



# АЛГОРИТМЫ И ИСПОЛНИТЕЛИ

## ОСНОВЫ АЛГОРИТМИЗАЦИИ

8 класс



ИЗДАТЕЛЬСТВО

**БИНОМ**

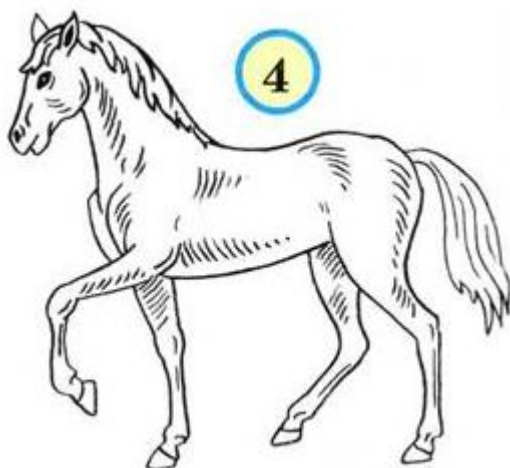
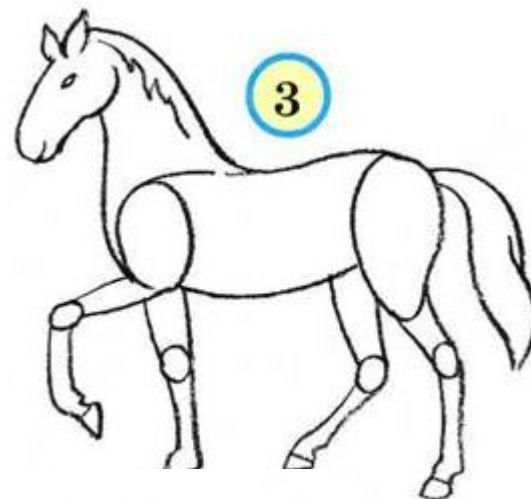
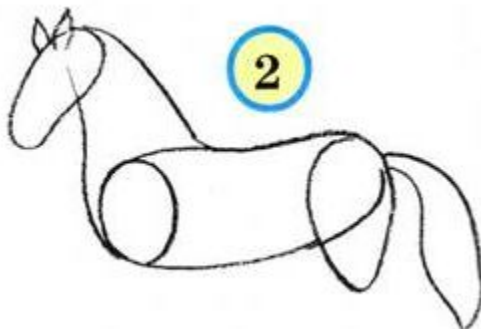
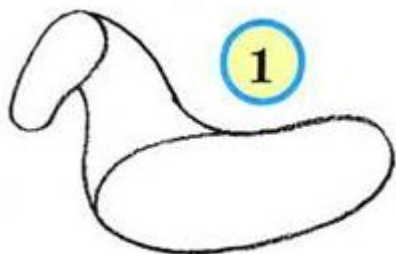
# Ключевые слова

- алгоритм
- свойства алгоритма
  - дискретность
  - понятность
  - определённость
  - результативность
  - массовость
- исполнитель
- характеристики исполнителя
  - круг решаемых задач
  - среда
  - режим работы
  - система команд
- формальное исполнение алгоритма



# Примеры алгоритмов

## Рисование лошади



# Примеры алгоритмов

## Вычислительный алгоритм

### Среднее арифметическое двух чисел

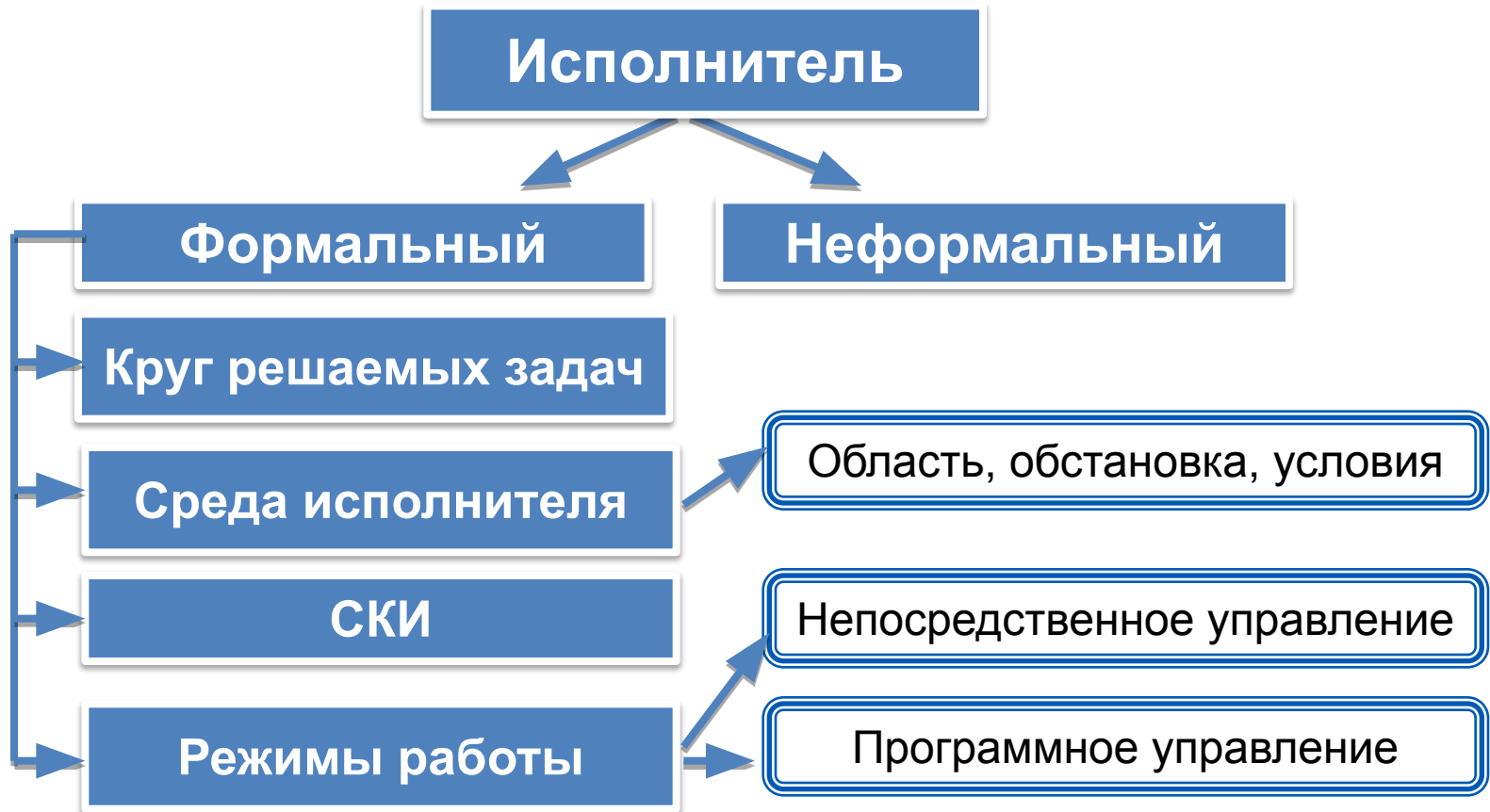
1. Задать два числа
2. Сложить заданные числа
3. Разделить сумму на 2

### Общая схема работы алгоритма



# Исполнитель алгоритма

Исполнитель - это некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.



# Исполнитель Робот

The image shows a screenshot of a programming environment. The main window is titled "Новая программа - Кумир" (New program - KUMIR). The menu bar includes "Программа", "Редактирование", "Вставка", "Выполнение", "Инструменты", "Робот", and "Чертежник". The toolbar contains various icons for editing and execution. The code editor displays the following code:

```
1  использовать Робот
2
3  алг
4  нач
5  ▪  вправо
6  ▪  закрасить; вниз
7  ▪  закрасить; влево
8  ▪  закрасить
9  кон
10
```

Below the code editor, there is a console window showing the following output:

```
>> 17:50:24 - Новая программа* - В
>> 17:50:24 - Новая программа* - В
```

Overlaid on the right side of the code editor is a window titled "Робот - 10x16.fil". This window displays a 10x16 grid with a green background. A yellow diamond-shaped robot is positioned at the top-left corner of the grid. The robot's current path is highlighted in grey, showing a 2x2 square area starting from its position.

