

# **Свойства алгоритма и его исполнители**

# Происхождение понятия «алгоритм»

---



**Algorithmi** – латинское  
написание имени  
выдающегося математика  
средневекового Востока  
**Муххамеда бен Мусы аль-  
Хорезми** (787-850)

Им были предложены приемы выполнения арифметических вычислений с многозначными числами.

# Исполнитель алгоритма

---

**Исполнитель алгоритма** – тот объект, для управления которым составлен алгоритм.

**Система команд исполнителя (СКИ)** – это вся совокупность команд, которые исполнитель умеет выполнять (понимает).

**Среда исполнителя** – обстановка, в которой функционирует исполнитель.

# Типы исполнителей

---

## Исполнители

```
graph TD; A[Исполнители] --> B[Формальные]; A --> C[Неформальные];
```

### Формальные

В роли исполнителя чаще всего выступает техническое устройство.

За действия исполнителя отвечает управляющий им объект.

Всегда одинаково выполняет одну и ту же команду

### Неформальные

В роли исполнителя чаще всего выступает человек.

За действия исполнителя отвечает сам исполнитель.

Не всегда может выполнять одни и те же команды совершенно одинаково.

# Исполнители алгоритма

---



**Назовите исполнителей и их тип:**

- Уборка мусора во дворе
- Обучение детей в школе
- Вождение автомобиля
- Ответ у доски
- Приготовление пицци
- Печать документа на принтере

# Человек – исполнитель алгоритмов

---



Выполните следующий алгоритм:

1. Задумайте число
2. Прибавьте столько же
3. Умножьте результат на 3
4. Разделите на задуманное число
5. Запишите ответ в тетрадь



Если после выполненных действий ответ совпал с вашим, значит, алгоритм был выполнен верно

# Свойства алгоритма

---

## 1. Дискретность

Процесс решения задачи должен быть разбит на последовательность отдельно выполняемых шагов

# Свойства алгоритма

---

## 1. Дискретность

Запишем алгоритм решения примера

$$(80+10)-5*(3+5)$$

1. Вычислить  $80+10$
2. Вычислить  $3+5$
3. Умножить 5 на результат 2 действия
4. Вычесть из результата 1 действия результат 3 действия



Если в данном алгоритме начать выполнять 4 действие, не дожидаясь окончания 3, то результат не может быть получен



# Свойства алгоритма

---

## 2. Понятность

Алгоритм, составленный для конкретного исполнителя, должен включать только те команды, которые входят в СКИ.

**Система команд исполнителя (СКИ)** – это вся совокупность команд, которые исполнитель умеет выполнять (понимает).

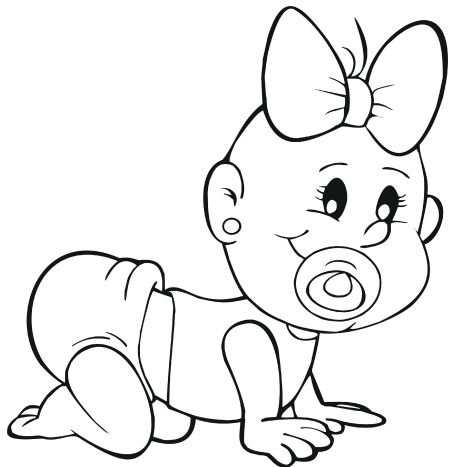
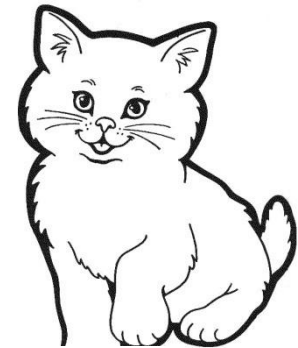
# Свойства алгоритма

## 2. Понятность



Кто сможет выполнить следующий алгоритм:

1. Пойти на кухню
2. Вскипятить чайник
3. Насыпать в чашку 1 чайную ложку коф
4. Насыпать в чашку 2 чайных ложки сахара
5. Налить в чашку кипяченой воды



# Свойства алгоритма

---

## 3. Точность

Каждая команда алгоритма должна определять однозначное действие исполнителя.

# Свойства алгоритма

## 3. Точность

Подскажите, как добраться до стадиона?



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



# Свойства алгоритма

---

## 4. Конечность (результативность)

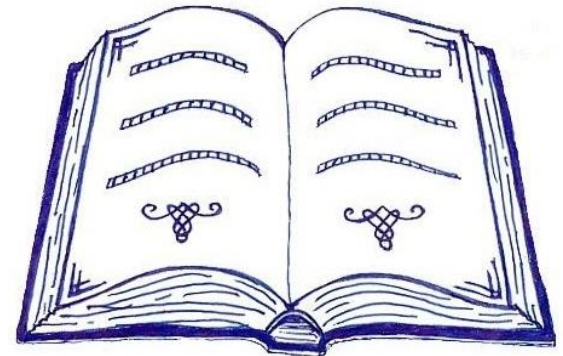
Исполнение алгоритма должно завершиться за конечное число шагов и должно привести к конкретному результату

# Свойства алгоритма

---

## 4. Конечность (результативность)

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



**Как изменить алгоритм, чтобы он стал конечным?**

# Свойства алгоритма

---

## 5. Полнота

Для успешного выполнения работы алгоритма, решения задачи необходимо сообщить (передать) исполнителю полный набор исходных данных.

