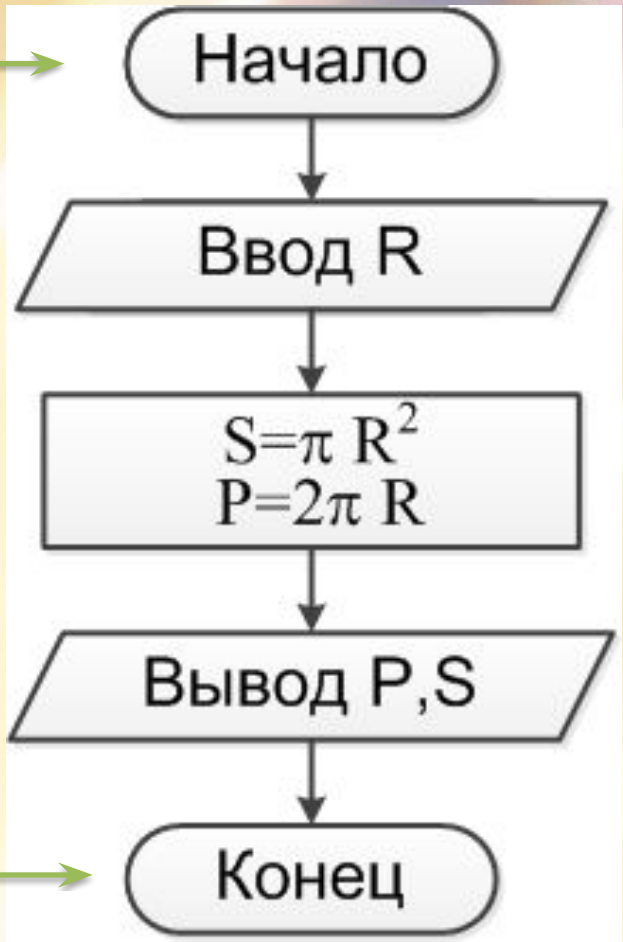
ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

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



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

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



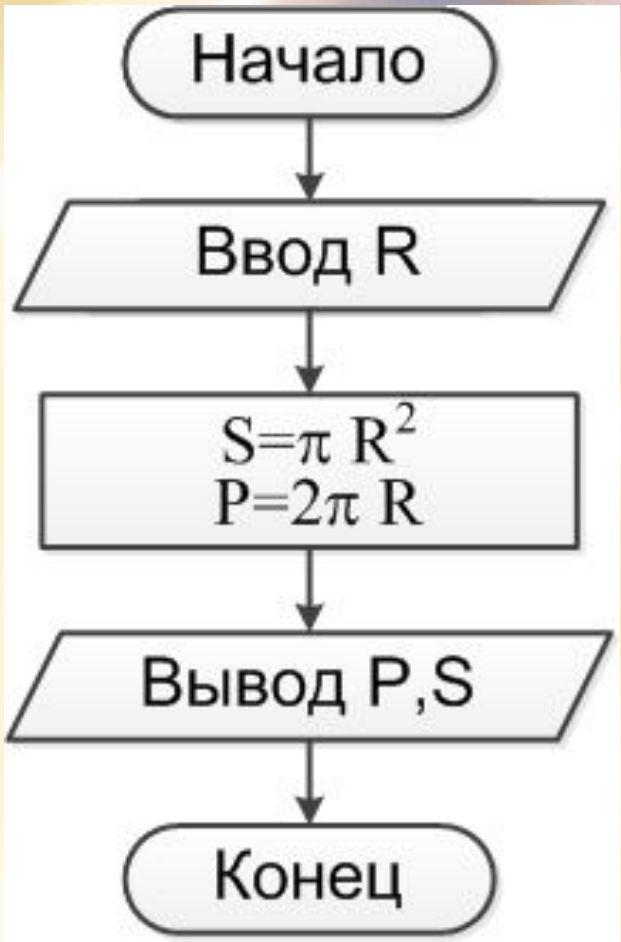
ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

