

# Алгоритмы и способы их описания

---

Основы алгоритмизации и программирования

# Алгоритм

---

## Алгоритм это-

- a) протокол вычислительной сети
- b) четко определенный план действий для исполнителя
- c) правила выполнения определенных действий

## Свойства алгоритма

**дискретность**: состоит из отдельных шагов (команд)

**понятность**: должен включать только команды, известные исполнителю (входящие в СКИ)

**определенность**: при одинаковых исходных данных всегда выдает один и тот же результат

**конечность**: заканчивается за конечное число шагов

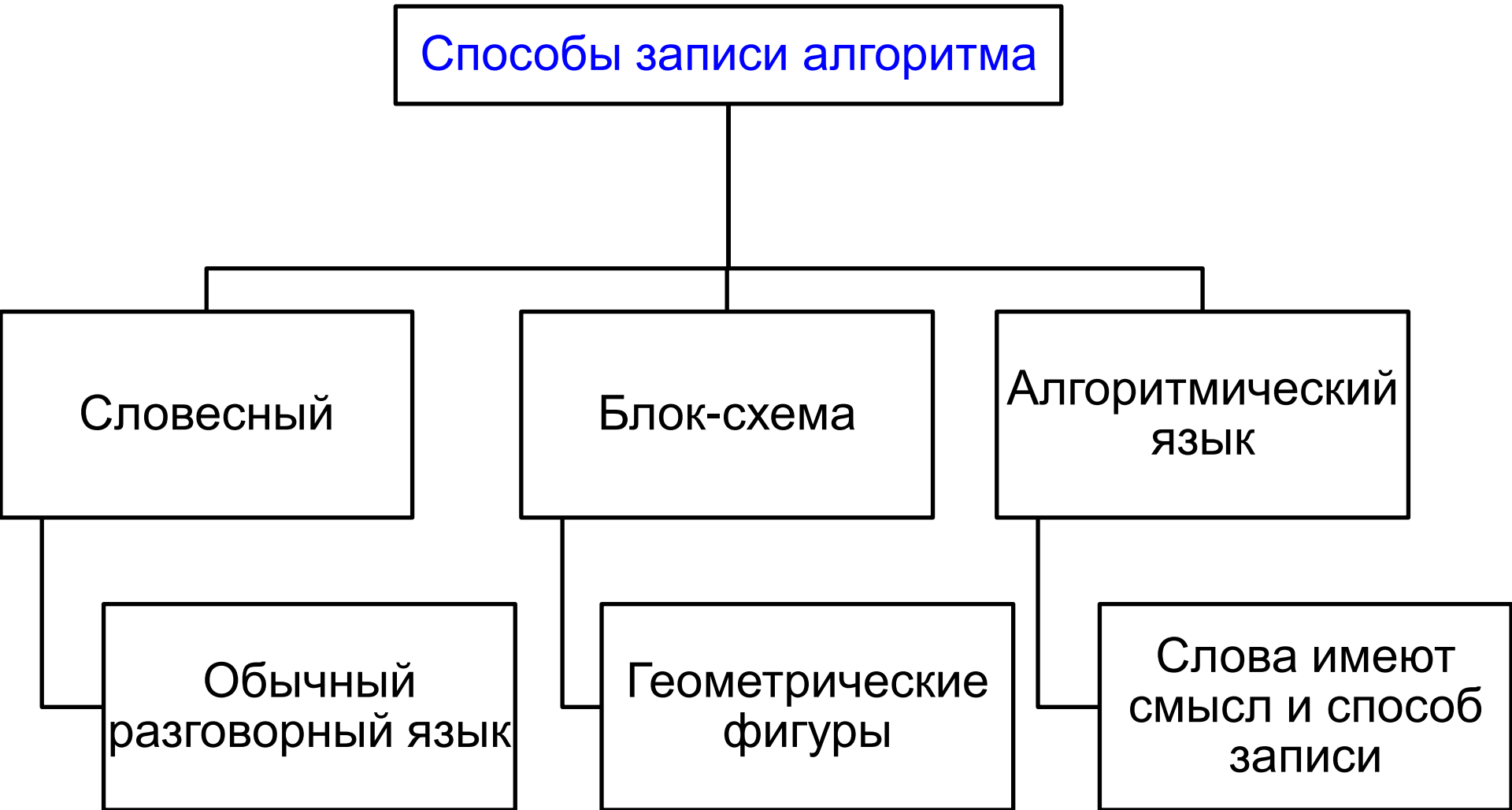
**массовость**: может применяться многократно при различных исходных данных

**корректность**: дает верное решение при любых допустимых исходных данных

# Исполнитель алгоритма

---

Субъект или устройство, способные правильно интерпретировать описание алгоритма и выполнить содержащийся в нем перечень действий



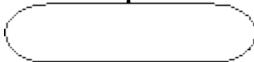

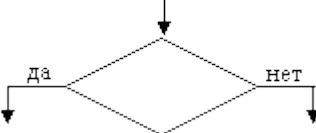
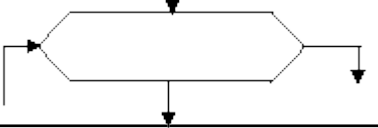
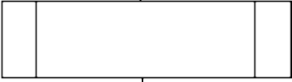
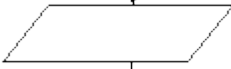
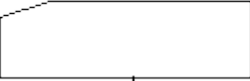

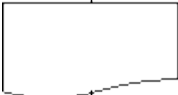
## Задание 1

Вы – студенты ЧТОТиБ. На 2 курсе, уже совсем скоро, вам предстоит трудоустройство на производственную практику.

### **Задание:**

1. Записать в словесной форме все шаги алгоритма «Трудоустройство на производственную практику».
2. Составить блок-схему к данному алгоритму.
3. Проанализируйте данное задание.
4. Ответьте на вопрос: Какие знания вам понадобятся для правильного выполнения данного задания?
5. Какими источниками информации можно воспользоваться для повторения необходимых знаний?

# Обозначение Блок-схем

наименование	обозначение	функции
Пуск-останов (начало и конец алгоритма)		Начало, конец, прерывание процесса обработки данных или выполнения алгоритма
Процесс (вычисление)		Выполнение операции или группы операций, в результате которых изменяется значение, форма представления или расположение данных.
Условие (выбор)		Выбор направления выполнения алгоритма в зависимости от некоторых переменных условий
Модификация		Выполнение операций, меняющих команды, или группы команд, изменяющих алгоритм
Предопределенный процесс (подпрограмма)		Использование ранее созданных и отдельно описанных алгоритмов.
Ввод-вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).
Перфокарта		Ввод-вывод данных, носителем которых является перфокарта
Дисплей		Ввод данных с подключенного к ЭВМ дисплея или вывод данных на дисплей.
Документ		Ввод-вывод данных, носителем которых служит бумага.

# Программа

---

**Программа** – это

- алгоритм, записанный на каком-либо языке программирования
- набор команд для компьютера