# Свойства алгоритма

## 5. Полнота



Хватит ли денег на покупку продуктов, если будете вычислять общую стоимость по следующему алгоритму:

1. Умножить стоимость 1 кг сахарного песка на 2
2. Умножить стоимость 1 кг муки на 3
3. Умножить стоимость 1 батона на 2
4. Сложить все полученные результаты





# Свойства алгоритма

---

## 6. Массовость

Алгоритм может быть применен для целого класса задач, которые будут отличаться только исходными данными

# Определение алгоритма

---

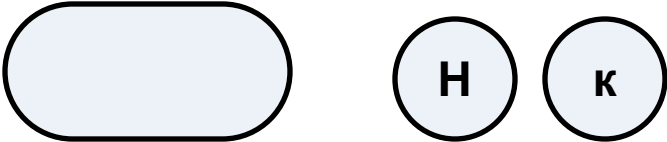
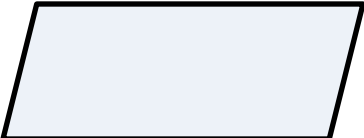
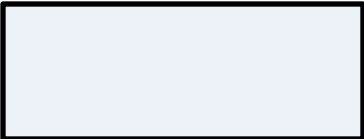
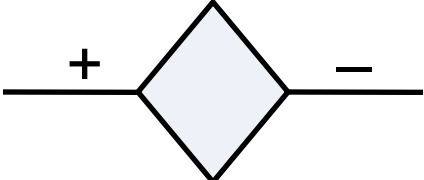
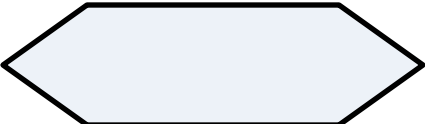
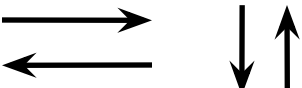
**Алгоритм** – это понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящую от исходных данных к искомому результату

# Способы представления алгоритма

---

1. **Словесный** (на естественном языке)
2. **Блок-схема** – графическое изображение алгоритма, в котором каждое элементарное действие представляется в виде графического знака.
3. С помощью **формул**
4. **Программа** – алгоритм, записанный на языке исполнителя.

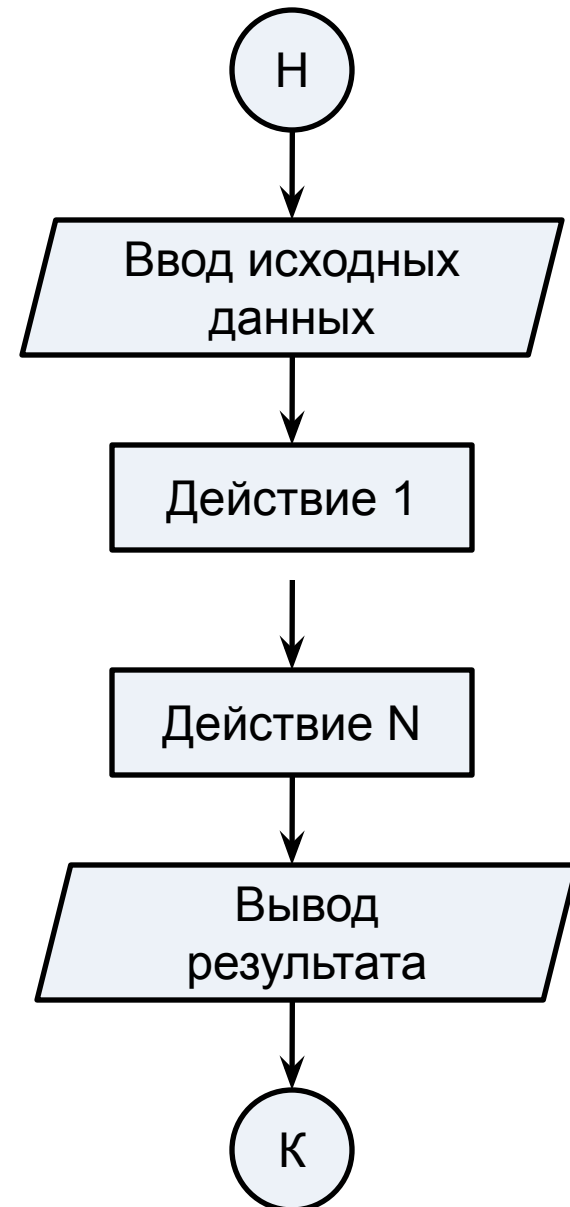
# Блок-схемы алгоритмов

Обозначение	Предназначение
	Начало, конец алгоритма
	Ввод, вывод данных
	Вычисление
	Проверка условия
	Счетчик количества повторов
	Соединяющие линии