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

Заголовок

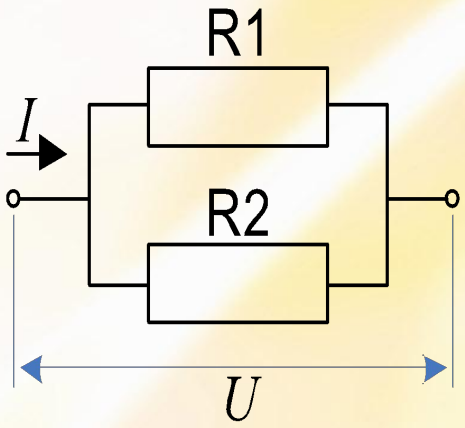
Блок описания данных

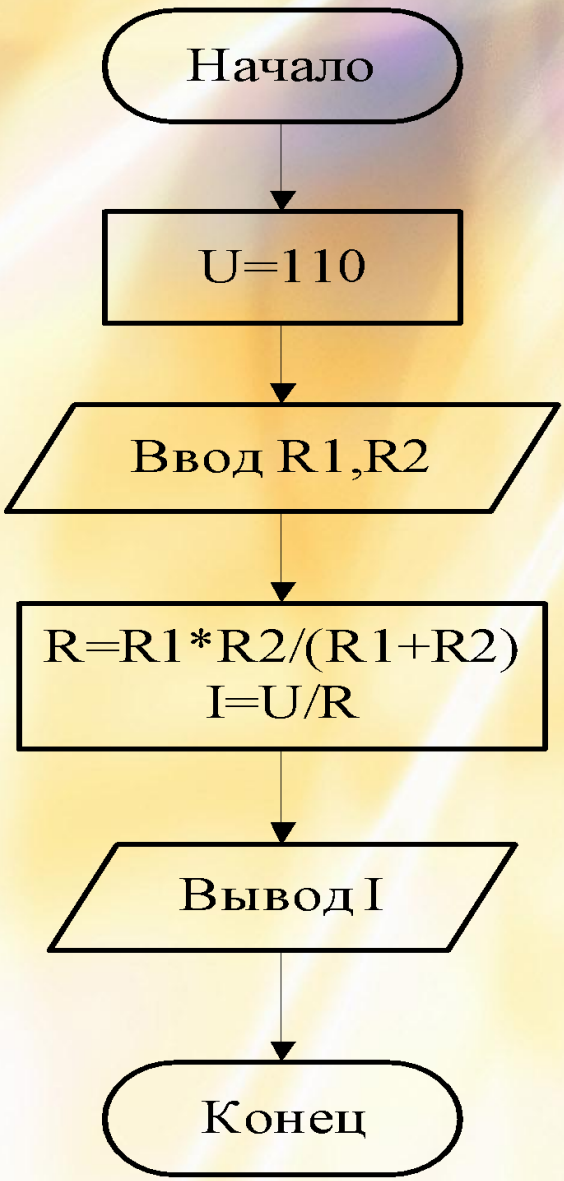
Тело программы



ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

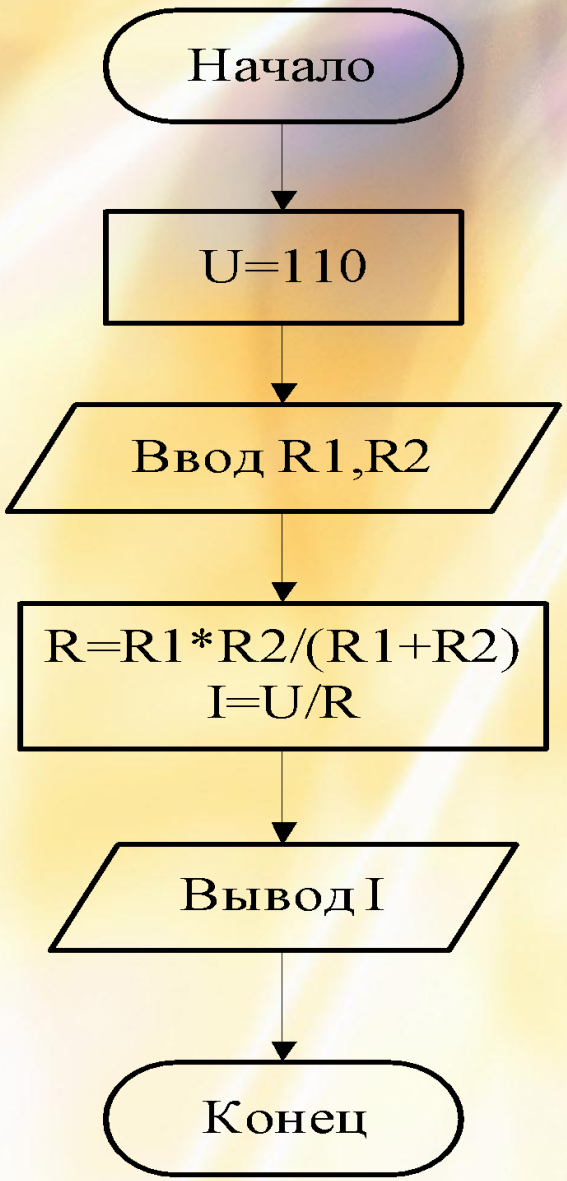
Задача 2: Дана электрическая схема, в которой U=110 В. Для произвольно заданных значений сопротивлений R1 и R2 вычислить ток I, проходящий через цепь.





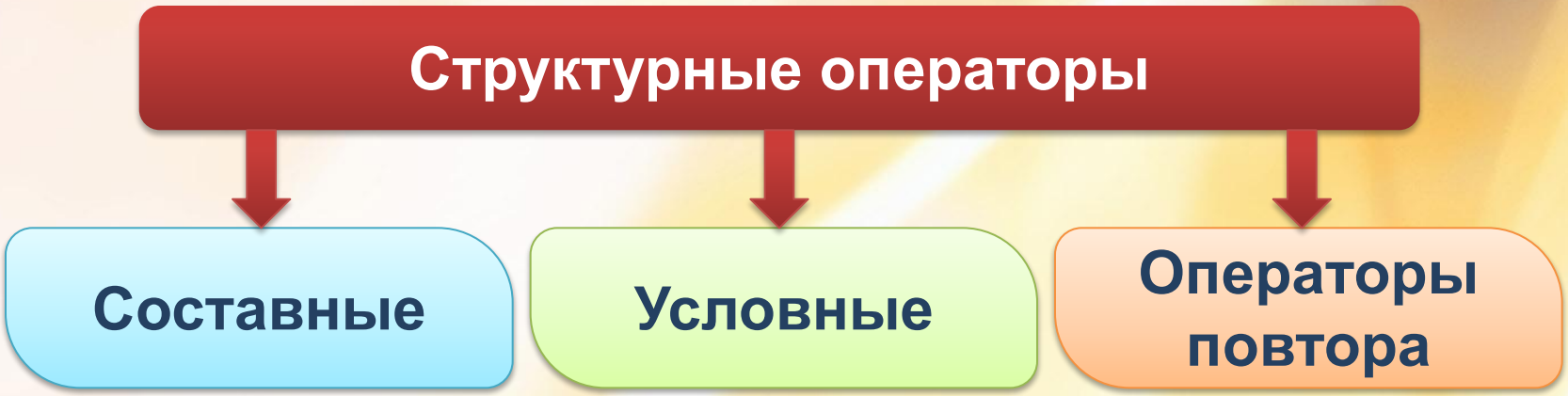
ОПЕРАТОРЫ В ЯЗЫКЕ ПАСКАЛЬ

```
Program Z2;
Const U=110;
Var R1, R2, R, I : Real;
Begin
        ReadIn(R1, R2);
        R:=R1*R2/(R1+R2);
        I:=U/R;
        WriteIn(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;
```