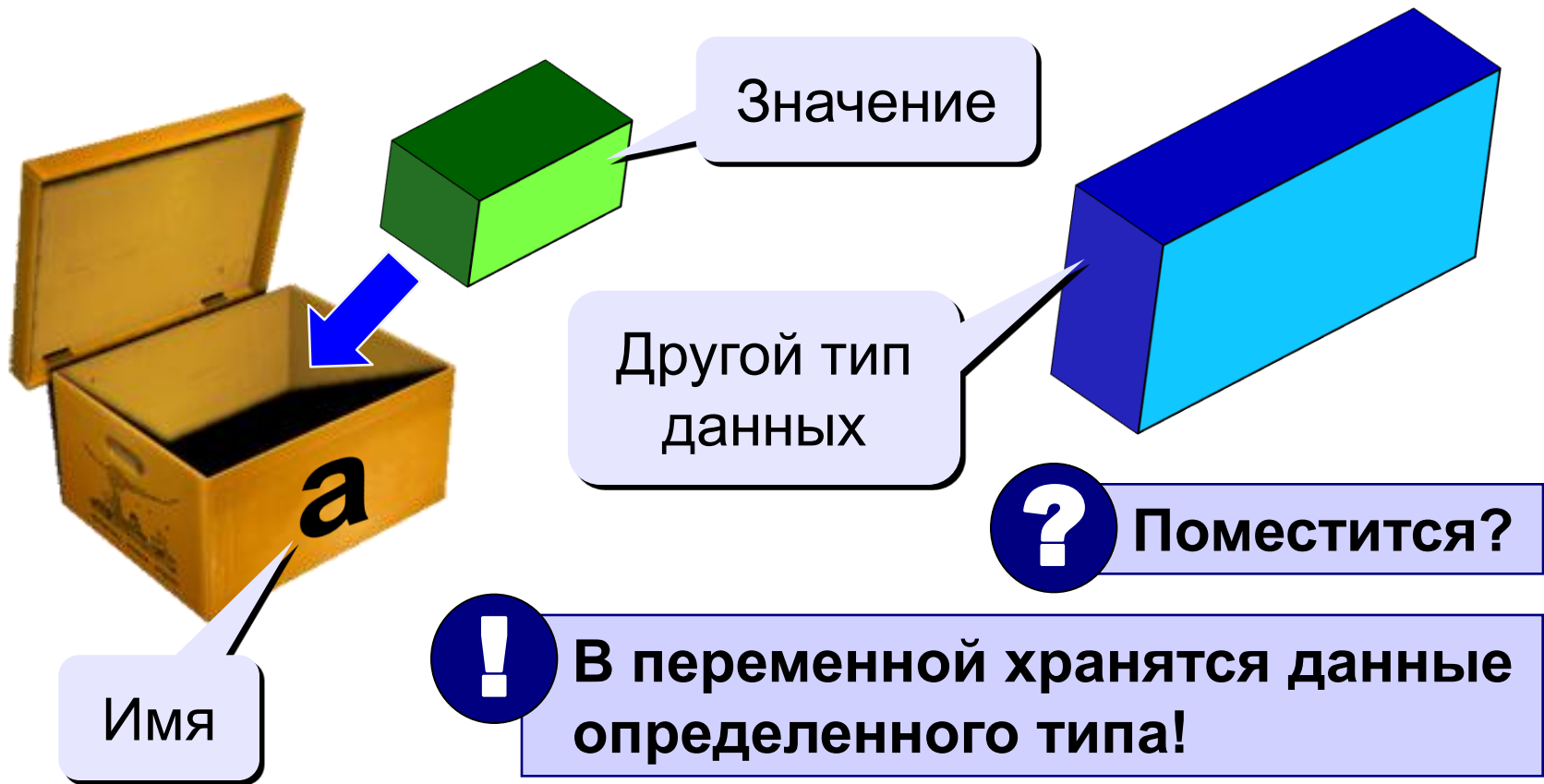
**Команда** – это описание действий, которые должен выполнить компьютер.

- откуда взять исходные данные?
- что нужно с ними сделать?

**Оператор** – это команда языка программирования высокого уровня.

# Переменные

**Переменная** – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



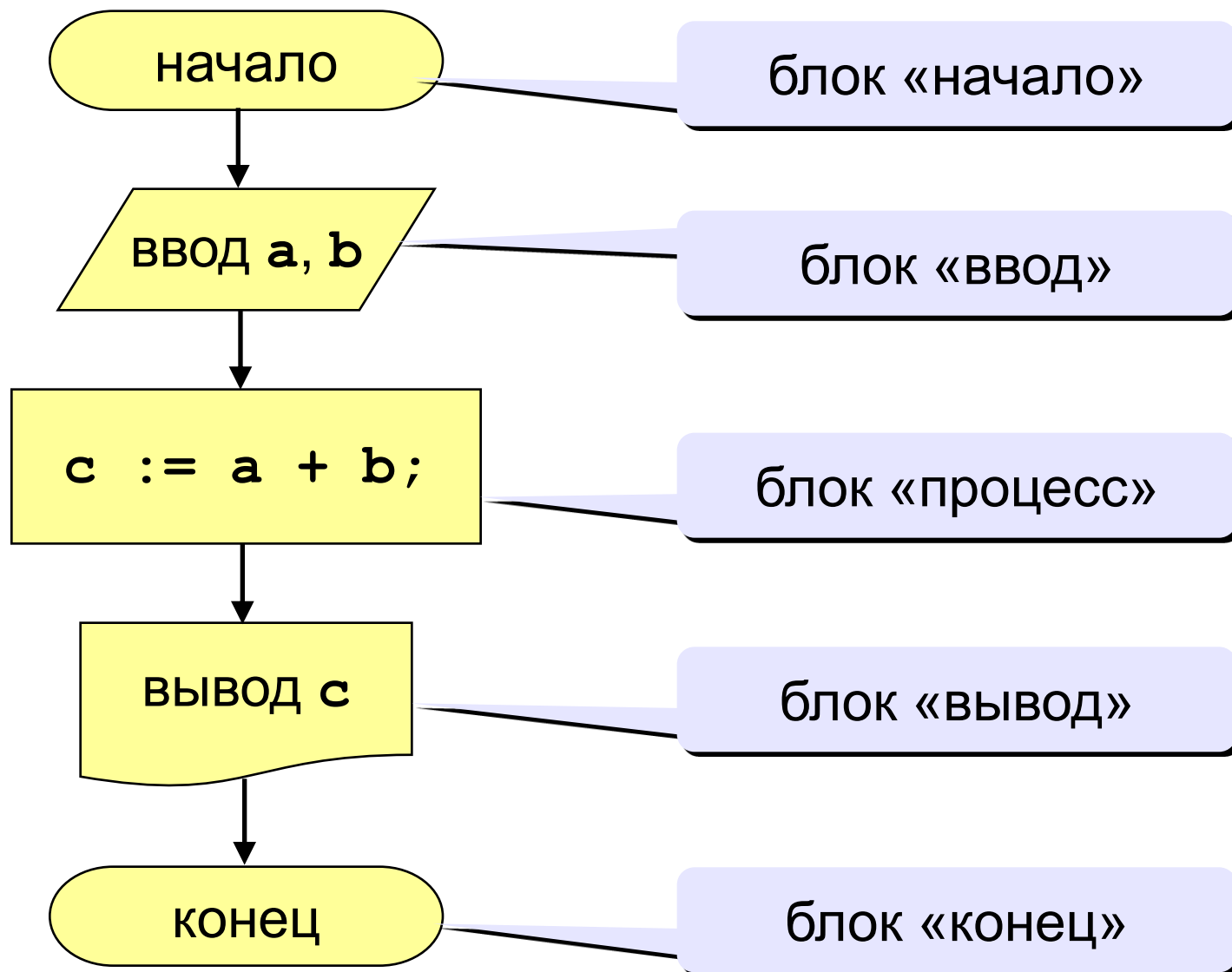


## Типы данных в языке программирования Паскаль

<i>Название типа</i>	<i>Описание</i>	<i>Диапазон значений</i>	<i>Размерность, байт</i>
<i>Byte</i>	<i>Целочисленный</i>	<i>0..255</i>	<i>1</i>
<i>Integer</i>		<i>-32768..32767</i>	<i>2</i>
<i>Longint</i>		<i>-2147483648..2147483647</i>	<i>4</i>
<i>Real</i>	<i>Вещественный</i>	<i><math>-2,9 \times 10^{-39} .. 1,7 \times 10^{38}</math></i>	<i>6</i>
<i>Double</i>		<i><math>-5 \times 10^{-324} .. 1,7 \times 10^{308}</math></i>	<i>10</i>
<i>Boolean</i>	<i>Логический</i>	<i>true, false</i>	<i>1</i>
<i>Char</i>	<i>Символьный</i>	<i>Все символы кода ASCII</i>	<i>1</i>