# Виды алгоритмов

---

1. **Линейный алгоритм** – это алгоритм, в котором команды выполняются последовательно одна за другой



# Виды алгоритмов. Линейный алгоритм

---

Задание 1а. Длина прямоугольника 10 см. Ширина на 4 см меньше. Найдите периметр прямоугольника

Составление плана решения (алгоритм для себя)

1. Найти ширину

$$10-4=6$$

2. Найти периметр по формуле  $P=2*(a+b)$

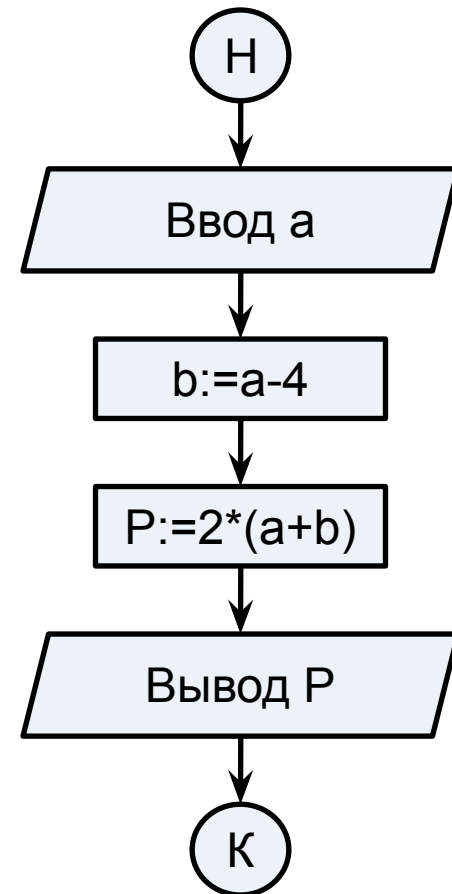
$$2*(10+6)=32$$

# Виды алгоритмов. Линейный алгоритм

Задание 1б. Составить алгоритм нахождения периметра прямоугольника для всех случаев, где ширина меньше длины на 4.

(алгоритм для другого)

1. Ввести  $a$
2.  $b = a - 4$
3.  $P = 2 * (a + b)$
4. Вывести  $P$



# Виды алгоритмов. Линейный алгоритм

---

Задание 2. Автомобиль 1 участок дороги преодолел за 2 часа, двигаясь со скоростью 60 км/ч. Оставшуюся часть пути в 80 км он преодолел за 1 час. Определите среднюю скорость автомобиля.