# Исполнитель Кузнечик

The image shows a screenshot of the 'Кузнечик' (The Grasshopper) programming environment. The main window is titled '1.kum - Кумир' and contains a menu bar with 'Программа', 'Редактирование', 'Вставка', 'Выполнение', 'Инструменты', 'Робот', and 'Чертежник'. Below the menu is a toolbar with various icons for file operations, editing, and execution. The main area displays a script with the following commands:

```
1  использовать Кузнечик
2  вперед 3
3  назад 2
4  назад 2
5  назад 2
6
7
8
9
10
11
12
13
```

An inset window titled 'Кузнечик - нет файла' shows a number line from -7 to 7. The number 0 is highlighted in green. A blue arrow points to the number -3, and three curved arrows above the line indicate jumps of 3 units forward from -3 to 0, 1, and 2.

# Свойства алгоритма

## Свойства алгоритма

**Дискретность**

Путь решения задачи разделён на отдельные шаги

**Понятность**

Алгоритм состоит из команд, входящих в СИ

**Определённость  
(детерминированность)**

Команды понимаются однозначно

**Результативность**

Обеспечивается получение ожидаемого результата

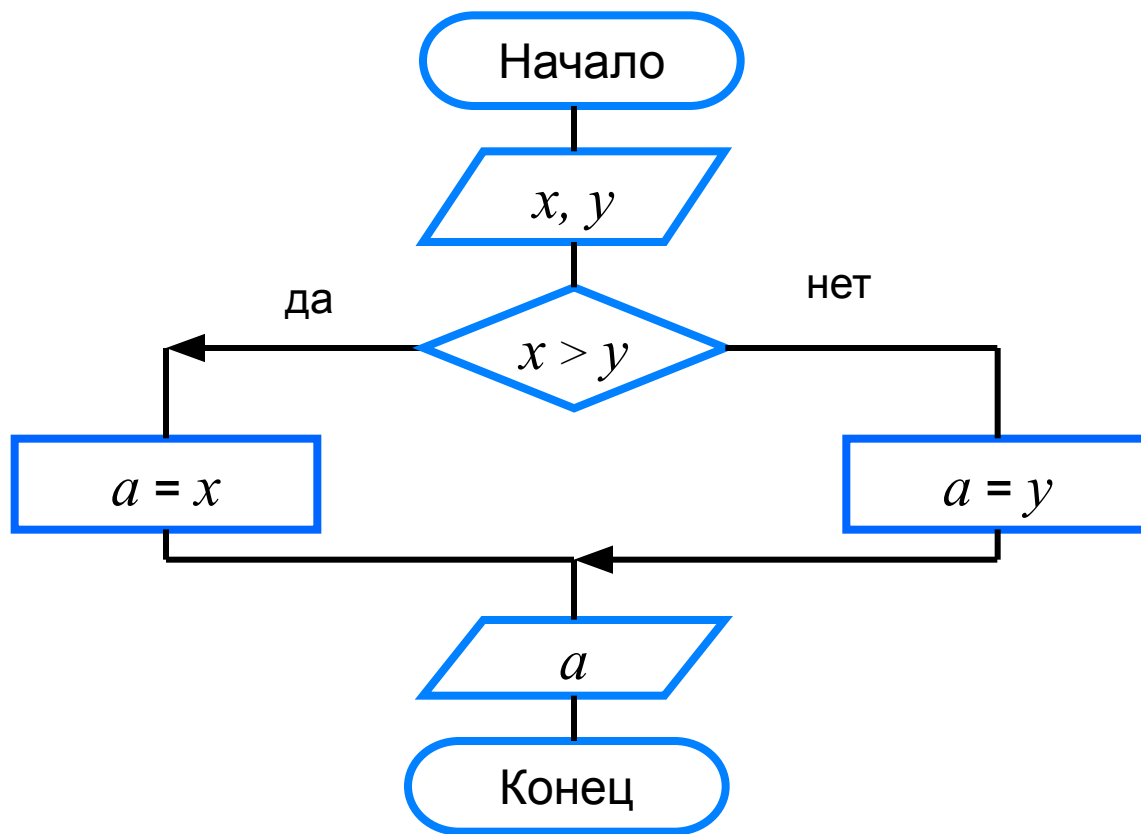
**Массовость**

Обеспечивается решение задач с различными исходными данными





**Дискретность** (от лат. *discretus* – разделенный, прерывистый) означает, что путь решения задачи разделён на отдельные шаги (действия). Каждому действию соответствует предписание (команда). Только выполнив одну команду, исполнитель сможет приступить к выполнению следующей.



**Понятность** означает, что алгоритм состоит только из команд, входящих в систему команд исполнителя, т. е. из таких команд, которые исполнитель может воспринять и по которым может выполнить требуемые действия.



### Окрошка «Мясная»

1.5 л кваса	300 г огурцов
500 г картофеля	зелень по вкусу
300 г колбасы	сметана
3 яйца	соль
200 г редиса	перец

### Рецепт приготовления

Картофель отварить до готовности.  
Остудить, почистить.  
Нарезать кубиками.  
Колбасу нарезать кубиками.  
Яйца нарезать кубиками.  
Редис тонко нарезать.  
Огурцы нарезать кубиками.

Смешать картофель, колбасу, яйца, редис, огурцы.  
Посолить, поперчить.  
Выложить в тарелки.  
Залить квасом, посыпать зеленью.  
Подавать со сметаной.



**Определённость** означает, что в алгоритме нет команд, смысл которых может быть истолкован исполнителем неоднозначно; недопустимы ситуации, когда после выполнения очередной команды исполнителю неясно, какую команду выполнять на следующем шаге.



## Доехать до стадиона

1. Идти прямо
2. Повернуть
3. Идти прямо
4. Сесть в автобус
5. Доехать до остановки «Стадион»



Данная последовательность действий не обладает свойством определённости!

Какое расстояние нужно пройти прямо?

В какую сторону повернуть?

В какой автобус сесть?



**Результативность** означает, что алгоритм должен обеспечивать возможность получения результата после конечного, возможно, очень большого, числа шагов. При этом результатом считается не только обусловленный постановкой задачи ответ, но и вывод о невозможности продолжения по какой-либо причине решения данной задачи.

### Чтение книги



1. Взять книгу
2. Открыть первую страницу
3. Пока не конец книги выполнять следующие действия:
  - 3.1 Прочитать текст
  - 3.2 Перелистнуть страницу
  - 3.3 Прочитать текст
  - 3.4 Открыть первую страницу

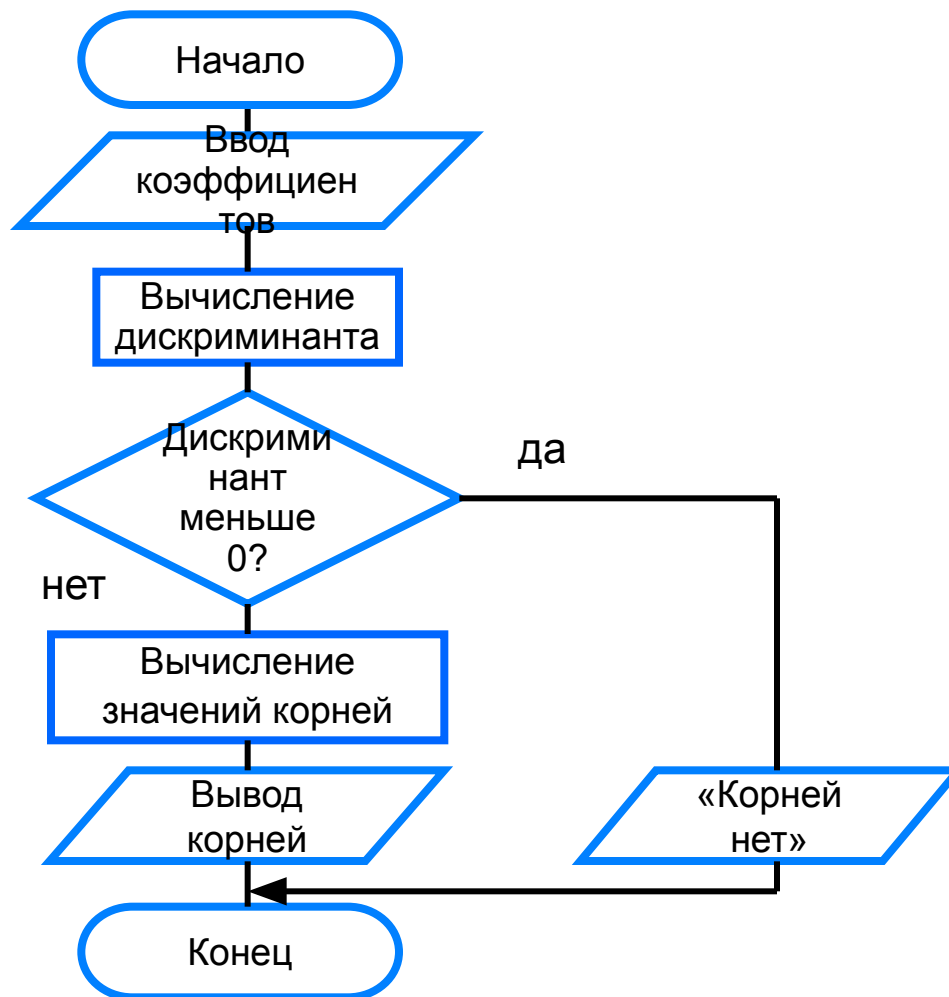
Данная последовательность команд не обладает свойством результативности. Что нужно изменить?