# Блок-схема линейного алгоритма



# Разветвляющиеся алгоритмы

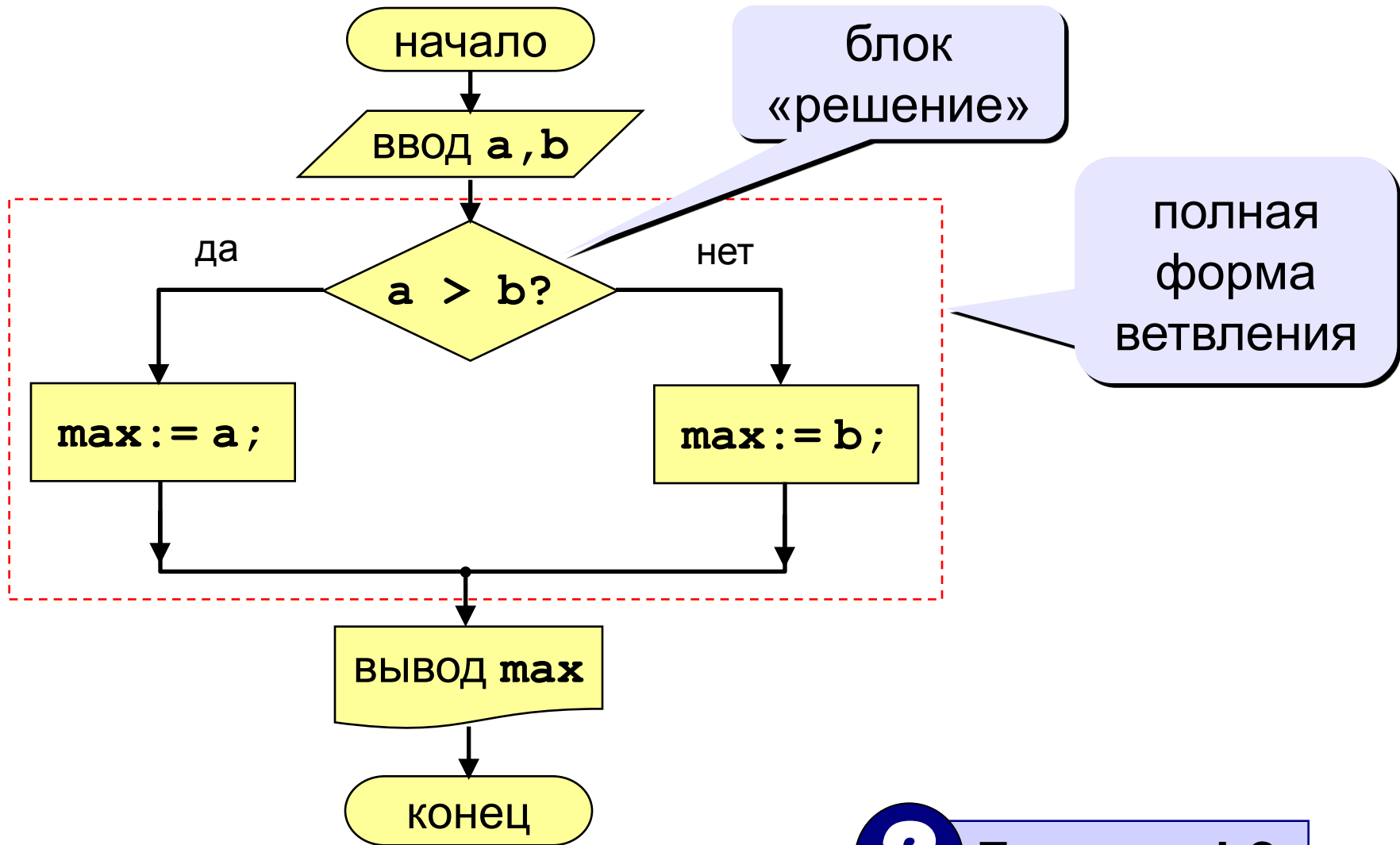
---

Алгоритмы, в которых последовательность шагов зависит от выполнения некоторых условий, называются **разветвляющимися**.

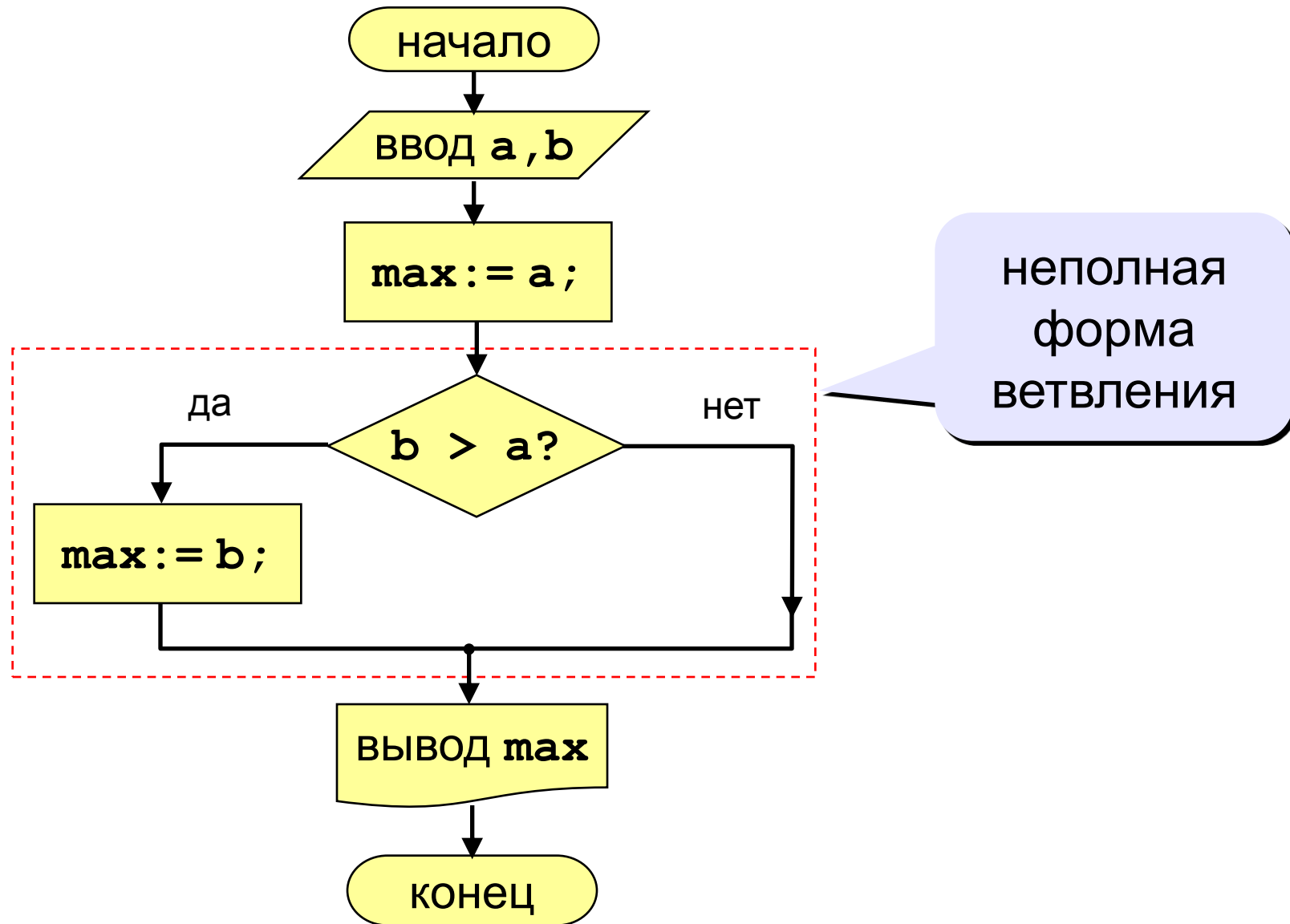
**Задача.** Ввести два целых числа и вывести на экран наибольшее из них.

исполнитель зависят от некоторых условий (***если ... иначе ...***).

# Вариант 1. Блок-схема



# Вариант 2. Блок-схема

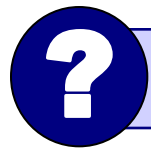


# Сложные условия

---

**Задача.** Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

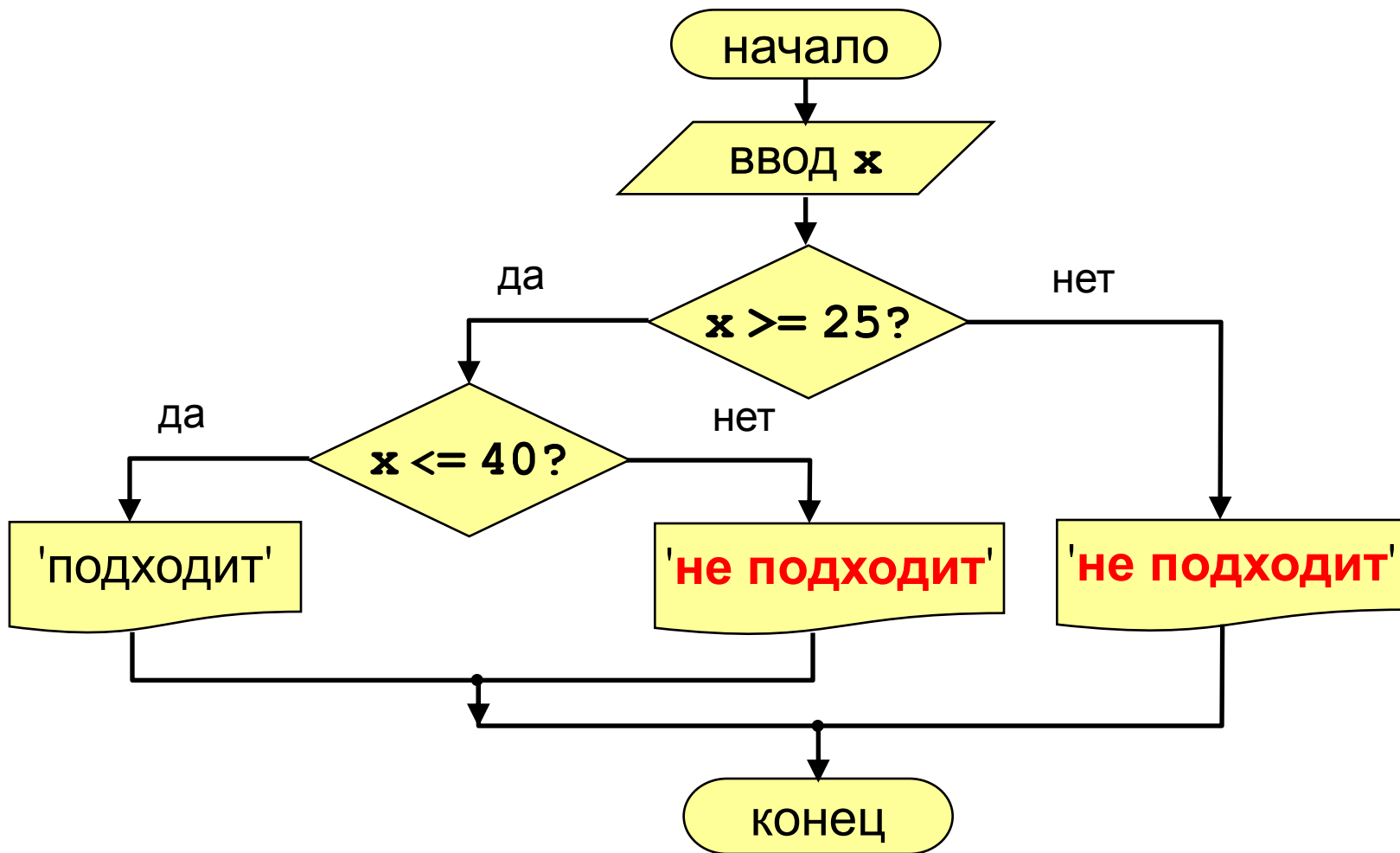
**Особенность:** надо проверить, выполняются ли два условия одновременно.