1. Найти  $S_1$

$$S_1 = 2 * 60$$

2. Найти  $S$

$$S = S_1 + 80$$

3. Найти  $t$

$$t = 2 + 1$$

4. Найти  $V_{cp}$

$$V_{cp} = S / t$$



# Виды алгоритмов. Линейный алгоритм

## Задание 2.

1. Найти  $S1$

$$S1 = 2 * 60$$

2. Найти  $S$

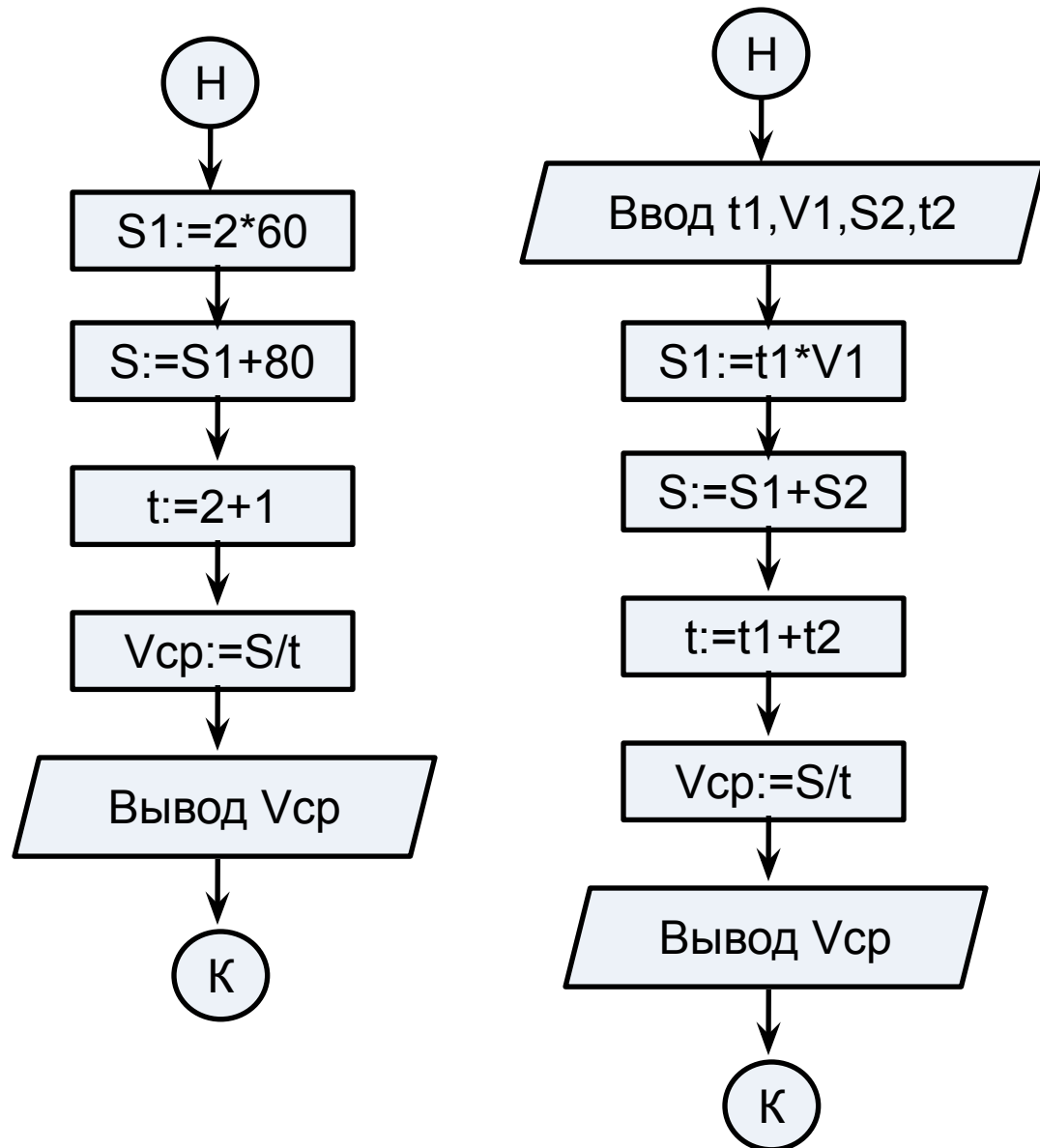
$$S = S1 + 80$$

3. Найти  $t$

$$t = 2 + 1$$

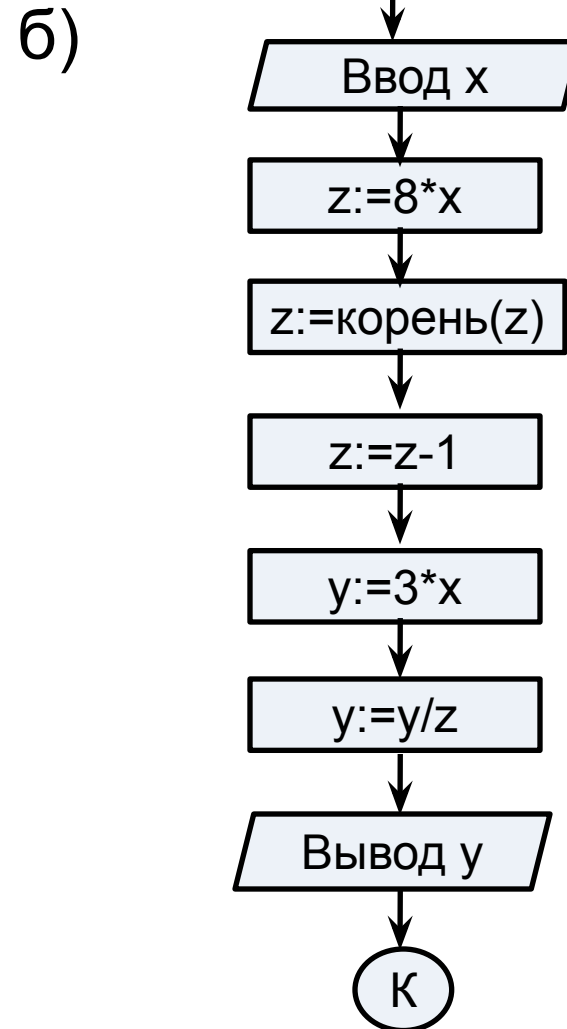
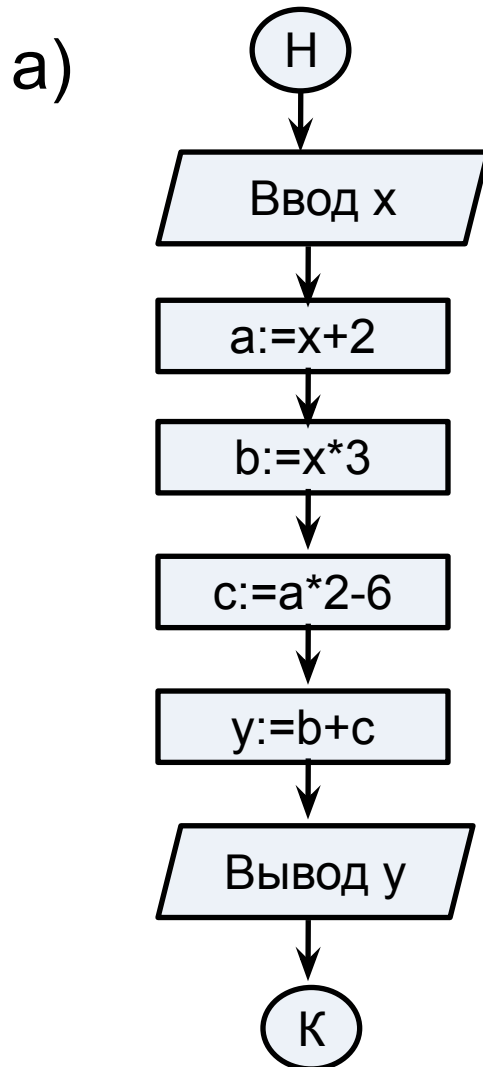
4. Найти  $V_{cp}$