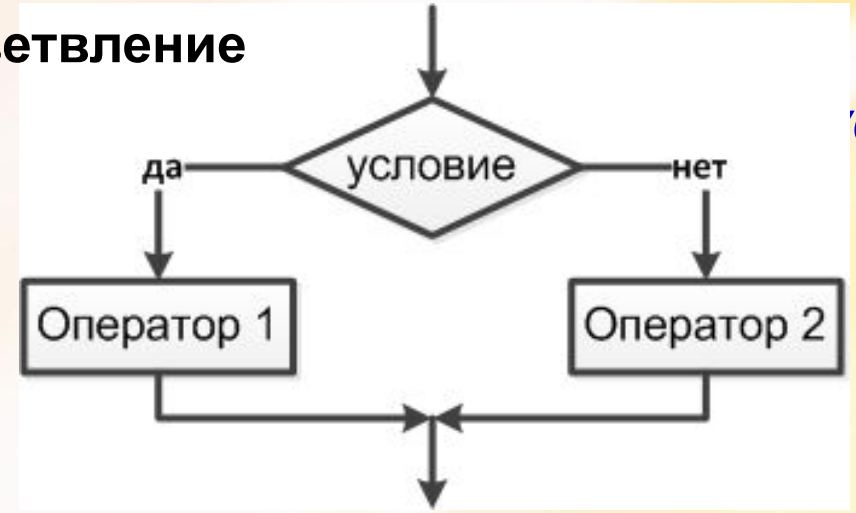
составной оператор

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***

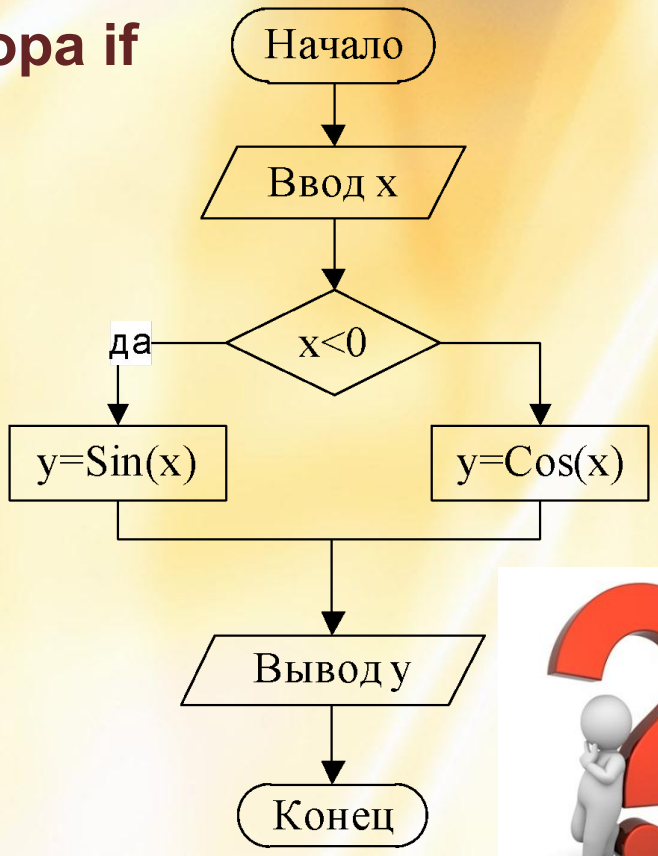
СТРУКТУРНЫЕ ОПЕРАТОРЫ

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

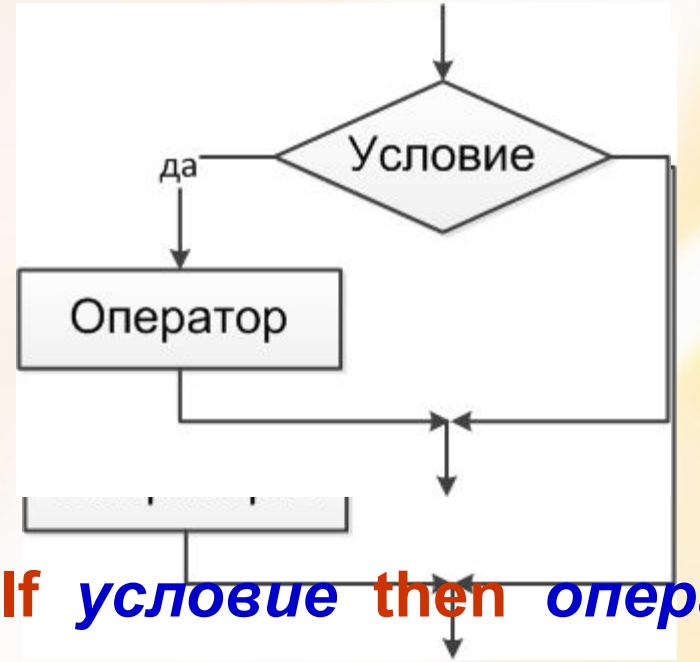
$$y = \begin{cases} \sin(x), & \text{если } x < 0 \\ \cos(x), & \text{если } x \geq 0 \end{cases}$$



СТРУКТУРНЫЕ ОПЕРАТОРЫ

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

Варианты записи условного оператора if



If ***условие*** **then** ***оператор***;

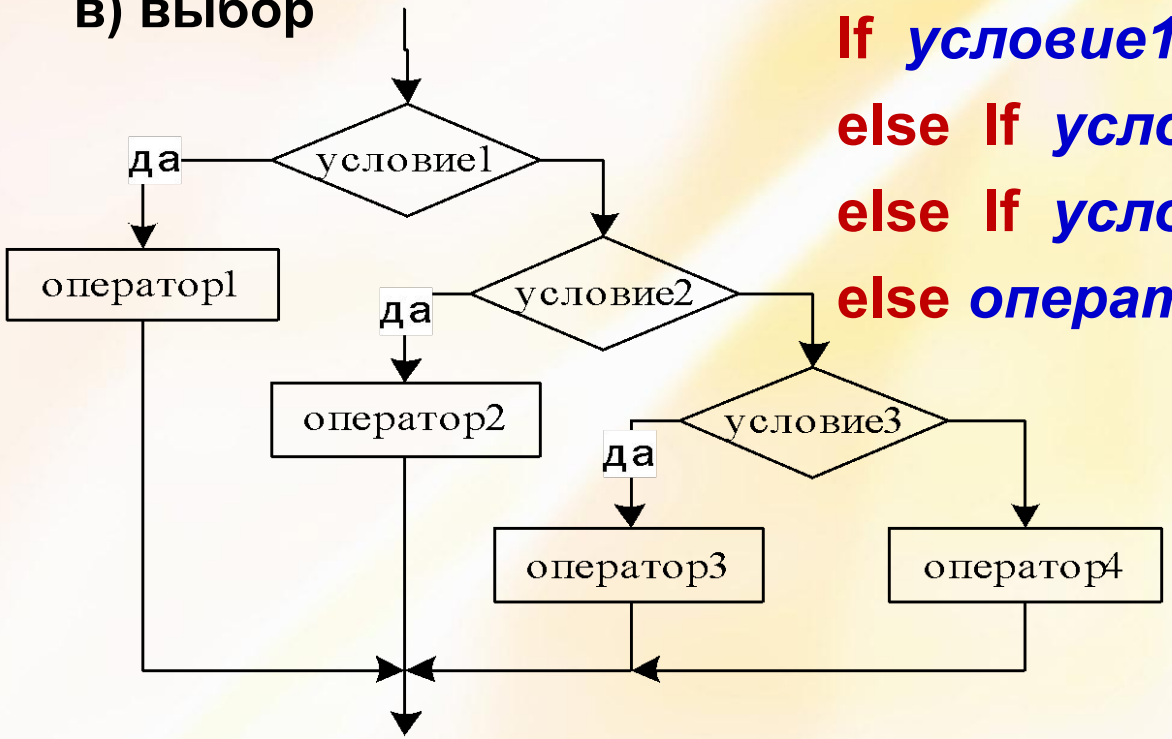
If ***условие*** **then**
Begin
оператор1;
оператор2
End;