**Массовость** означает, что алгоритм должен обеспечивать возможность его применения для решения любой задачи из некоторого класса задач с различными исходными данными.

### Алгоритм вычисления корней квадратного уравнения.

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



**Алгоритм** - это предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами:

- ***дискретности***
- ***понятности***
- ***определённости***
- ***результативности***
- ***массовости***

# Основные способы записи алгоритма

## Словесные

Словесное описание

Построчная запись

## Графические

Последовательность рисунков

Структурограмма

Блок-схема

На алгоритмических языках

Школьный алгоритмический язык

Язык программирования



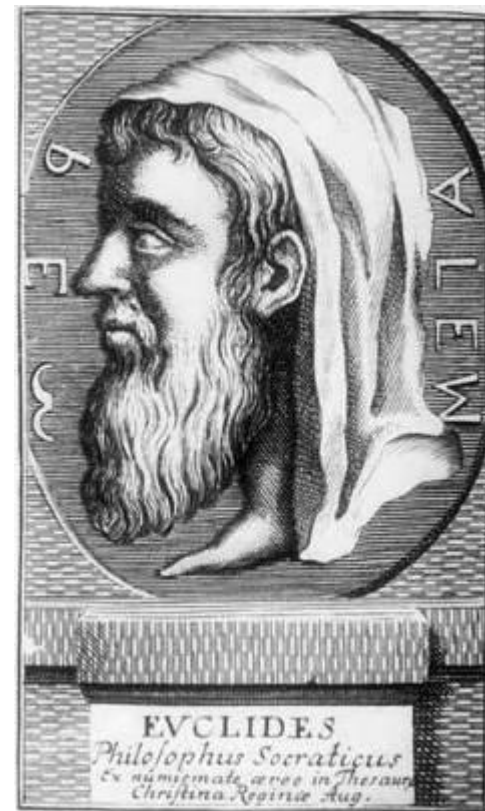
# Словесное описание

**Словесное описание** - самая простая запись алгоритма в виде набора высказываний на обычном разговорном языке.

**Пример.** Словесное описание алгоритма нахождения наибольшего общего делителя (НОД) пары целых чисел (алгоритм Евклида).

Чтобы найти НОД двух чисел, составьте таблицу из двух столбцов и назовите столбцы  $X$  и  $Y$ . Запишите первое из заданных чисел в столбец  $X$ , а второе - в столбец  $Y$ . Если данные числа не равны, замените большее из них на результат вычитания из большего числа меньшего.

Повторяйте такие замены до тех пор, пока числа не окажутся равными, после чего число из столбца  $X$  считайте искомым результатом.





# Построчная запись

## Правила построчной записи алгоритма

Каждое предписание записывается с новой строки

Предписание (шаги) алгоритма нумеруются

Исполнение алгоритма происходит в порядке возрастания номеров шагов, начиная с первого, если нет особых указаний

Кроме слов естественного языка предписания могут содержать математические выражения и формулы.

# Построчная запись алгоритма Евклида

## Построчная запись алгоритма Евклида

1. Начало.
2. Обозначить первое из заданных чисел  $X$ , второе -  $Y$ .
3. Если  $X = Y$  то перейти к п. 9.
4. Если  $X > Y$ , то перейти к п. 5, иначе перейти к п. 7.
5. Заменить  $X$  на  $X - Y$ .
6. Перейти к п. 3.
7. Заменить  $Y$  на  $Y - X$
8. Перейти к п. 3.
9. Считать  $X$  искомым результатом.
10. Конец.

# Блок-схемы

В блок-схеме предписания изображаются с помощью различных геометрических фигур, а последовательность выполнения шагов указывается с помощью линий.