$$V_{cp} = S / t$$



# Виды алгоритмов. Линейный алгоритм

Задание 3. Вычислите значение функции  $y$  при  $x=2$ , используя блок-схему алгоритма:



# Виды алгоритмов

---

Задание 4. Вычислите модуль числа  $a$ .

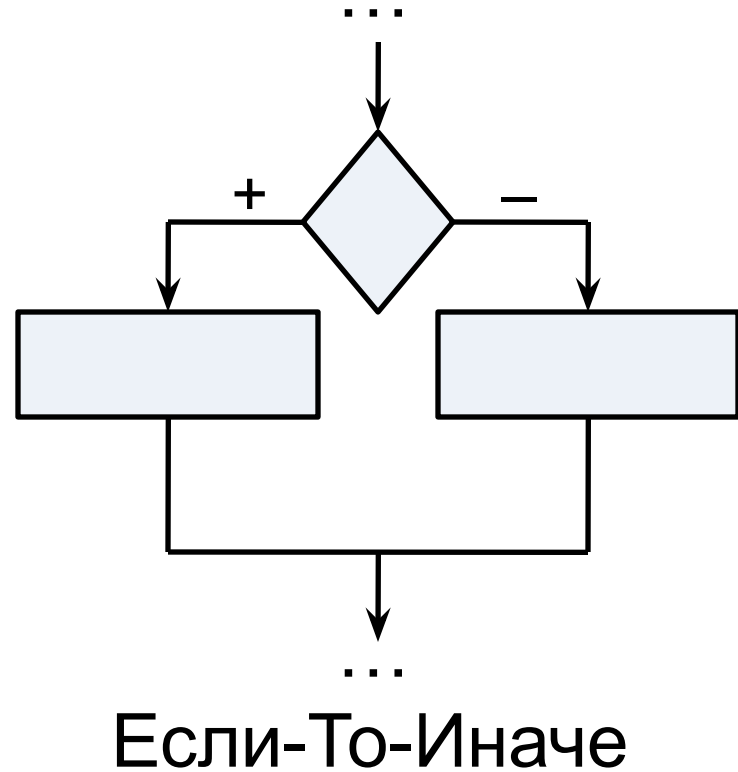
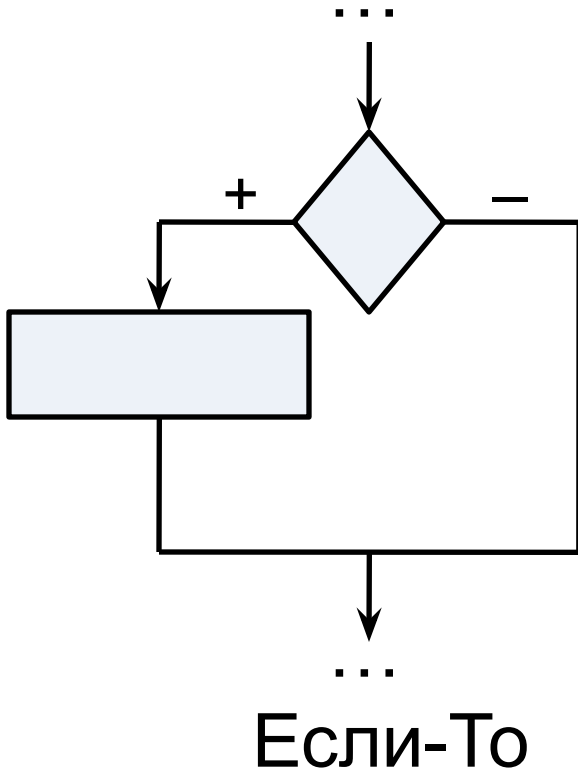
Составление плана решения (**алгоритм для себя**)

1. Ввести  $a$
2. Если  $a \geq 0$ , то  $m = a$ , иначе  $m = -a$
3. Вывод  $m$

**2. Алгоритм с ветвящейся структурой (разветвляющийся алгоритм)** – алгоритм, выбор следующего действия в котором осуществляется после проверки условия