Можно ли решить известными методами?

# Вариант 1. Алгоритм

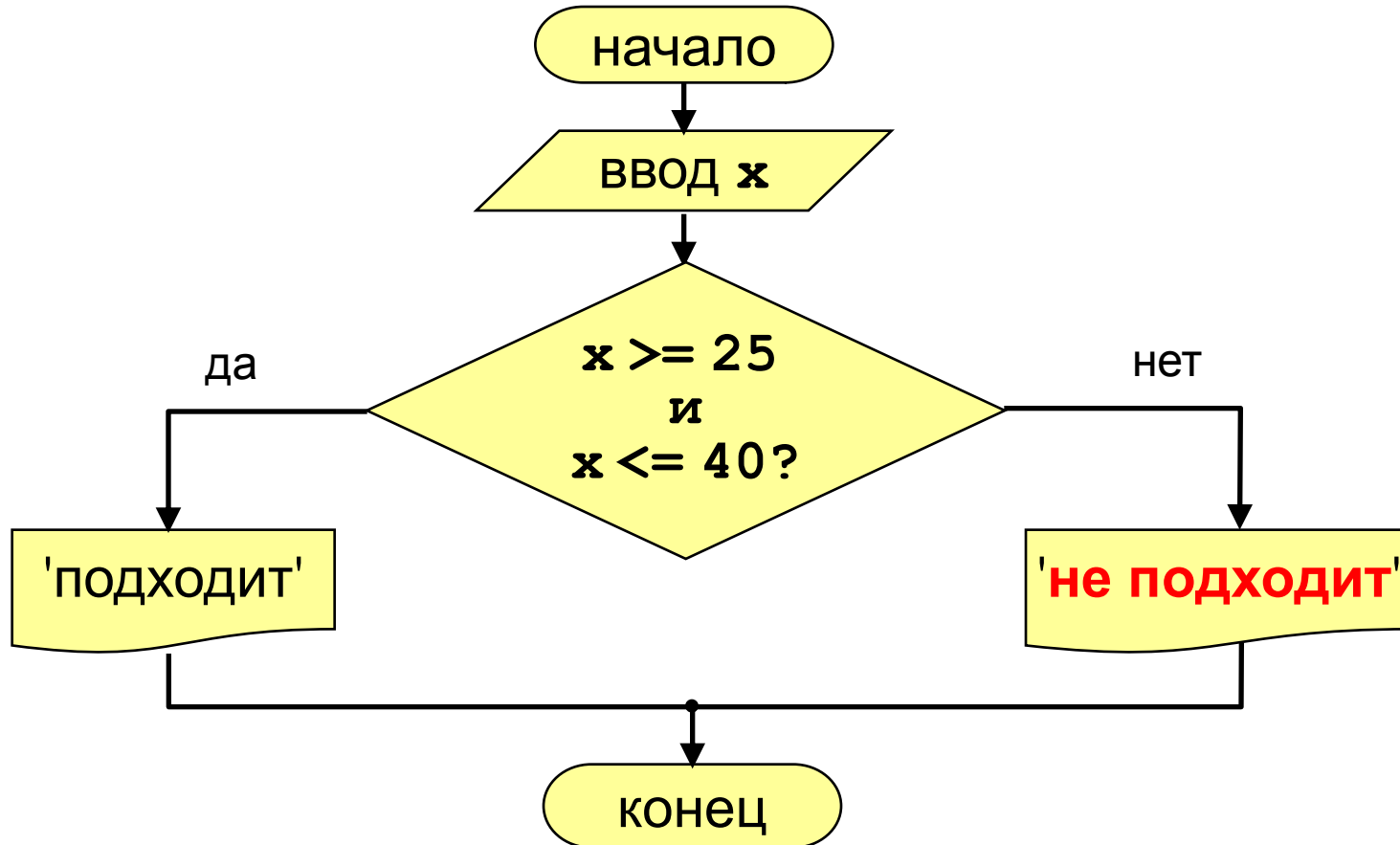
---





# Вариант 2. Алгоритм

---



# Сложные условия

---

## Простые условия (отношения)

<   <=   >   >=   =   <>

равно

не равно

**Сложное условие** – это условие, состоящее из нескольких простых условий (отношений), связанных с помощью **логических операций**:

- **not** – НЕ (отрицание, инверсия)
- **and** – И (логическое умножение, конъюнкция, одновременное выполнение условий)
- **or** – ИЛИ (логическое сложение, дизъюнкция, выполнение хотя бы одного из условий)
- **xor** – исключающее ИЛИ (выполнение только одного из двух условий, но не обоих)

# Циклы

---

**Цикл** – это многократное выполнение одинаковой последовательности действий.

- цикл с **известным** числом шагов (цикл с параметром)
- цикл с **неизвестным** числом шагов (цикл с условием)

Циклы по условию бывают двух основных видов: **циклы с предусловием** и **циклы с постусловием**