СТРУКТУРНЫЕ ОПЕРАТОРЫ

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

Варианты записи условного оператора if

в) выбор



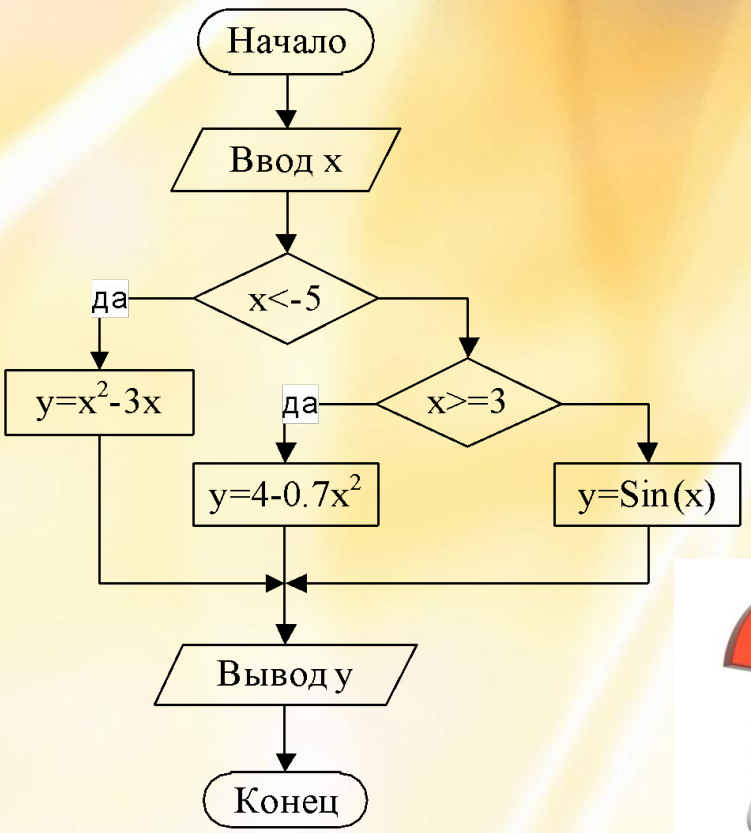
If ***условие1*** **then** ***оператор1***
else If ***условие2*** **then** ***оператор2***
else If ***условие3*** **then** ***оператор3***
else ***оператор4*****;**

СТРУКТУРНЫЕ ОПЕРАТОРЫ

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

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

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



СТАНДАРТНЫЕ ФУНКЦИИ ЯЗЫКА ПАСКАЛЬ

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

СТАНДАРТНЫЕ ФУНКЦИИ ЯЗЫКА ПАСКАЛЬ

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

Возведение в произвольную степень $y=x^a$

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

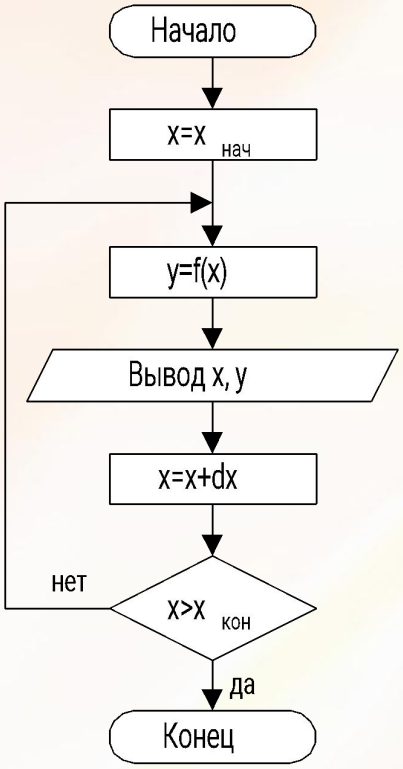
Вычисление логарифма $y=\log_a x$

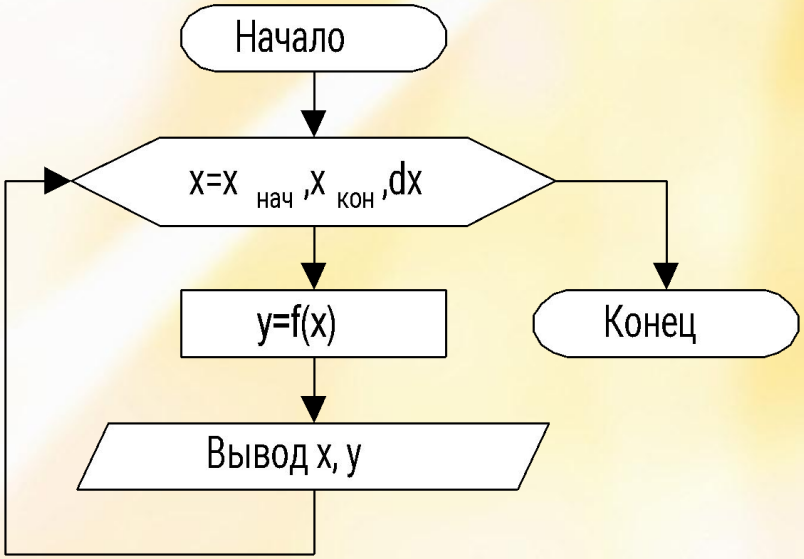
y:=ln(x)/ln(a)

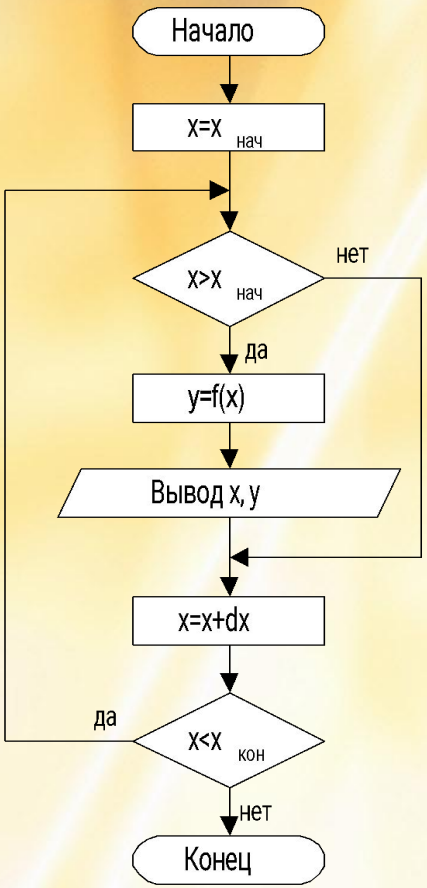
ОПЕРАТОРЫ ПОВТОРА

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 Δx=1)