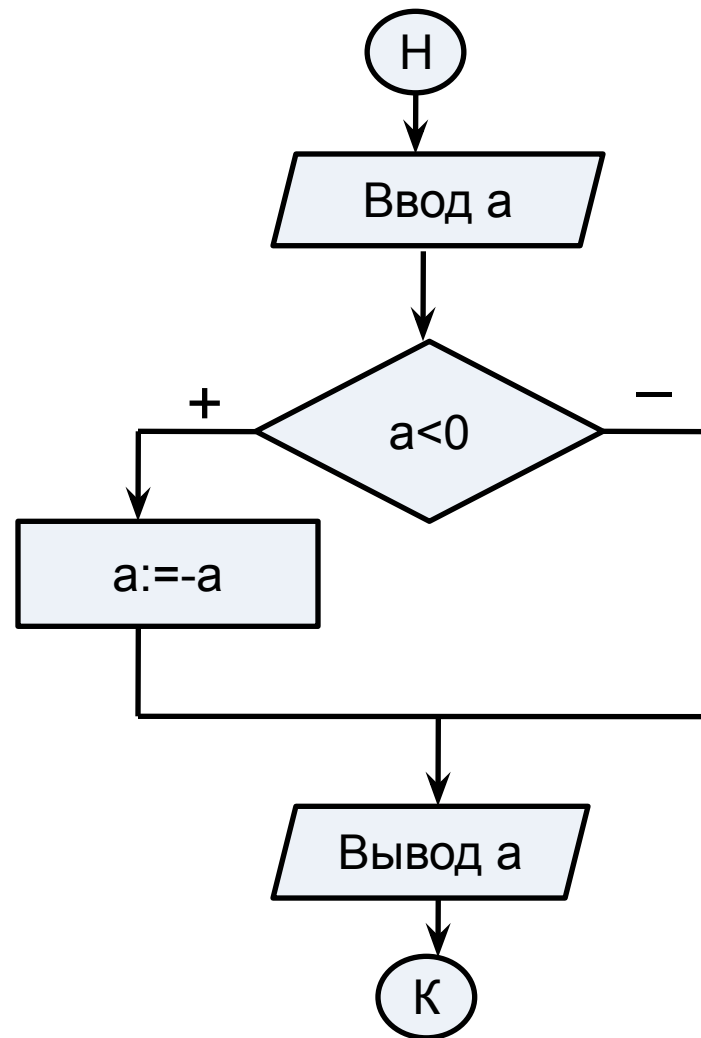
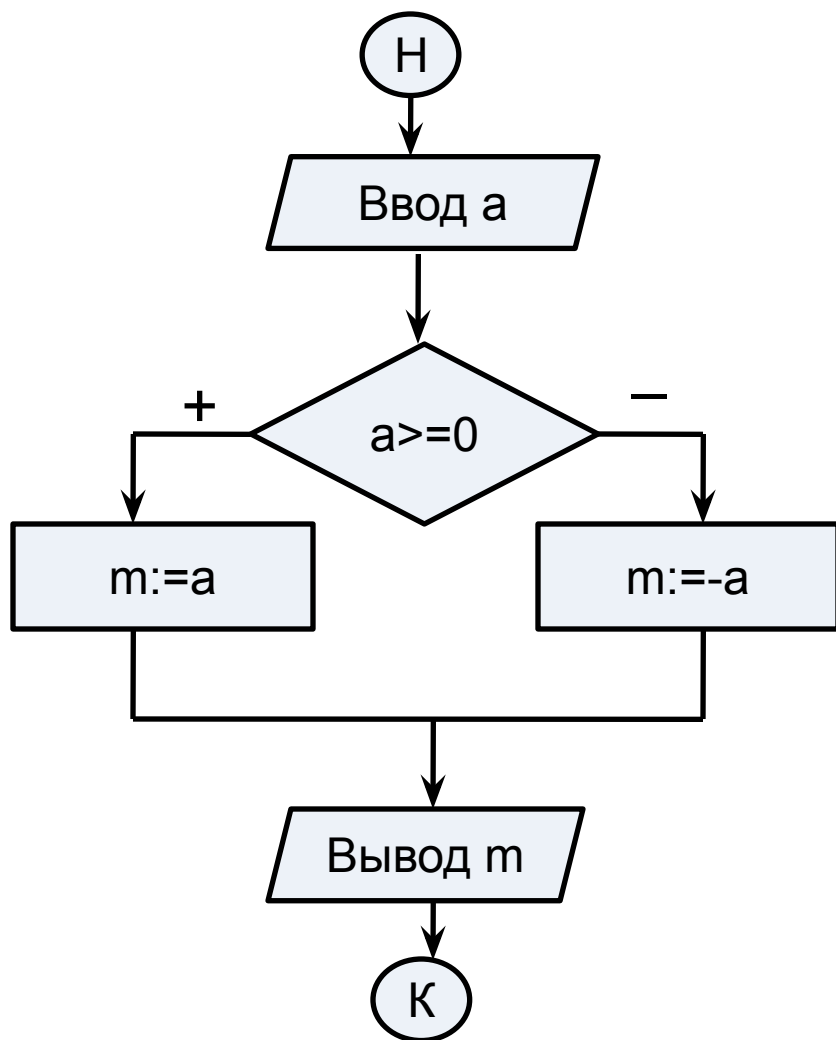
# Виды алгоритмов