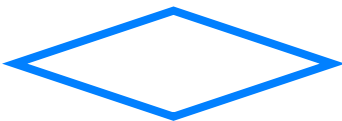
Блок начала или конца алгоритма



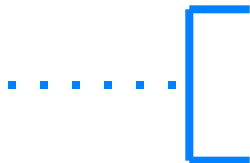
Блок ввода или вывода данных



Блок обработки данных

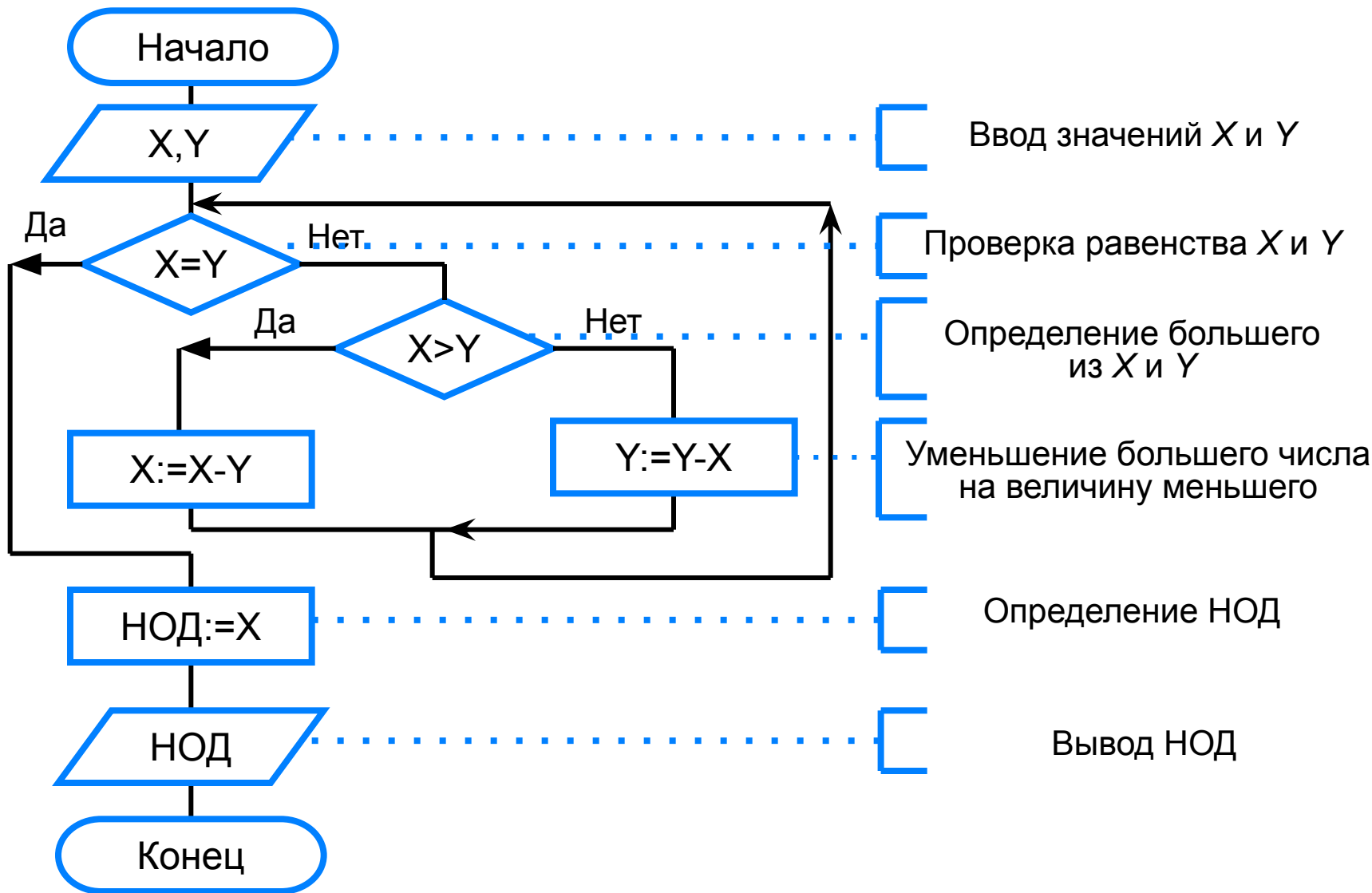


Блок проверки условия



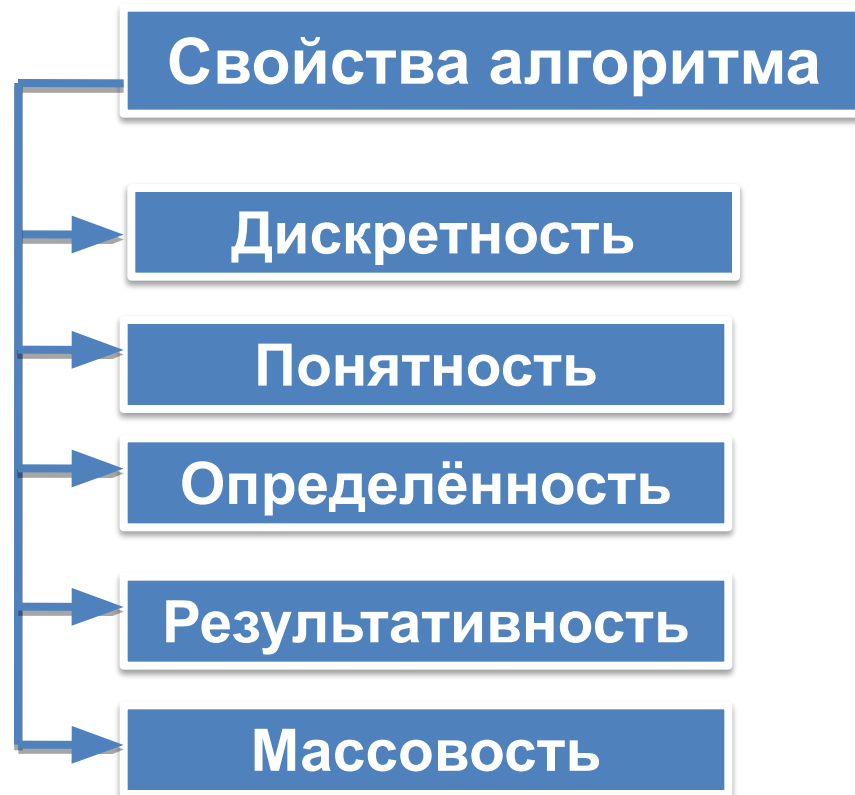
Блок пояснительных записей

# Запись алгоритма Евклида с помощью блок-схемы



# Опорный конспект

**Алгоритм** - это предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами **дискретности, понятности, определённости, результативности и массовости.**



# Способы записи алгоритмов

```
graph TD; A[Способы записи алгоритмов] --> B[Словесный]; A --> C[Блок-схема]; A --> D[Алгоритмический язык]; B --> E[Обычный разговорный язык]; C --> F[Геометрические фигуры]; D --> G[Слова имеют заданный смысл и способ записи];
```