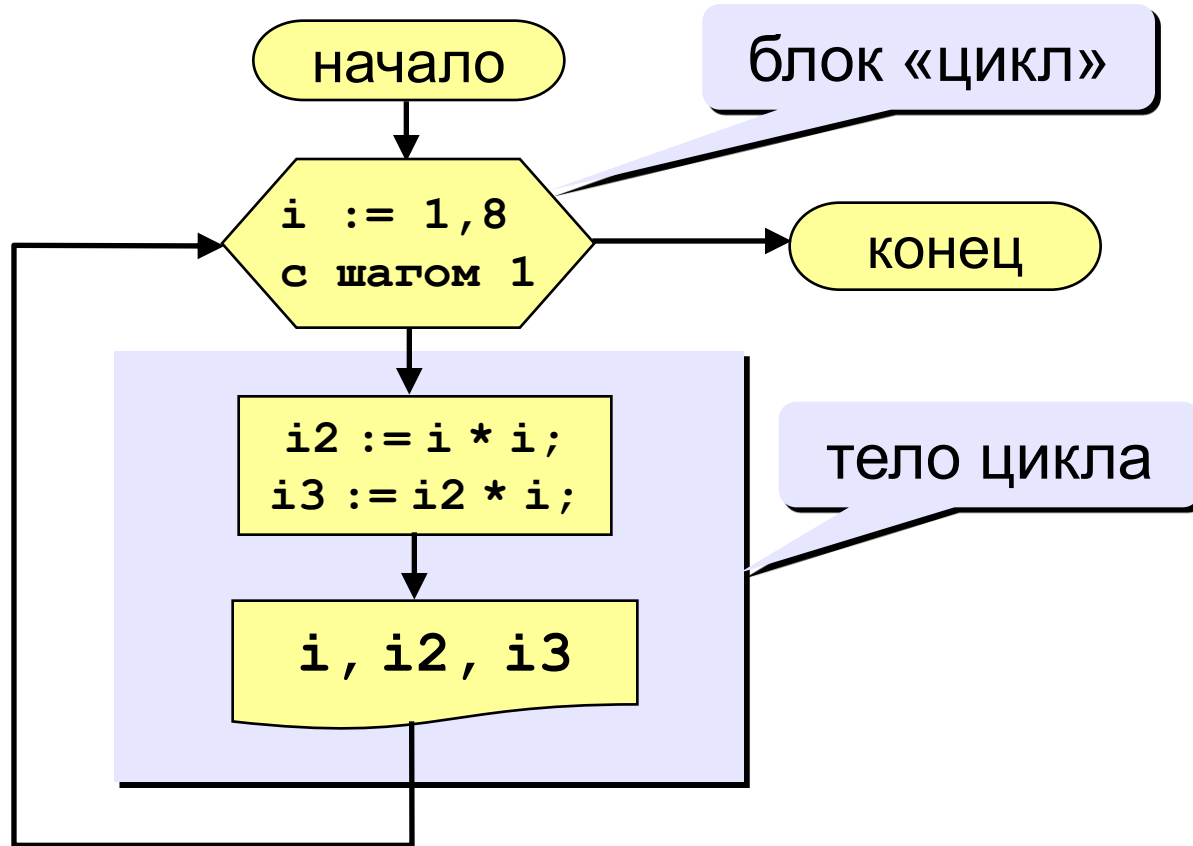
# Циклы

---

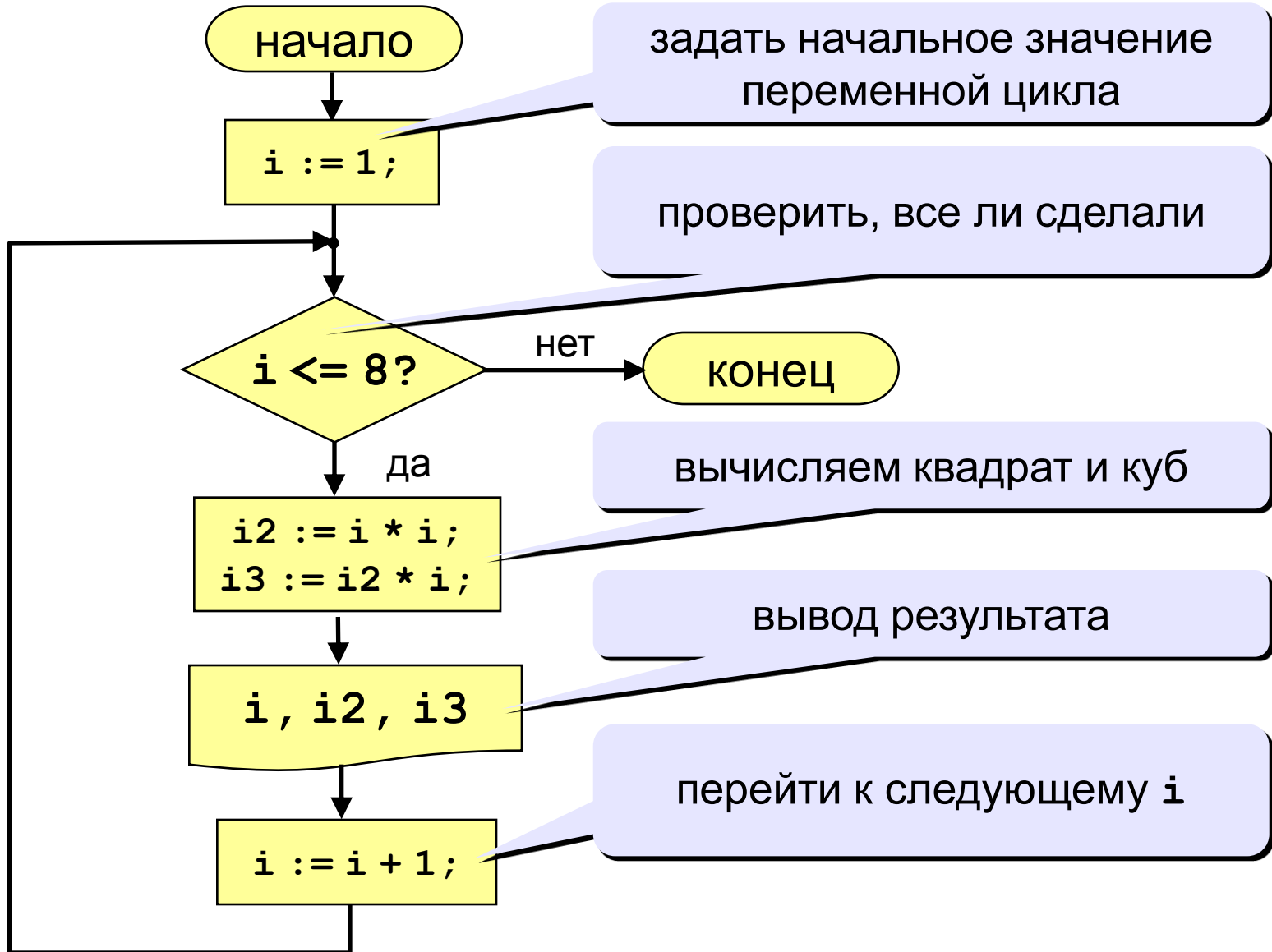
**Задача.** Вывести на экран квадраты и кубы целых чисел от 1 до 8 (от **a** до **b**).

**Особенность:** одинаковые действия выполняются 8 раз.

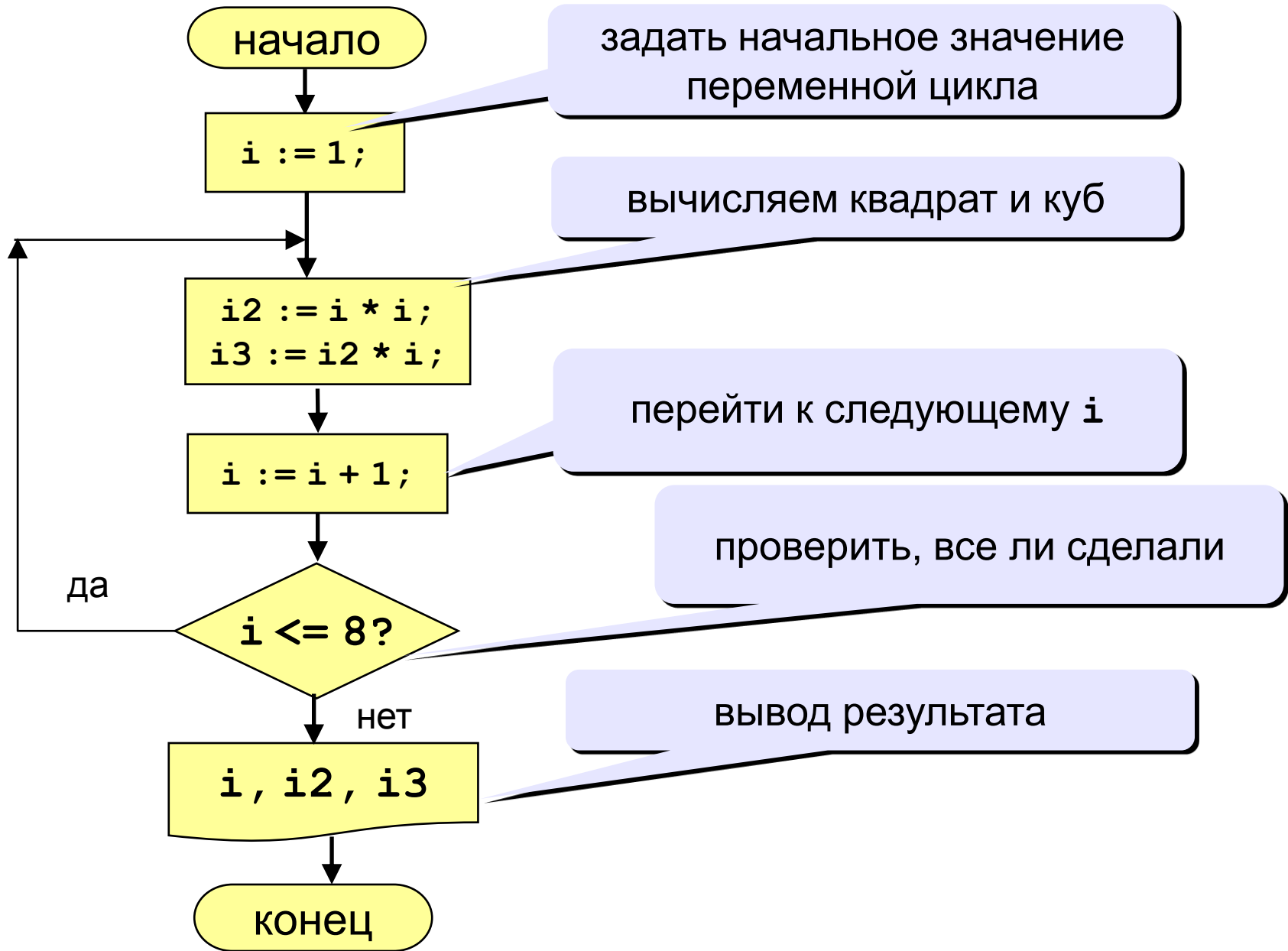
# Цикл с параметром



# Цикл с предусловием



# Цикл с постусловием



# Тренировочное задание

---

Найдите сумму и произведение трех введенных с клавиатуры чисел.