**2. Алгоритм с ветвящейся структурой** – алгоритм, выбор следующего действия в котором осуществляется после проверки условия



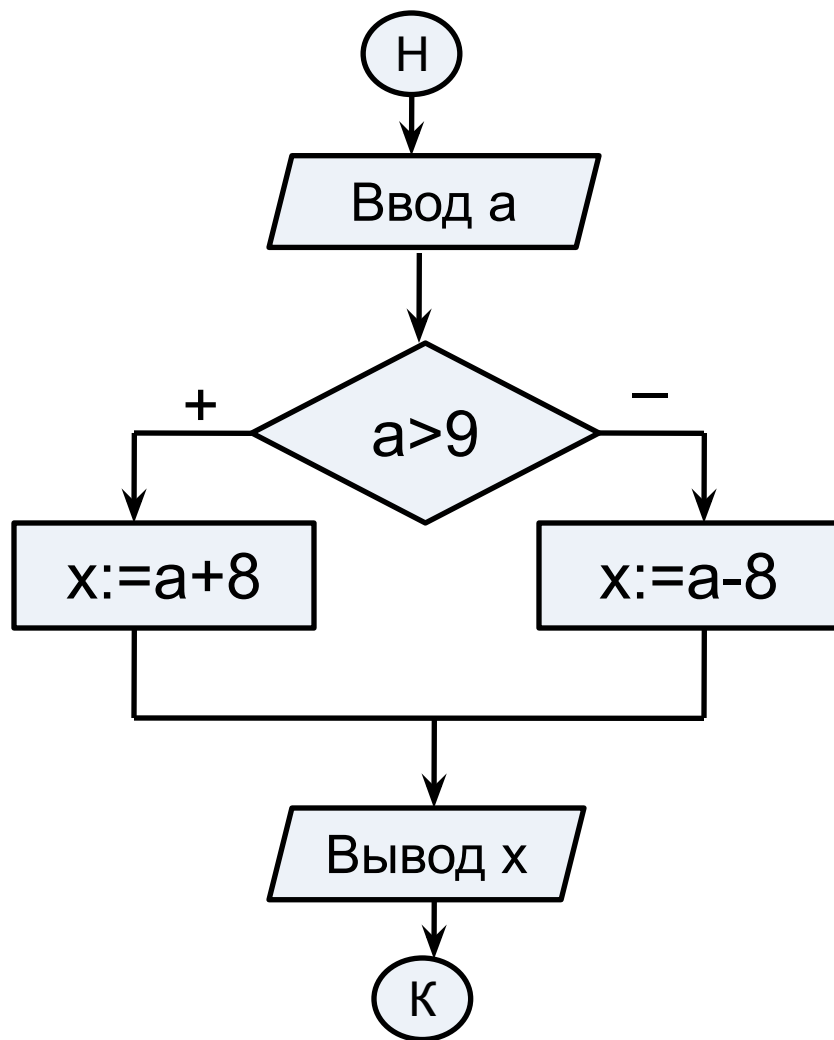
# Ветвящийся алгоритм

Задание 4. Вычислите модуль числа  $a$ .



# Ветвящийся алгоритм

Задание 5. Вычислите значение  $x$  по блок-схеме.



а)  $a=10$

б)  $a=4$

в)  $a=9$

# Ветвящийся алгоритм

---

Задание 6. Вычислите  $\min(a,b)$ .