**Словесный**

Обычный  
разговорный  
язык

**Блок-схема**

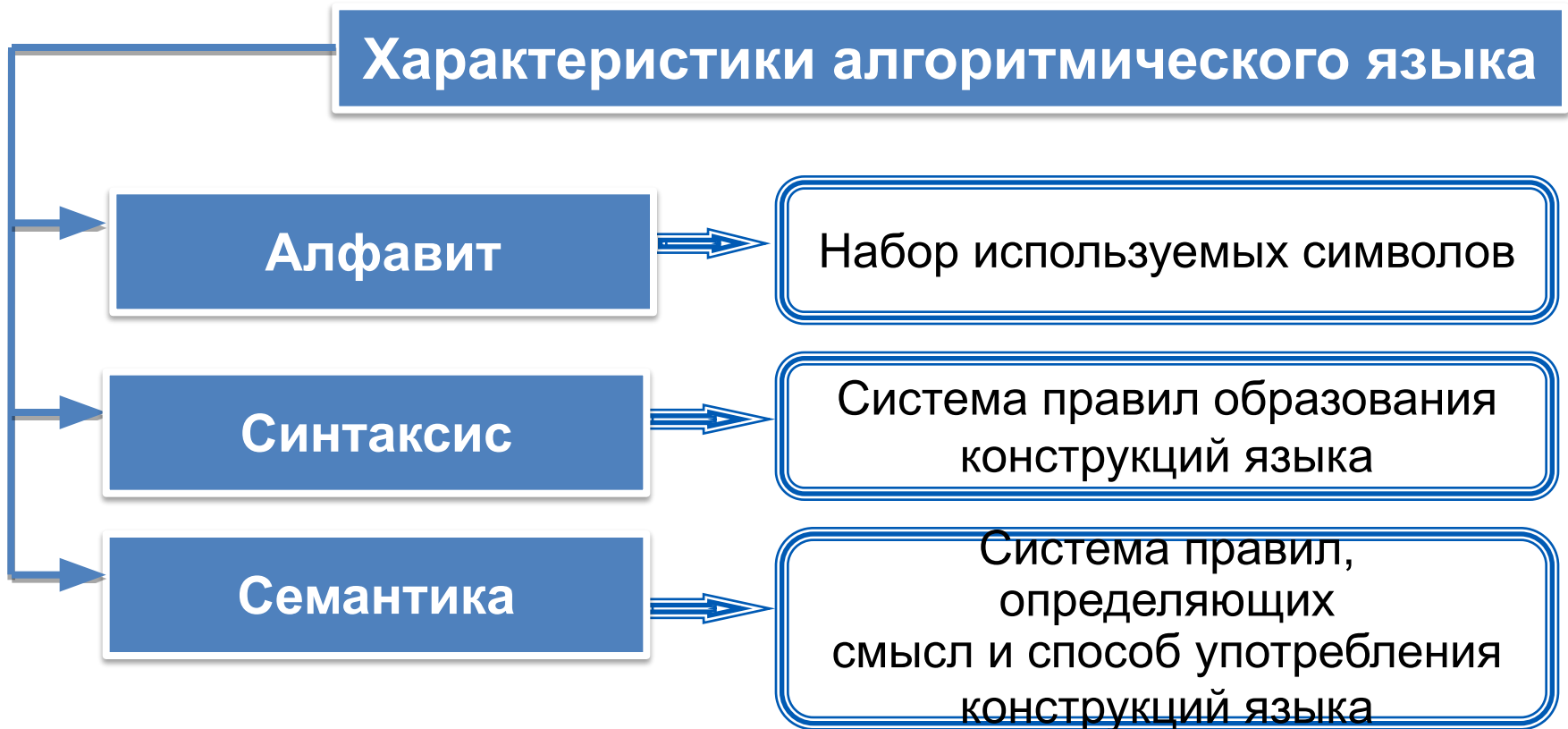
Геометрические  
фигуры

**Алгоритмический  
язык**

Слова имеют  
заданный смысл  
и  
способ записи

# Алгоритмические языки

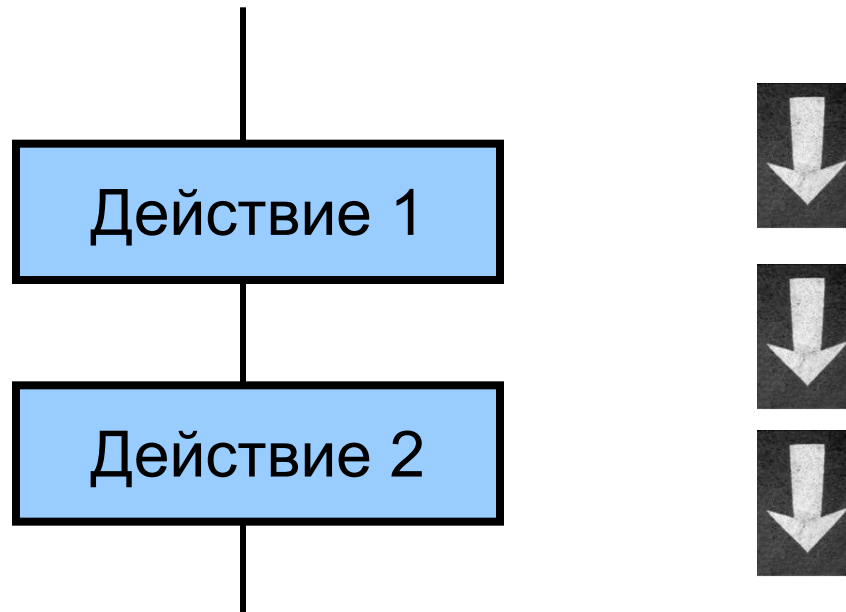
Алгоритмические языки - формальные языки, предназначенные для записи алгоритмов.



# Следование

**Следование** - алгоритмическая конструкция, отображающая естественный, последовательный порядок действий.

Алгоритмы, в которых используется только структура «следование», называются **линейными алгоритмами**.



*Алгоритмическая структура «следование»*



# Ветвление

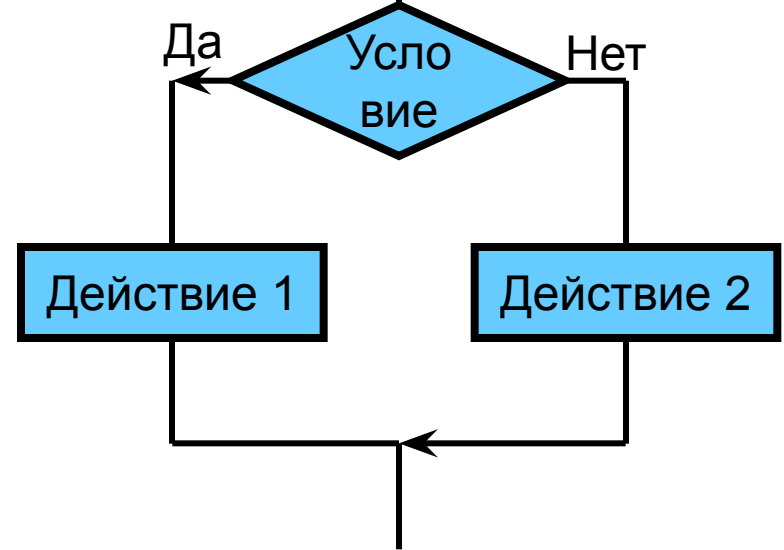
**Ветвление** - алгоритмическая конструкция, в которой в зависимости от результата проверки условия («да» или «нет») предусмотрен выбор одной из двух последовательностей действий (ветвей).

Алгоритмы, в основе которых лежит структура «ветвление», называются **разветвляющимися**.



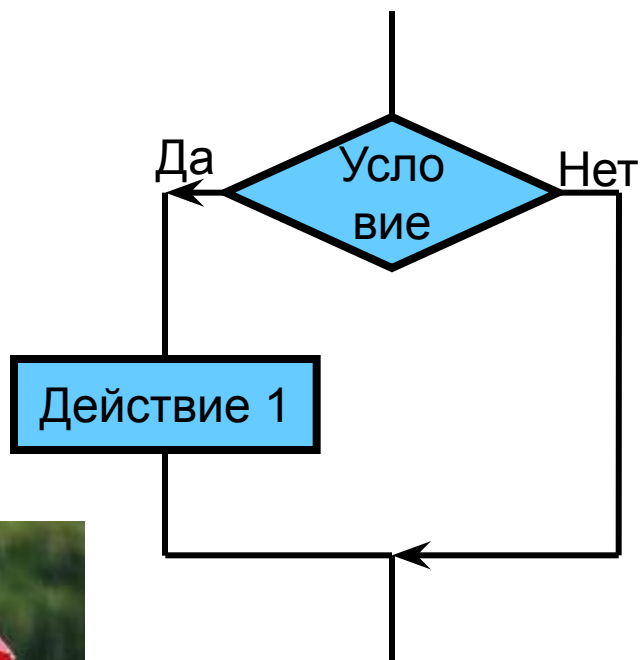
# Полная форма ветвления

**если** <условие>  
**то** <действие 1>  
**иначе** <действие 2>  
**все**



# Неполная форма ветвления

**если** <условие>  
**то** <действие 1>  
**все**



# Операции сравнения

$A < B$       A меньше B

$A \leq B$       A меньше или равно B

$A = B$       A равно B

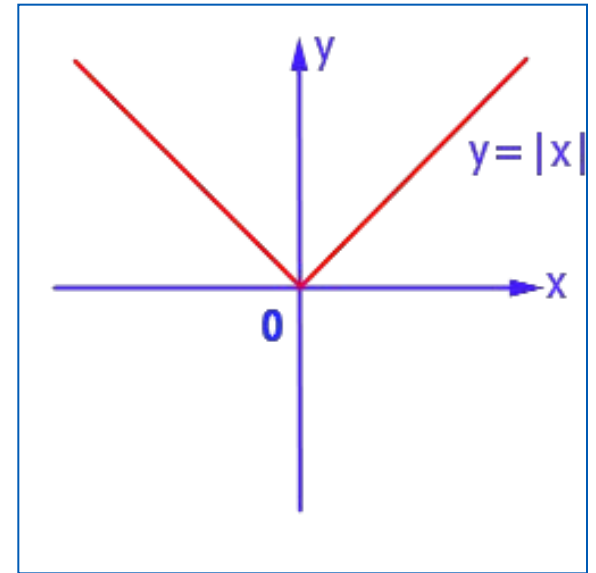
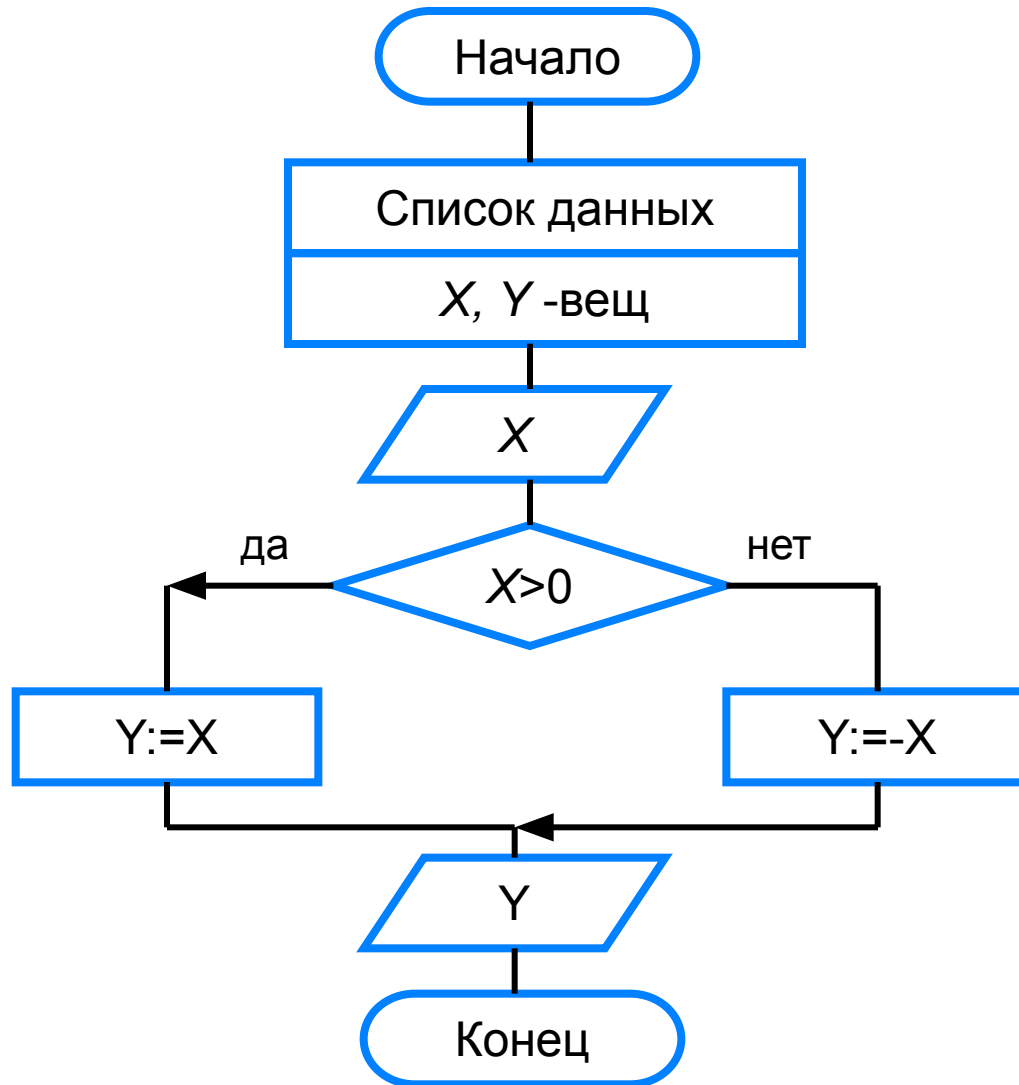
$A > B$       A больше B