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

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

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

ОПЕРАТОРЫ ПОВТОРА

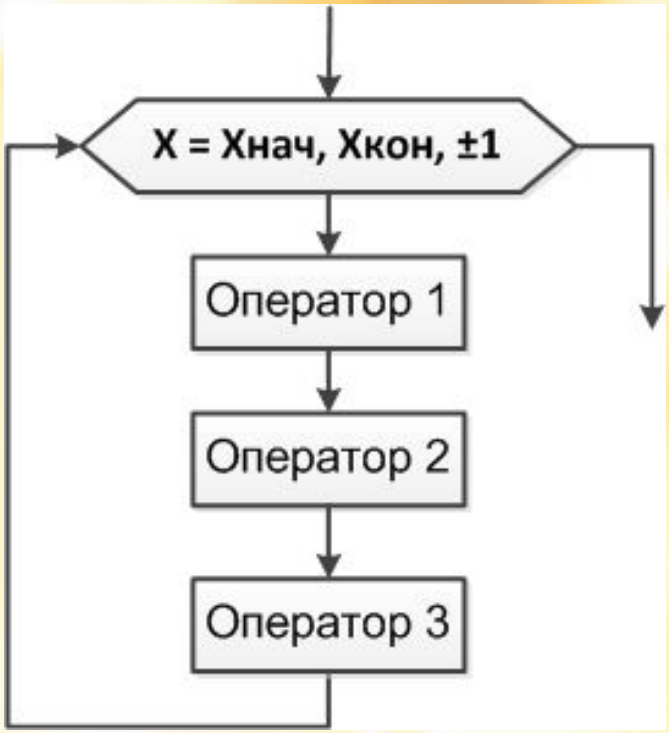
1) Оператор повтора For используется в циклах с шагом изменяемой переменной +1 или -1.

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

Описание:

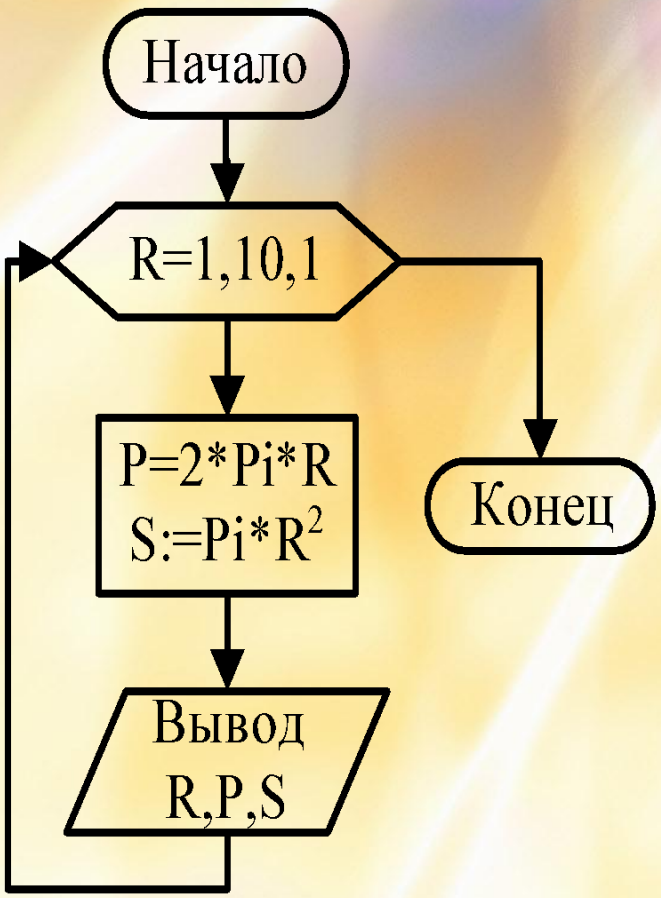
For $x:=x_{нач}$ **to** $x_{кон}$ **do**
begin
оператор1;
оператор2;
оператор3
end;