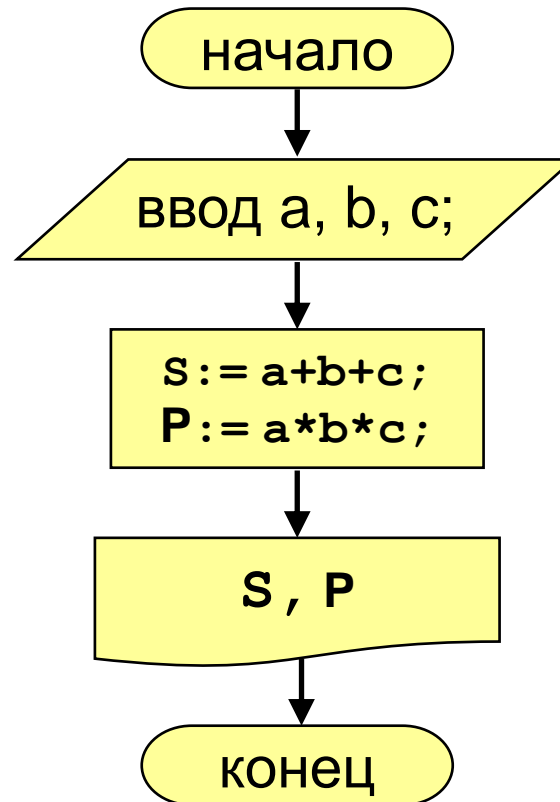
- Входные данные: 1, 2, 3.
- Выходные данные:  $S=?$ ,  $P=?$ .



# Тренировочное задание

Найдите сумму и произведение трех введенных с клавиатуры чисел.

- Входные данные: 1, 2, 3.
- Выходные данные:  $S=6$ ,  $P=6$ .



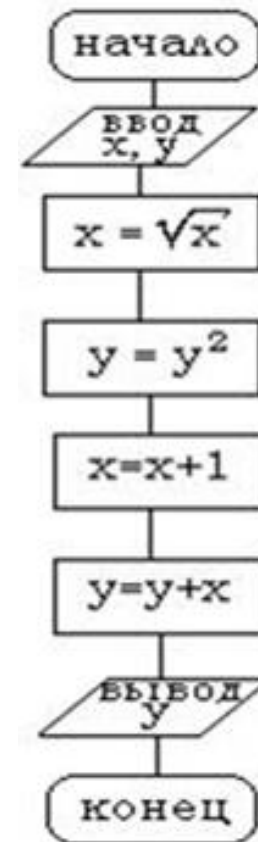
# Тренировочное задание

Определите результат выполнения алгоритма по заданной блок-схеме.

- Найдите значение  $y$  при  $x=4$ ,  $y=3$
- Найдите значение  $y$  при  $x=64$ ,  $y=10$

Создание трассировочных таблиц он-лайн

[https://www.bzfar.org/load/informatika/didakticheskij\\_material/trassirovochnaja\\_tablica/3-1-0-33](https://www.bzfar.org/load/informatika/didakticheskij_material/trassirovochnaja_tablica/3-1-0-33)

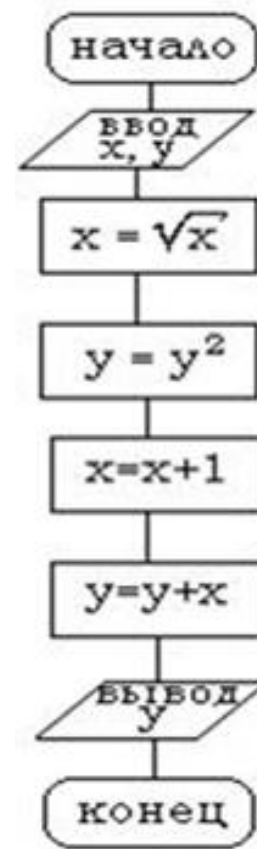


# Тренировочное задание

Определите результата выполнения алгоритма по заданной блок-схеме.

- а) Найдите значение  $y$  при  $x=4$ ,  $y=3$   
 б) Найдите значение  $y$  при  $x=64$ ,  $y=10$

	а		б	
	х	у	х	у
Ввод	4	3	64	10
$x=\sqrt{x}$	2		8	
$y=y^2$		9		100
$x=x+1$	3		9	
$y=y+x$		12		109
Вывод		12		109

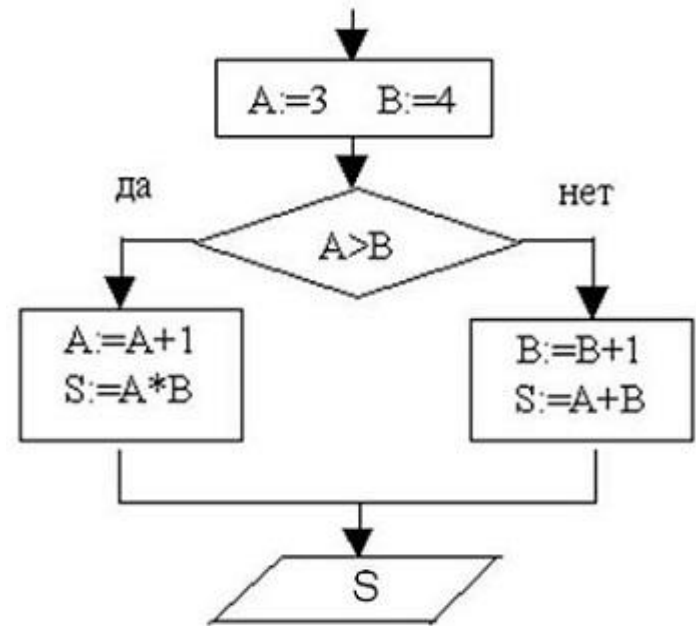


## Задание 2

# Значение переменной после выполнения алгоритма

Фрагмент алгоритма изображен в виде блок-схемы. Определите, какое значение примет переменная  $S$  после выполнения алгоритма.

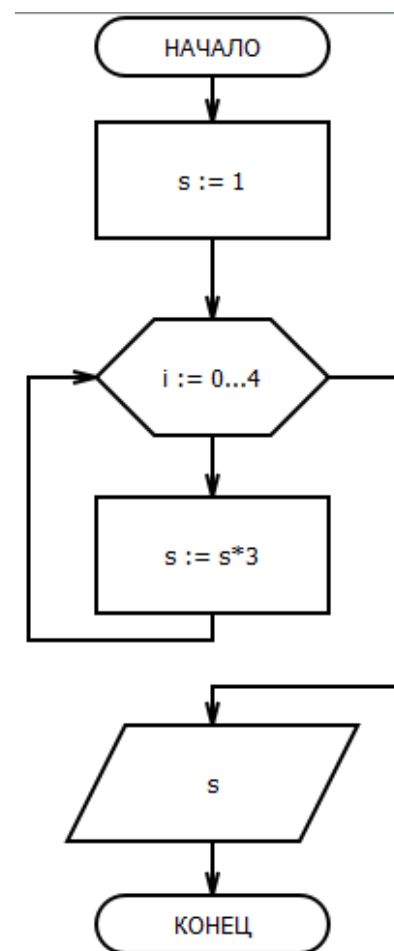
Постройте трассировочную таблицу



**Задание 3****Результат работы алгоритма**

Определите, что будет выведено в результате работы следующего алгоритма.