$A \geq B$       A больше или равно B

$A \neq B$       A не равно B



# Вычисление функции $f(x)=|x|$

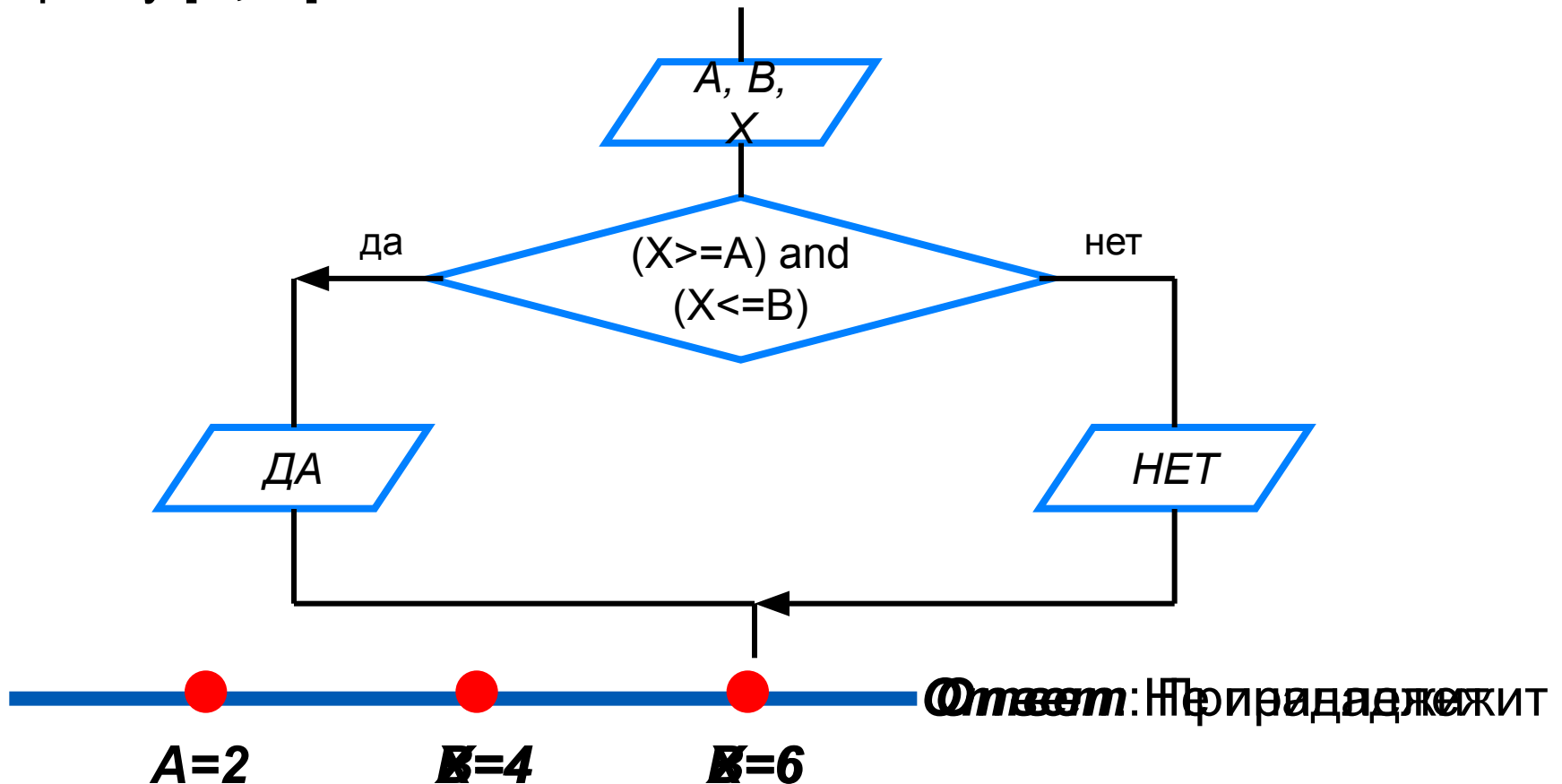


# Простые и составные условия

**Простые** условия состоят из одной операции сравнения.

**Составные** условия получаются из простых с помощью логических связок *and* (**и**), *or* (**или**), *not* (**не**).

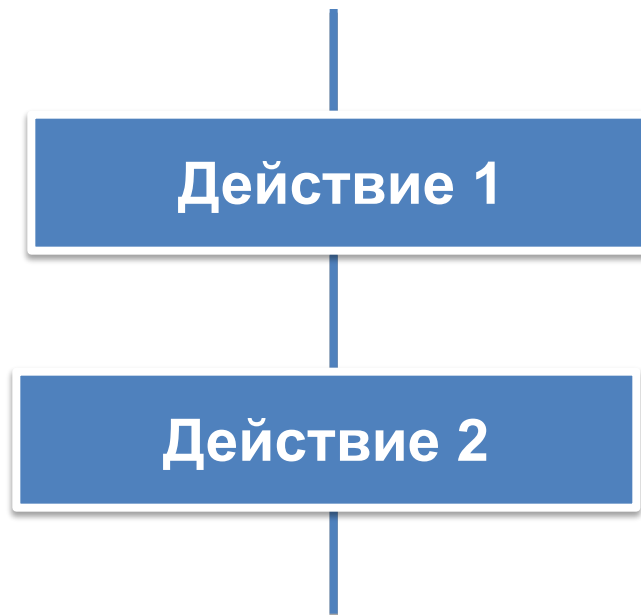
**Пример.** Алгоритм определения принадлежности точки  $X$  отрезку  $[A; B]$ .