Составной оператор



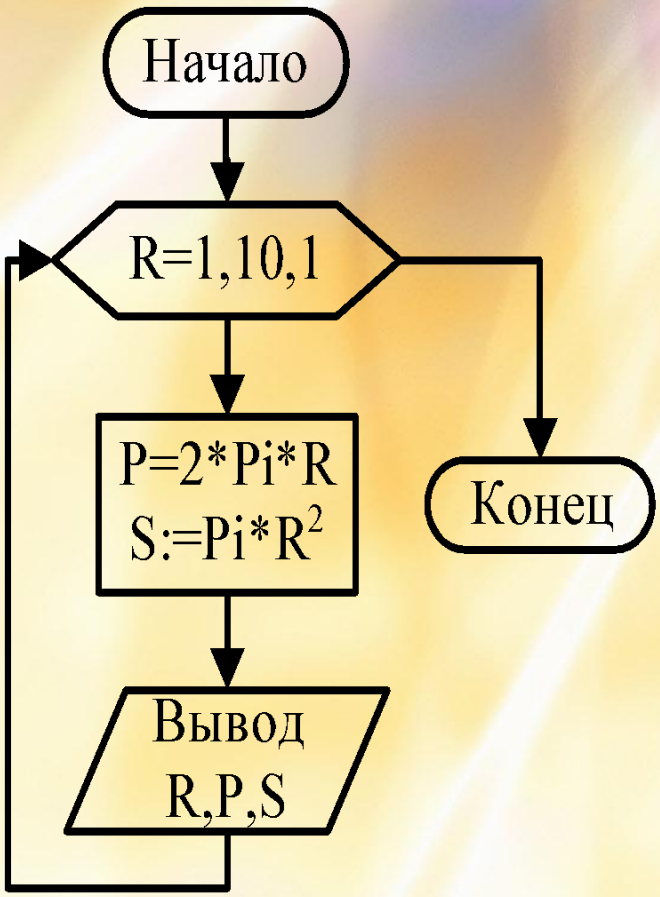
ОПЕРАТОРЫ ПОВТОРА

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



ОПЕРАТОРЫ ПОВТОРА

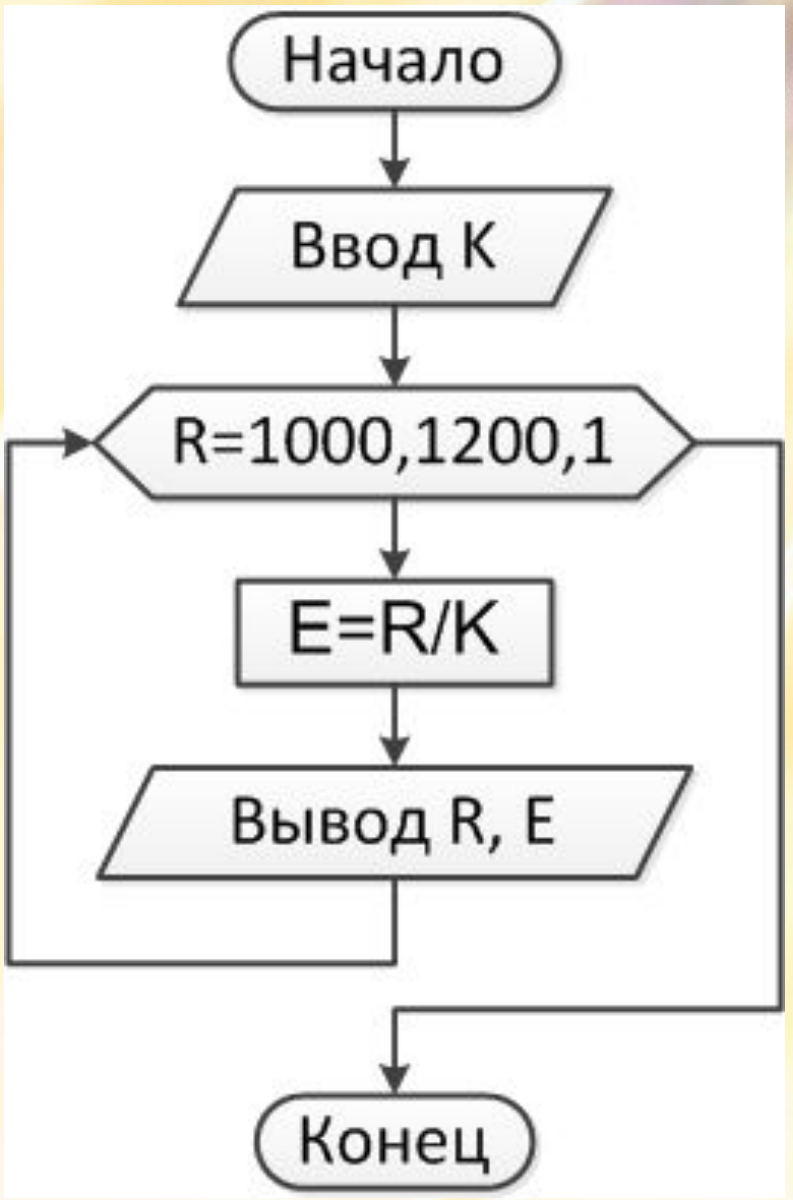
```
Program Krug;
Var R : Integer;
      P, S : Real;
Begin
    For R:=1 to 10 do
    Begin
            P:=2*Pi*R;
            S:=Pi*Sqr(R);
            Writeln(R, P, S);
    End;
End.
```



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

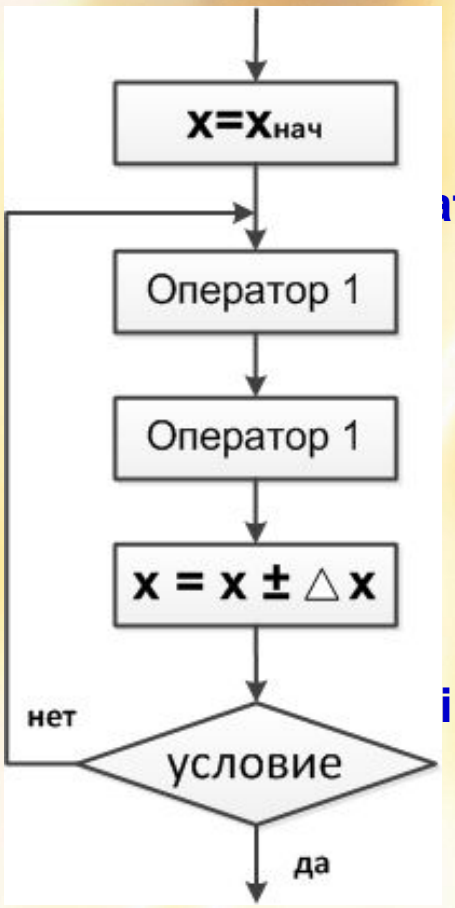
оператор1;

оператор2;

...

$x := x \pm \Delta x$

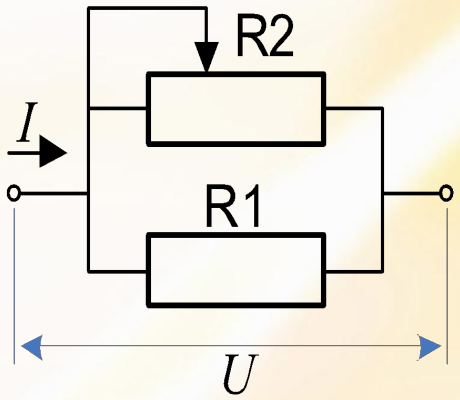
Until условие выхода из цикла;