Постройте трассировочную таблицу



**Задание № 4**

**Составьте блок-схему и трассировочную таблицу для задачи**

**Даны длины двух катетов прямоугольного треугольника.**

**Определить периметр этого треугольника.**

- Входные данные: 3 4
- Выходные данные:  $P=12$

**Составьте блок-схему и трассировочную таблицу для задачи**

**Даны два числа. Определить наибольшее из этих чисел.**

- **Входные данные: 3, 7.**
- **Выходные данные: 7**

# Язык программирования Паскаль

**Структура программы:**

**Program** *имя программы;*

**uses** *раздел описания модулей;*

**const** *раздел описания констант;*

**var** *раздел описания переменных;*

**procedure (function)** *раздел описания подпрограмм;*

**begin**

*раздел операторов;*

**end.**

**Оператор** – команда алгоритма, записанная на ЯП

**Комментарий в программе** можно записывать в { } на любом языке.

Все, что записано в { } компилятором не воспринимается.



**Имена заголовка программы, переменных составляются по следующим правилам:**

имя должно начинаться с буквы, может содержать латинские буквы, цифры и знак \_

длина имени должна быть меньше 255 символов, но значащиеся только первые 63 символа;

прописные буквы (A...Z) тождественны строчным (a...z).

Например: book, KNIGA=kniga, nom\_1, A2

Неверные имена: 2A, Vip\*bum

**Описание переменных** происходит в разделе описания переменных:

**var** переменная: тип;

Например: **var** x: integer;

**var** a, b, c: real;

**var** a, b: integer;

После определения типов переменных компилятор для каждой переменной отводит определенную ячейку памяти, где будут храниться некоторые данные.

## Оператор присваивания

Имя переменной := выражение;

Например,  $c:=a+b$

Сначала вычисляется выражение, затем полученное выражение присваивается переменной.

Чтобы компьютер знал над какими числами производить вычисления, необходимо ввести с клавиатуры значения переменных.

## Оператор ввода

**read ( )** – оператор ввода значений переменных (читать)

**readln ( )** – оператор ввода значений переменных с переводом курсора на следующую строку (читать строку)

Например, **read (a)** или **read (a, b)**

**read (a, b)** – программа будет ждать, пока не введете значение переменных **a** и **b** через пробел

**readln (a, b)** – считывает значение **a** и **b**, затем переведет курсор на следующую строку.

## Оператор вывода

**write ( )** - оператор вывода значений переменных или фразы (писать)

**writeln ( )** – оператор вывода значений переменных или фразы с переводом курсора на следующую строку (писать в строку)

Например, **write (c)** – выведет на экран значение переменной **c** или **writeln (c)** - выведет на экран значение переменной и перейдет на следующую строку

**write (a, b, c)** – выведет значения переменных в строку без знаков препинания через позицию табуляции

**write ('мама')** – выведет слово, заключенное в апострофах

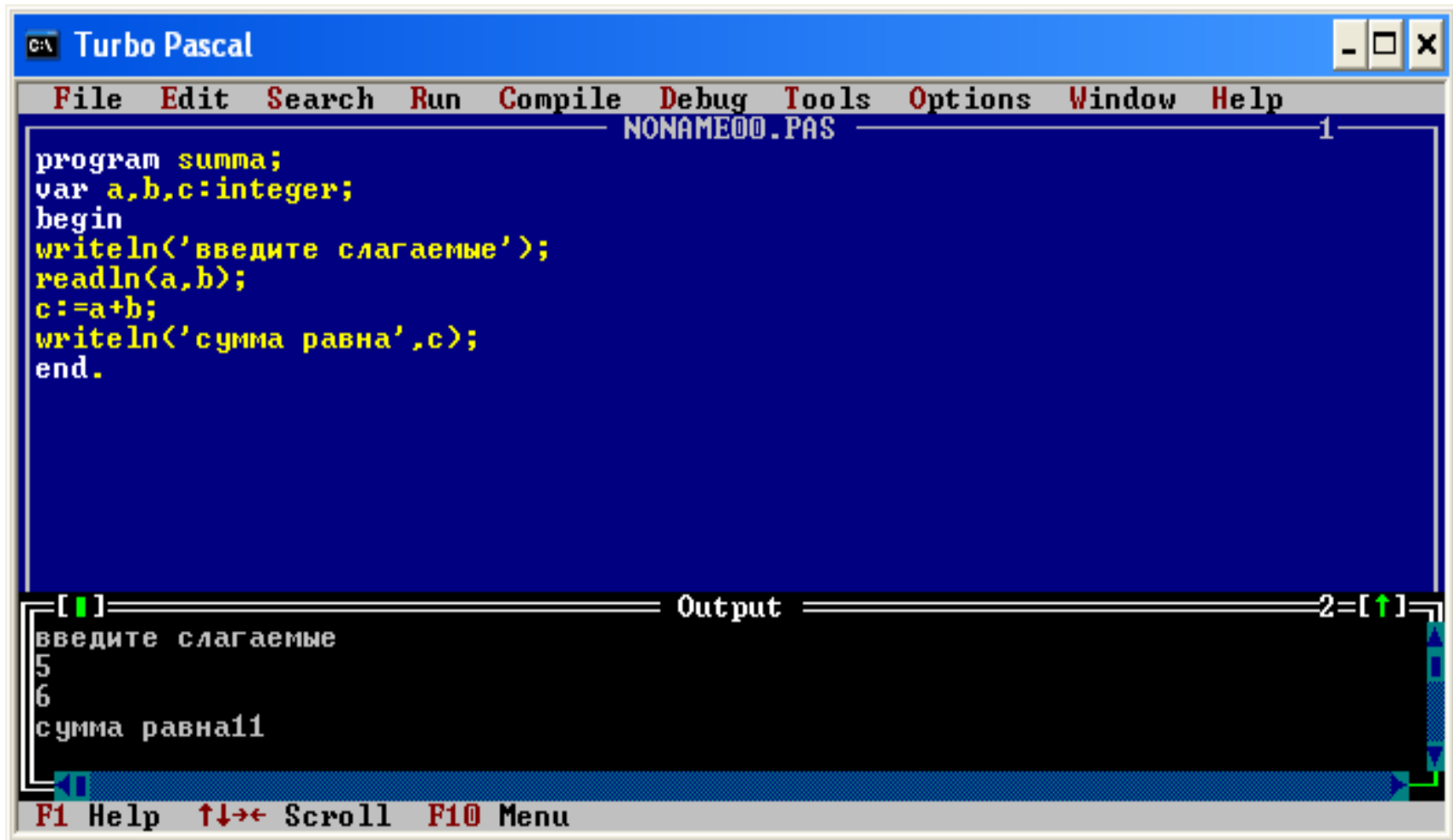
**write ('сумма равна ', c)** – выведет, например, **сумма равна 13**

**writeln (a, b, c)** – выведет значения переменных в строку без знаков препинания через позицию табуляции и переведет курсор на следующую строку.

**Запишем теперь полностью программу  
на Pascal.**

```
Program summa;  
var a, b, c: integer;  
begin  
writeln ('введите слагаемые');  
readln (a, b);  
c:=a+b;  
writeln ('сумма равна', c);  
end.
```

# Запустите программу



The image shows a screenshot of the Turbo Pascal IDE. The main window displays a Pascal program named 'NONAME00.PAS' with the following code:

```
program summa;  
var a,b,c:integer;  
begin  
writeln('введите слагаемые');  
readln(a,b);  
c:=a+b;  
writeln('сумма равна',c);  
end.
```