# Опорный концепт

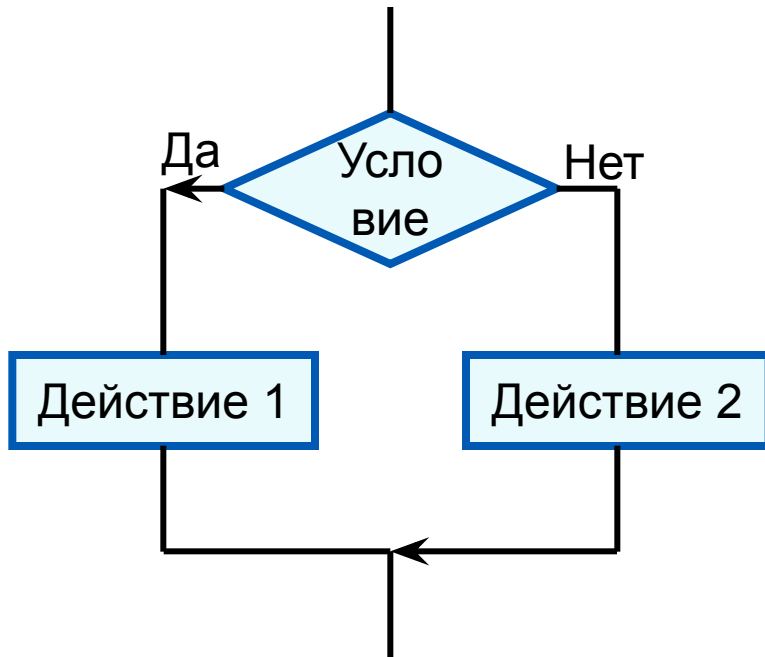
**Следование** - алгоритмическая конструкция, отображающая естественный, последовательный порядок действий.

Алгоритмы, в которых используется только структура «следование», называются **линейными**.

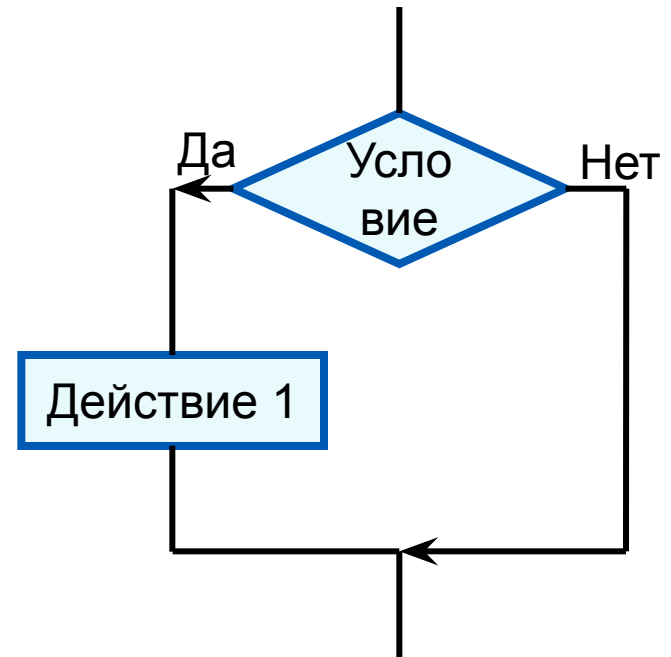


**Ветвление** - алгоритмическая конструкция, в которой в зависимости от результата проверки условия (да или нет) предусмотрен выбор одной из двух последовательностей действий (ветвей).

**Алгоритмы**, в основе которых лежит структура «ветвление», называют **разветвляющимися**.



*Полная форма ветвления*



*Неполная форма ветвления*



# Повторение

**Повторение** - последовательность действий, выполняемых многократно.

**Алгоритмы**, содержащие конструкцию повторения, называют **циклическими** или **циклами**.

Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется **телом цикла**.