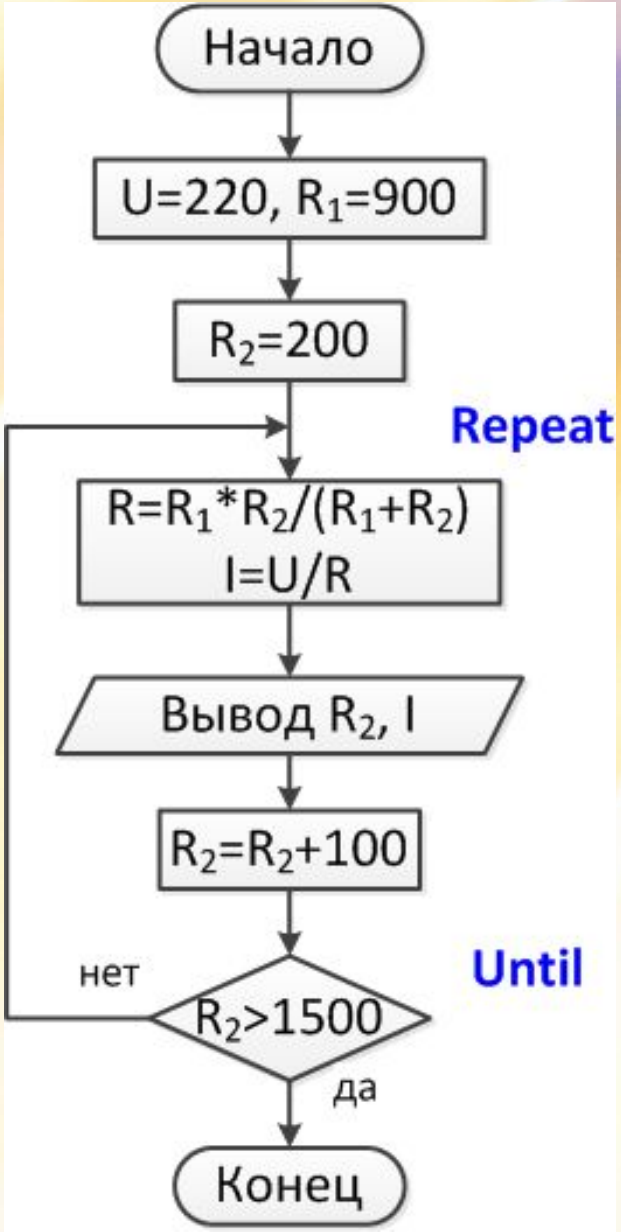
ОПЕРАТОРЫ ПОВТОРА

Задача 3:

Дана электрическая схема, в которой U=220 В, R1=900 Ом. Рассчитать значения тока в цепи при изменении сопротивления R2 от 200 до 1500 Ом с шагом 100 Ом.



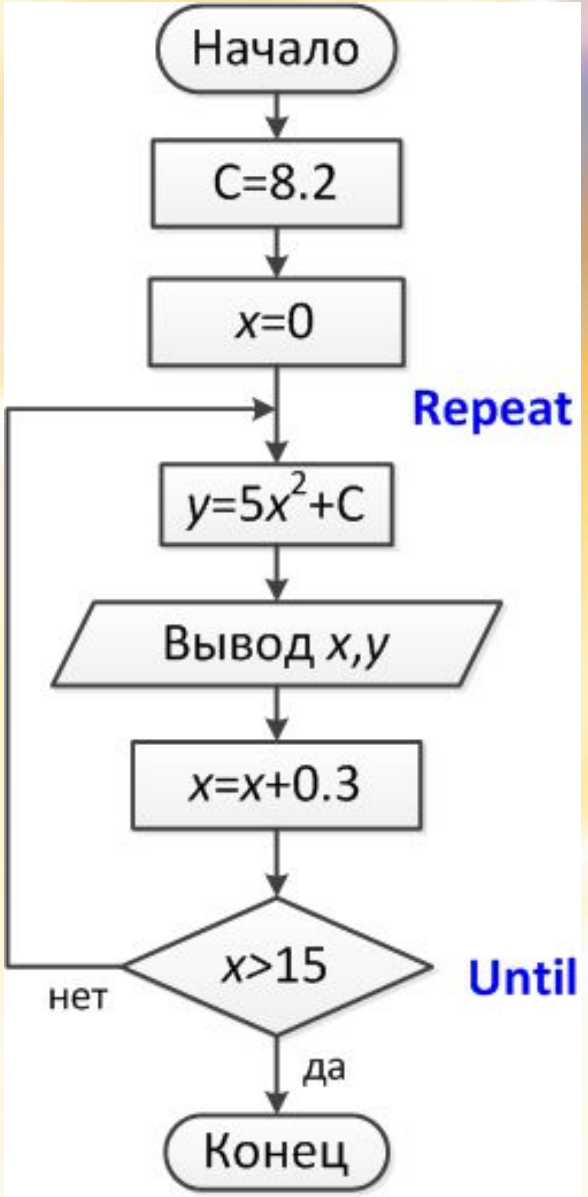
200 ≤ R2 ≤1500, ΔR2=100



ОПЕРАТОРЫ ПОВТОРА

Задача 4:

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



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

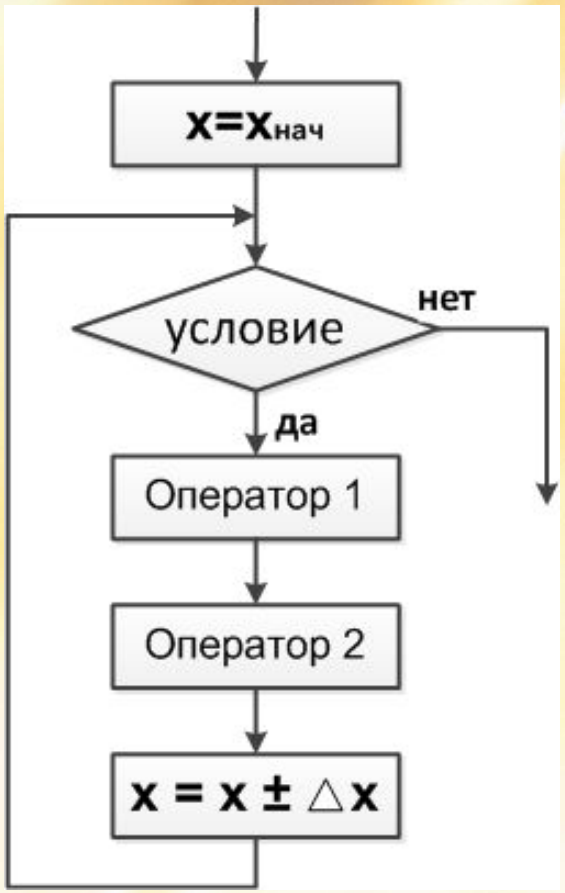
оператор1;

оператор2;

...