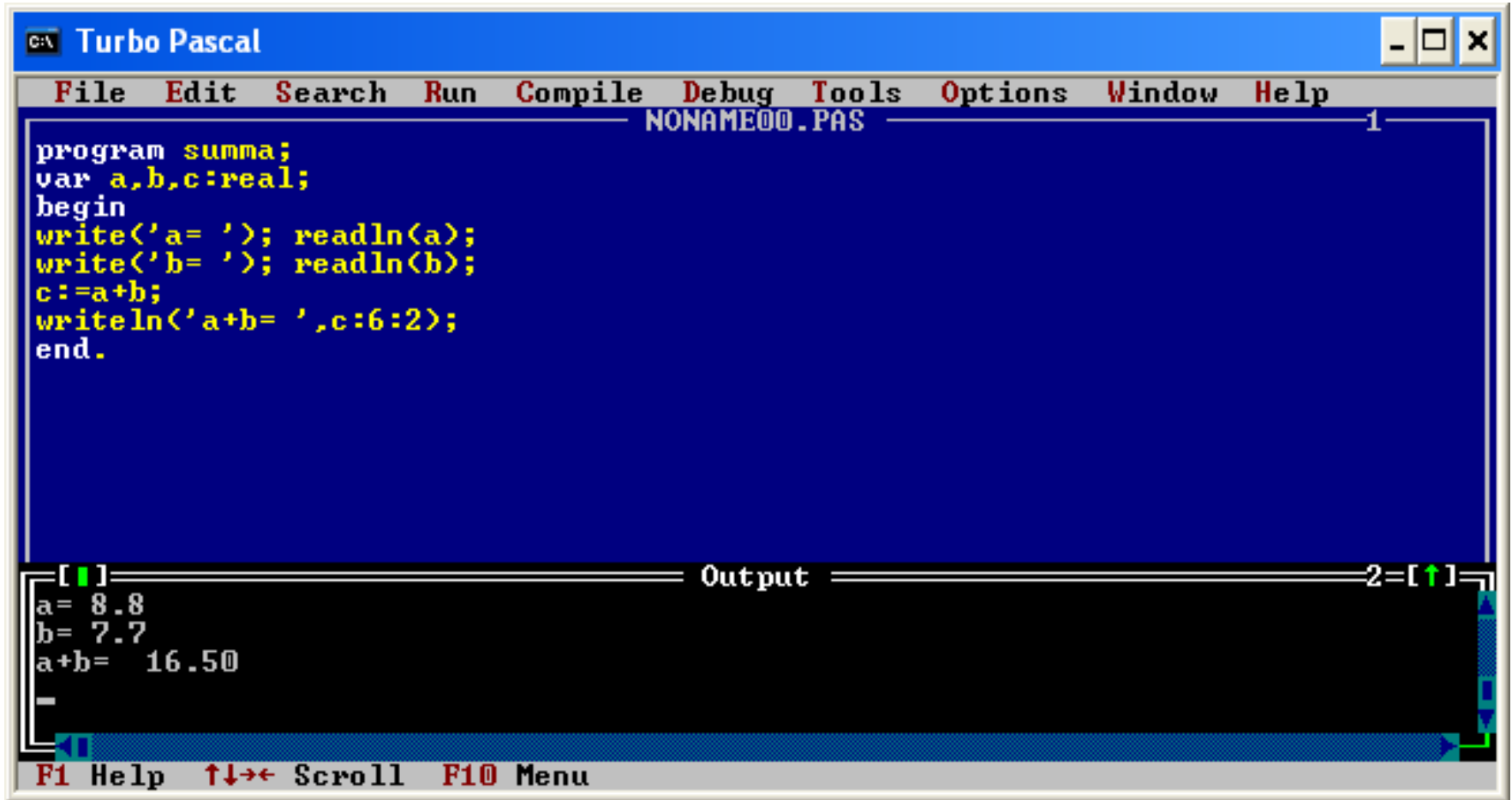
Below the code editor is an 'Output' window showing the program's execution:

```
введите слагаемые  
5  
6  
сумма равна11
```

The IDE interface includes a menu bar with 'File', 'Edit', 'Search', 'Run', 'Compile', 'Debug', 'Tools', 'Options', 'Window', and 'Help'. The status bar at the bottom shows 'F1 Help', '↑↓↔ Scroll', and 'F10 Menu'.

**Исправьте программу следующим образом,  
запустите и найдите отличия:**

```
Program summa;  
var a, b, c: integer;  
begin  
write ('a= '); readln (a);  
write ('b= '); readln (b);  
c:=a+b;  
writeln ('a+b= ',c);  
end.
```



The image shows a screenshot of the Turbo Pascal IDE. The window title is "Turbo Pascal". The menu bar includes "File", "Edit", "Search", "Run", "Compile", "Debug", "Tools", "Options", "Window", and "Help". The main editor area displays a Pascal program named "NONAME00.PAS" with the following code:

```
program summa;  
var a,b,c:real;  
begin  
write('a= '); readln(a);  
write('b= '); readln(b);  
c:=a+b;  
writeln('a+b= ',c:6:2);  
end.
```

Below the editor is an "Output" window showing the execution results:

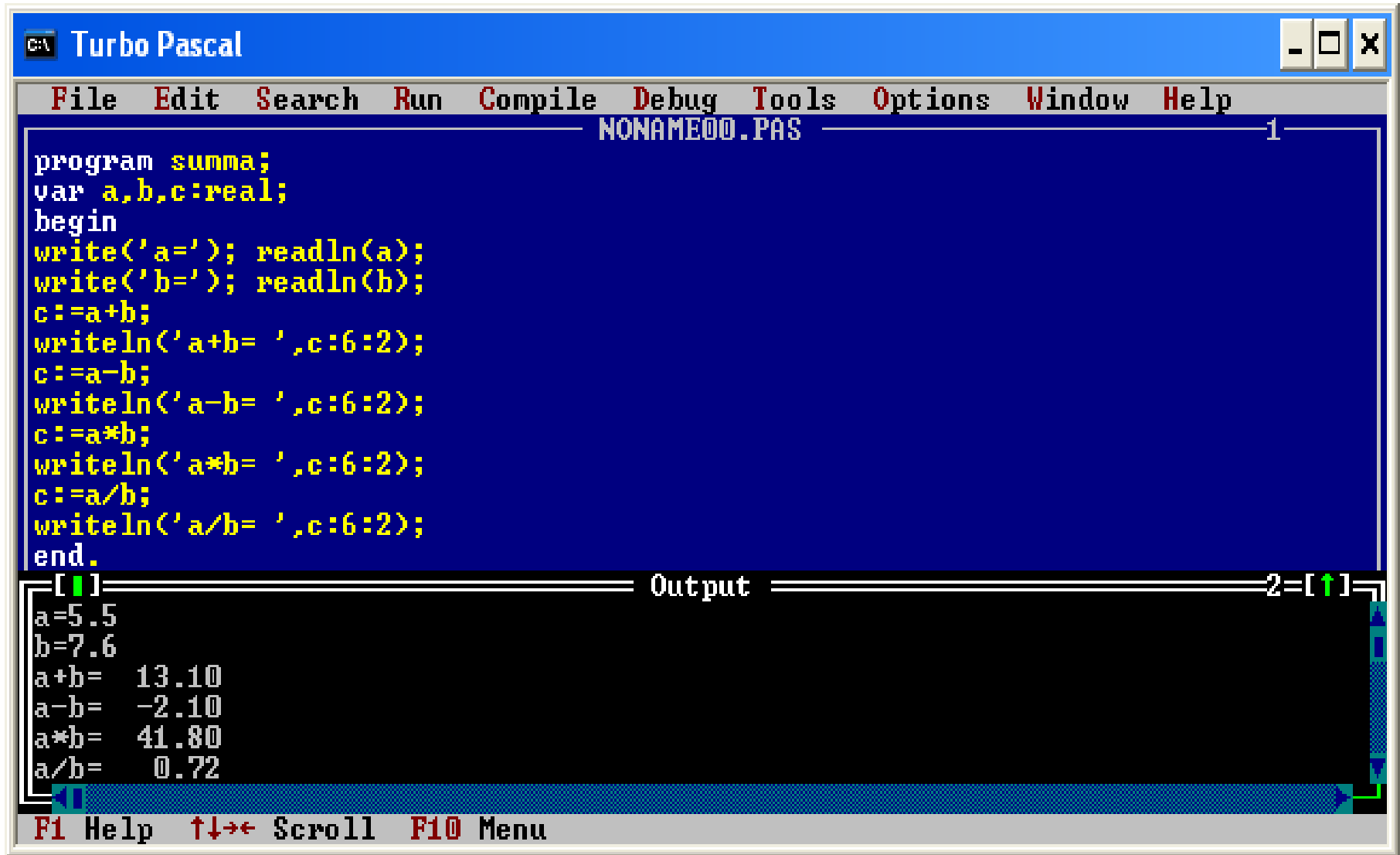
```
a= 8.8  
b= 7.7  
a+b= 16.50  
-
```

The status bar at the bottom of the IDE shows "F1 Help", "↑↓→← Scroll", and "F10 Menu".

Запись **c:6:2** означает, что значение переменной **c** будет выведено в шесть позиций, из них в две после запятой.



Добавьте к программе операторы, позволяющие производить основные математические операции (вычитание, умножение, деление).



The image shows a screenshot of the Turbo Pascal IDE. The main window displays a Pascal program named 'NONAME00.PAS'. The program defines a variable 'c' and performs arithmetic operations on 'a' and 'b'. The output window shows the results of these operations for 'a=5.5' and 'b=7.6'.

```
program summa;  
var a,b,c:real;  
begin  
write('a='); readln(a);  
write('b='); readln(b);  
c:=a+b;  
writeln('a+b= ',c:6:2);  
c:=a-b;  
writeln('a-b= ',c:6:2);  
c:=a*b;  
writeln('a*b= ',c:6:2);  
c:=a/b;  
writeln('a/b= ',c:6:2);  
end.
```

Output

```
a=5.5  
b=7.6  
a+b= 13.10  
a-b= -2.10  
a*b= 41.80  
a/b= 0.72
```

F1 Help ↑↓← Scroll F10 Menu

# Спасибо за внимание

---

**Ваше настроение- ваш статус.**