# Типы циклов



Могут быть

Заданы условия  
продолжения  
работы

*Пока есть кирпич*

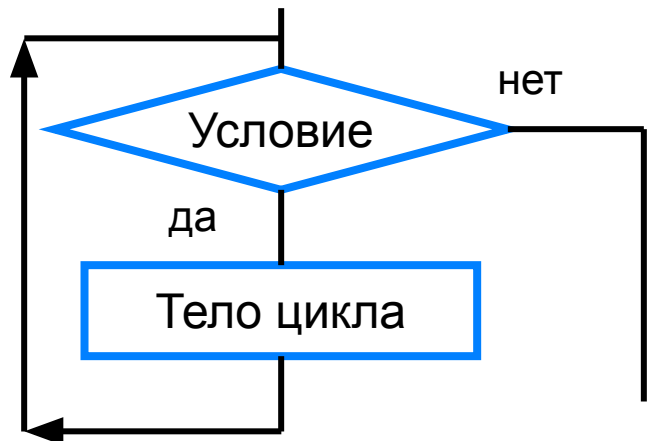
Заданы условия  
окончания работы

*Пока не наступит  
ночь*

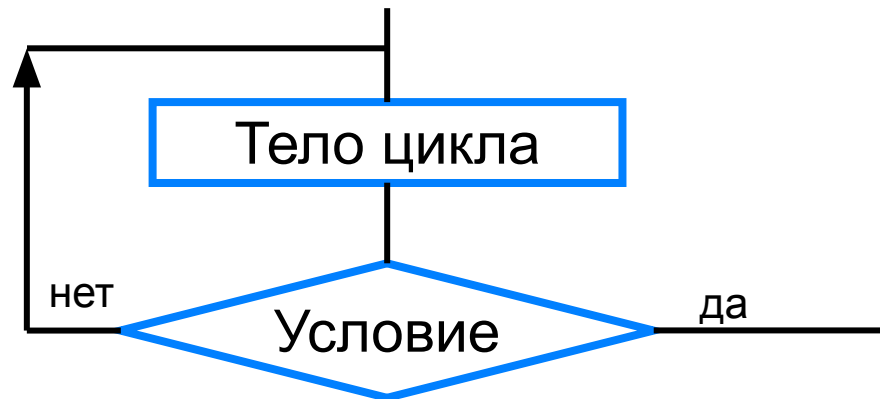
Задано число  
повторений

*Ровно 100 кирпичей*

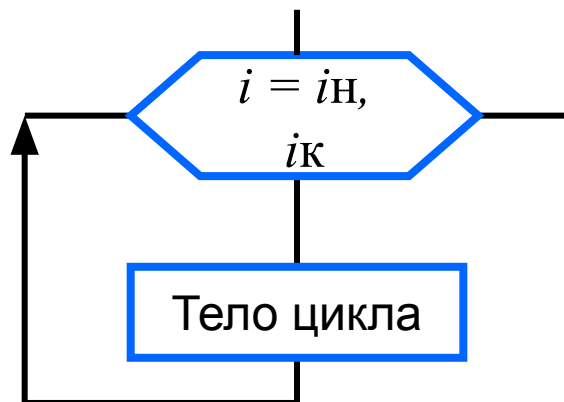
## цикл-ПОКА, цикл с предусловием



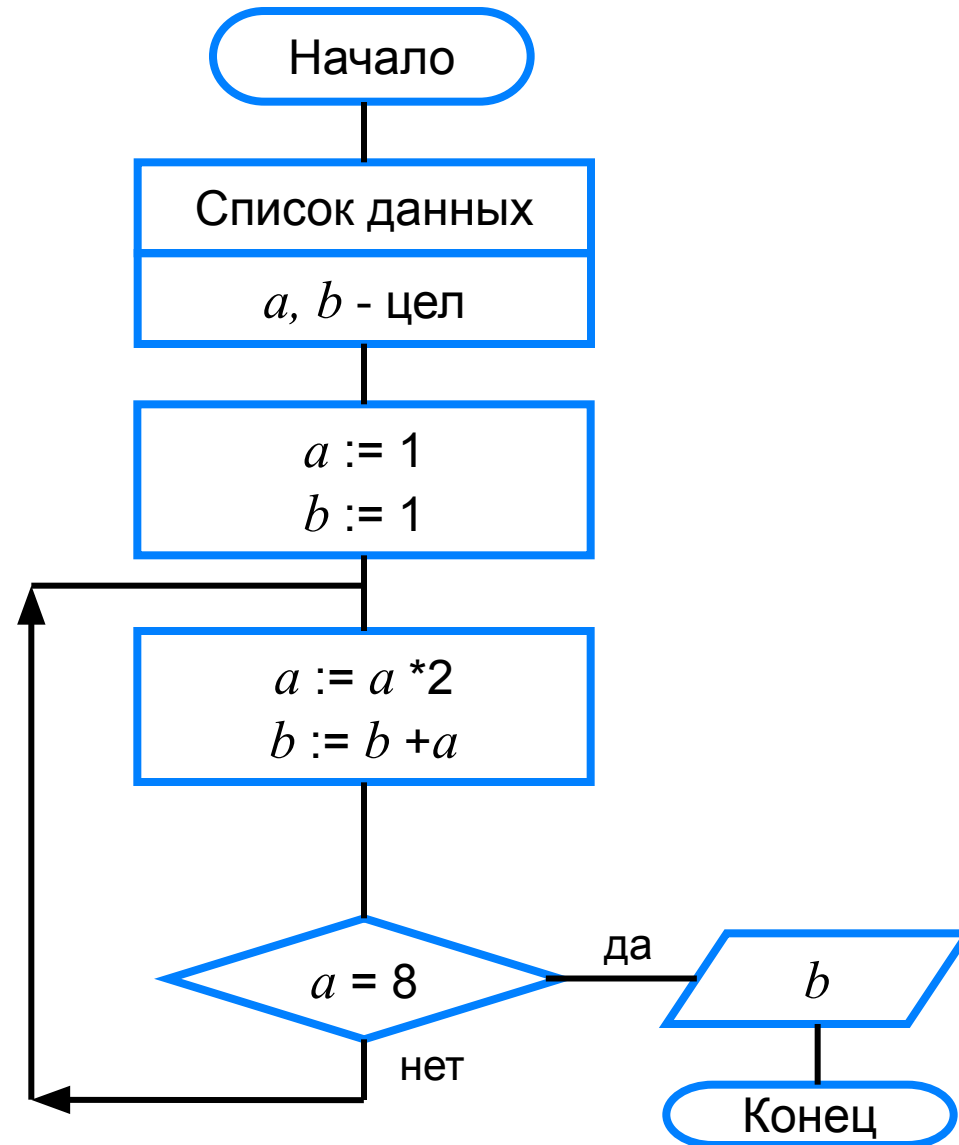
## цикл-ДО, цикл с постусловием



## (цикл-ДЛЯ, цикл с параметром)



# Вычисление значения переменной $b$



# Таблица значений переменных

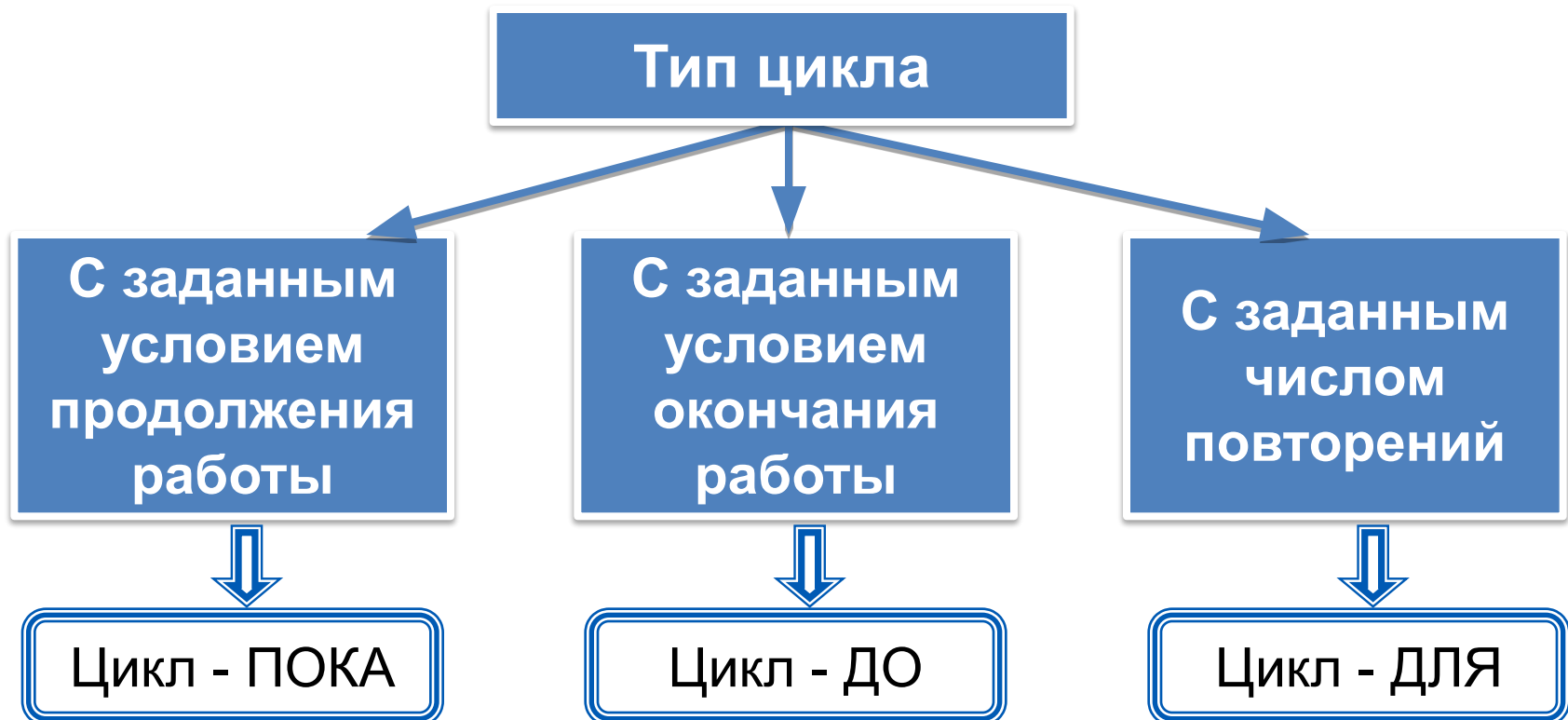
Шаг алгоритма	Операция	Переменные		Условие
		$a$	$b$	$a = 8$
1	$a := 1$	1		
2	$b := 1$	1	1	
3	$a := a * 2$	2	1	
4	$b := b + a$	2	3	
5	$a = 8$			2 = 8 (Нет)
6	$a := a * 2$	4	3	
7	$b := b + a$	4	7	
8	$a = 8$			4 = 8 (Нет)
9	$a := a * 2$	8	7	
10	$b := b + a$	8	15	
11	$a = 8$			8 = 8 (Да)

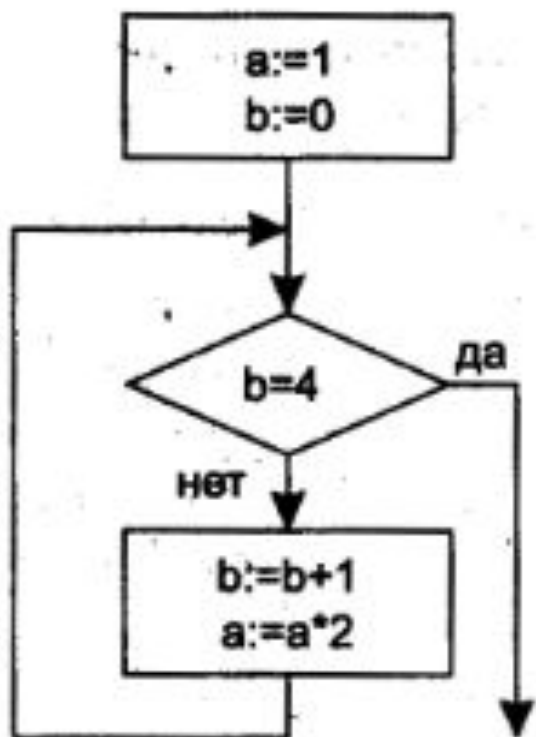
# Опорный конспект

**Повторение** - алгоритмическая конструкция, представляющая собой последовательность действий, выполняемых многократно.

Алгоритмы, содержащие конструкцию «повторение», называют **циклическими** или **циклами**.

Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется **телом цикла**.





# Электронные образовательные ресурсы

1. <http://school-collection.edu.ru/catalog/res/07e215ef-cd48-450d-8cf4-f5777cd832b2/?interface=catalog> – решето Эратосфена
2. [http://files.school-collection.edu.ru/dlrstore/58e9a0c3-11df-4c94-a5eb-b0a7b359ea35/9\\_32.swf](http://files.school-collection.edu.ru/dlrstore/58e9a0c3-11df-4c94-a5eb-b0a7b359ea35/9_32.swf) - исполнители алгоритмов