$x := x \pm \Delta x$

End;



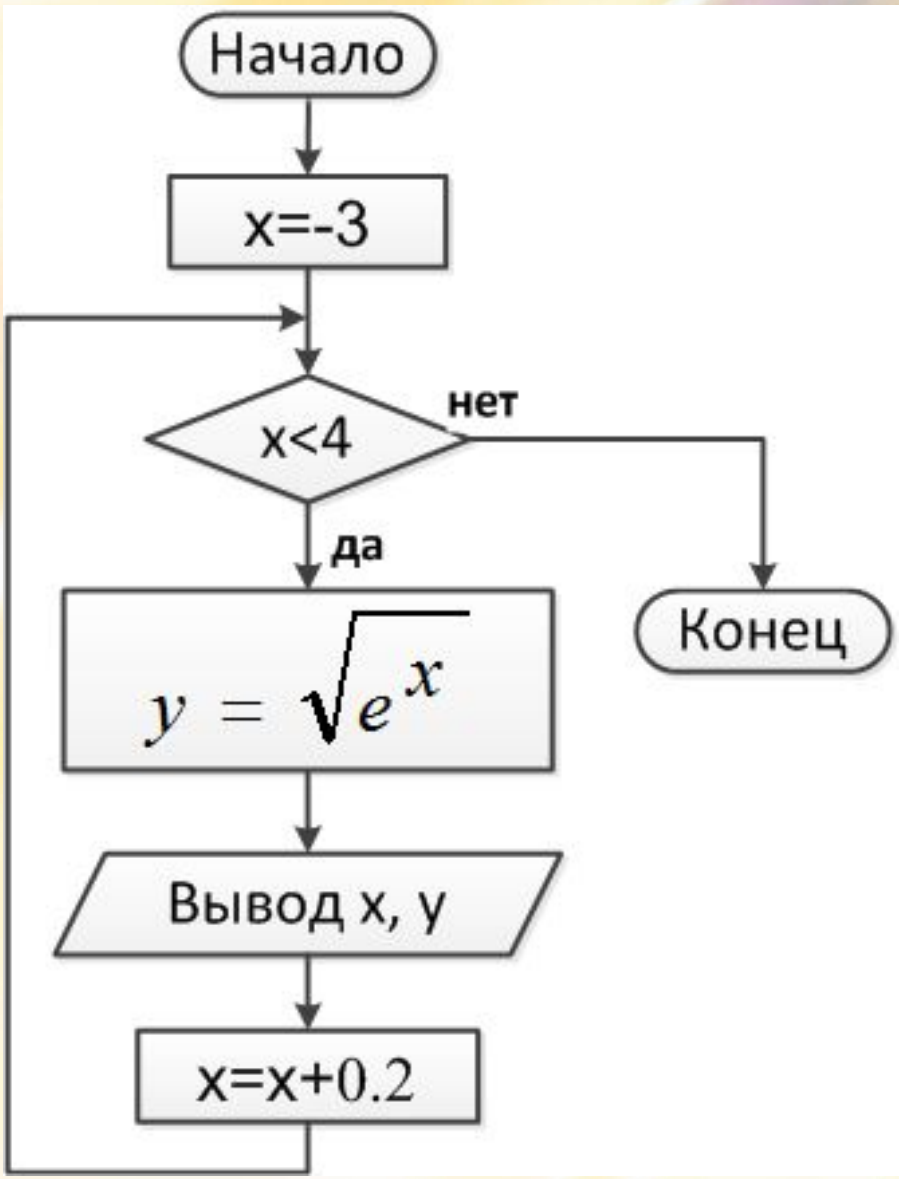
ОПЕРАТОРЫ ПОВТОРА

Задача 5:

Вычислить $y = \sqrt{e^x}$

для значений -3≤ x < 4

с шагом Δx= 0.2



ОПЕРАТОРЫ ПОВТОРА

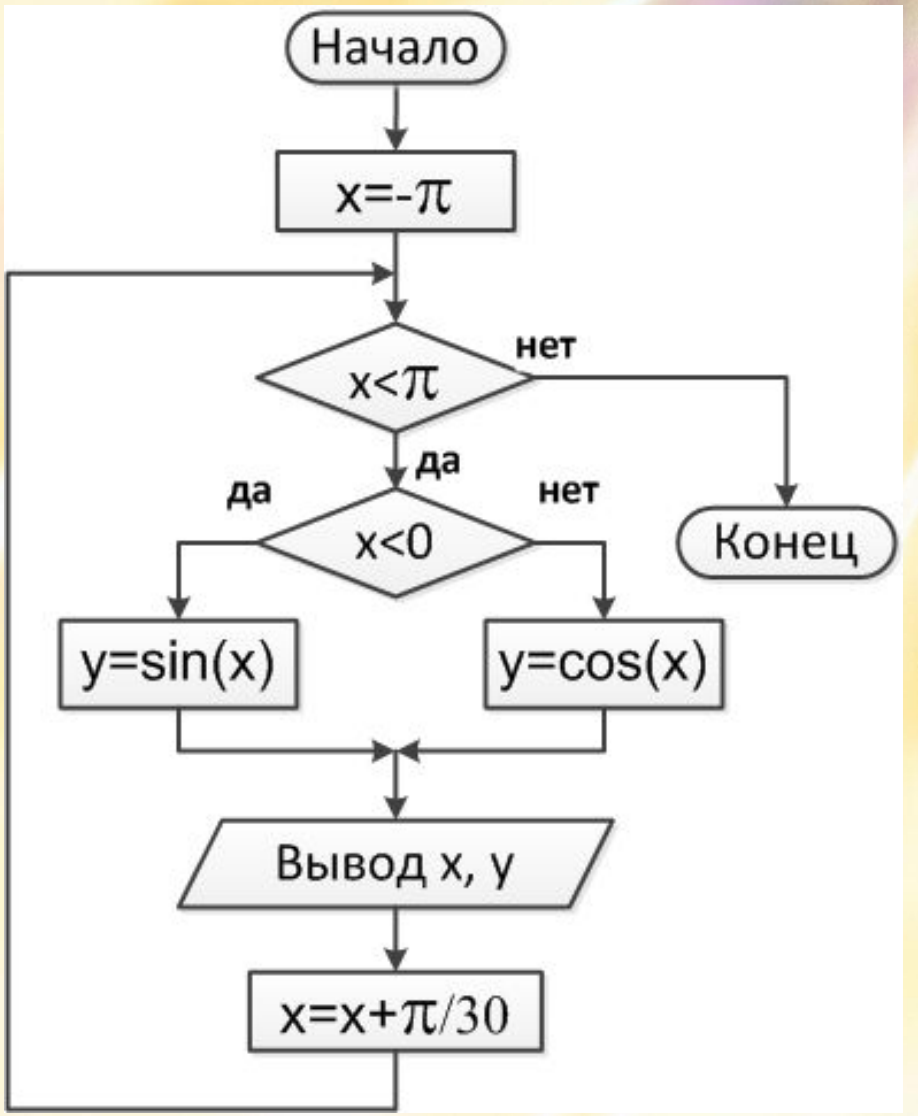
Задача 6:

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

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