Задание 7. Найдите значение  $y$ , если

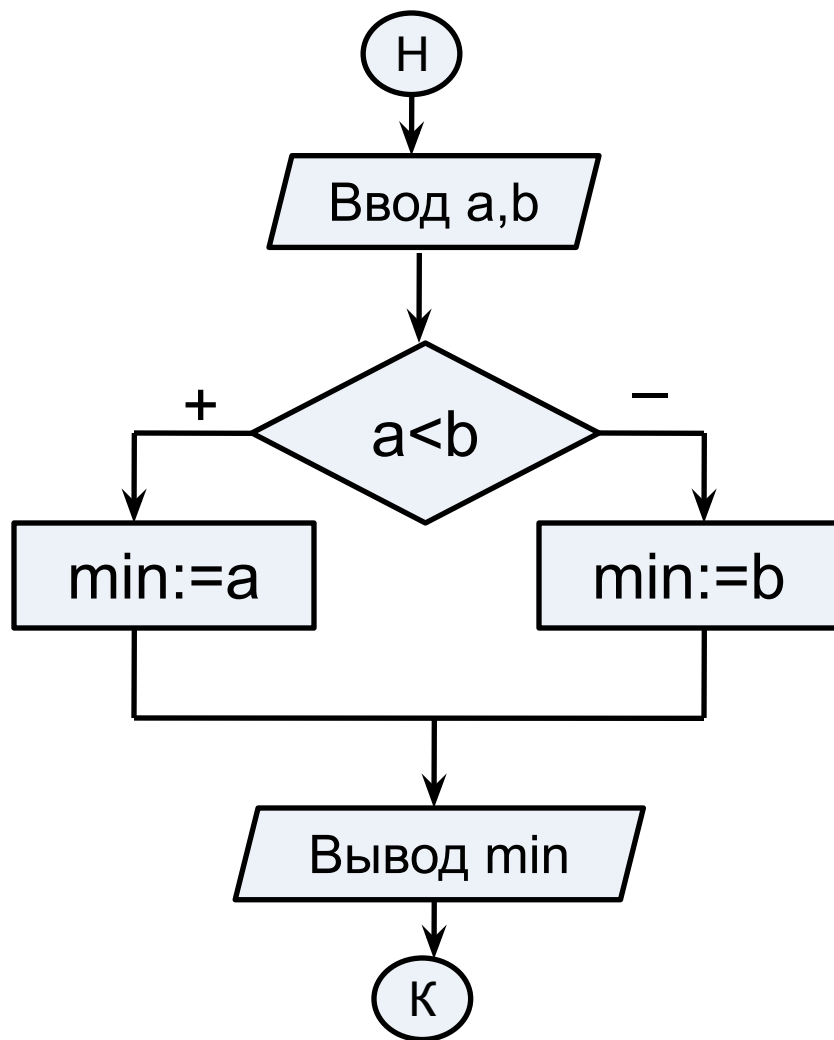
$$y = \begin{cases} 2 + x, & x \geq 0, \\ 3x, & x < 0. \end{cases}$$

Задание 8. Найдите значение  $y$ , если

$$y = \begin{cases} x, & 0 < x, \\ 2x, & -5 < x \leq 0, \\ 4 - x, & x \leq -5. \end{cases}$$

# Ветвящийся алгоритм

Задание 6. Вычислите  $\min(a,b)$ .

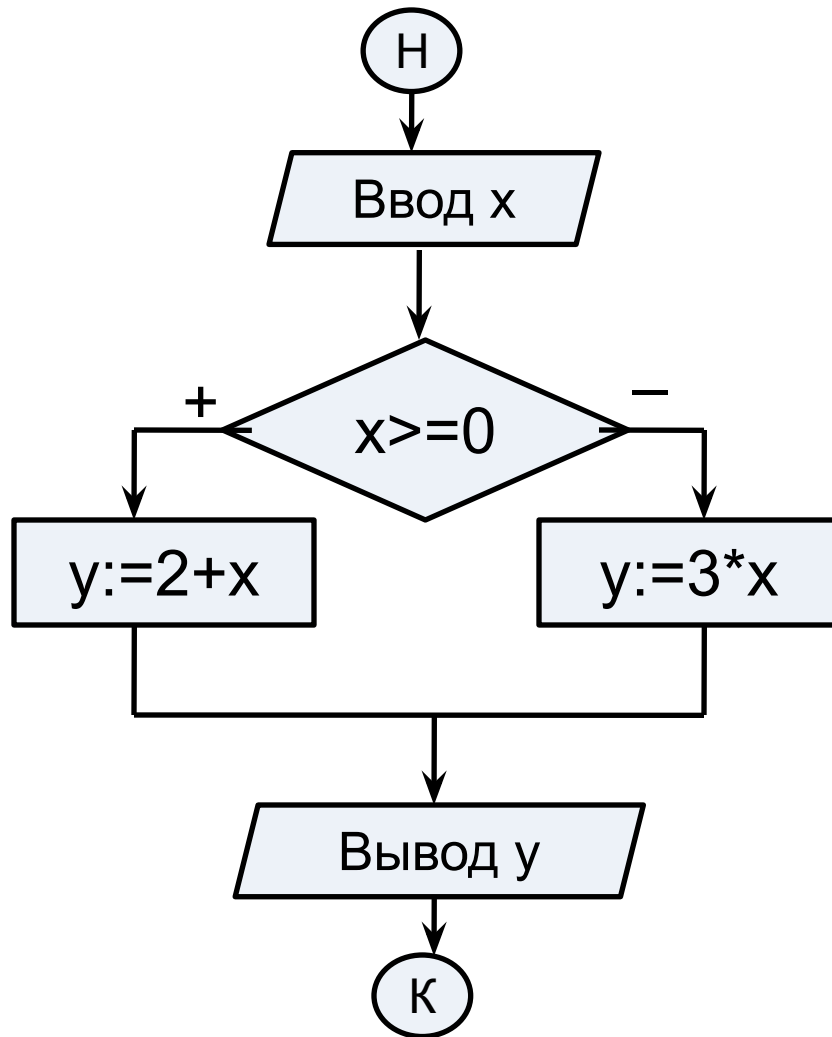




# Ветвящийся алгоритм

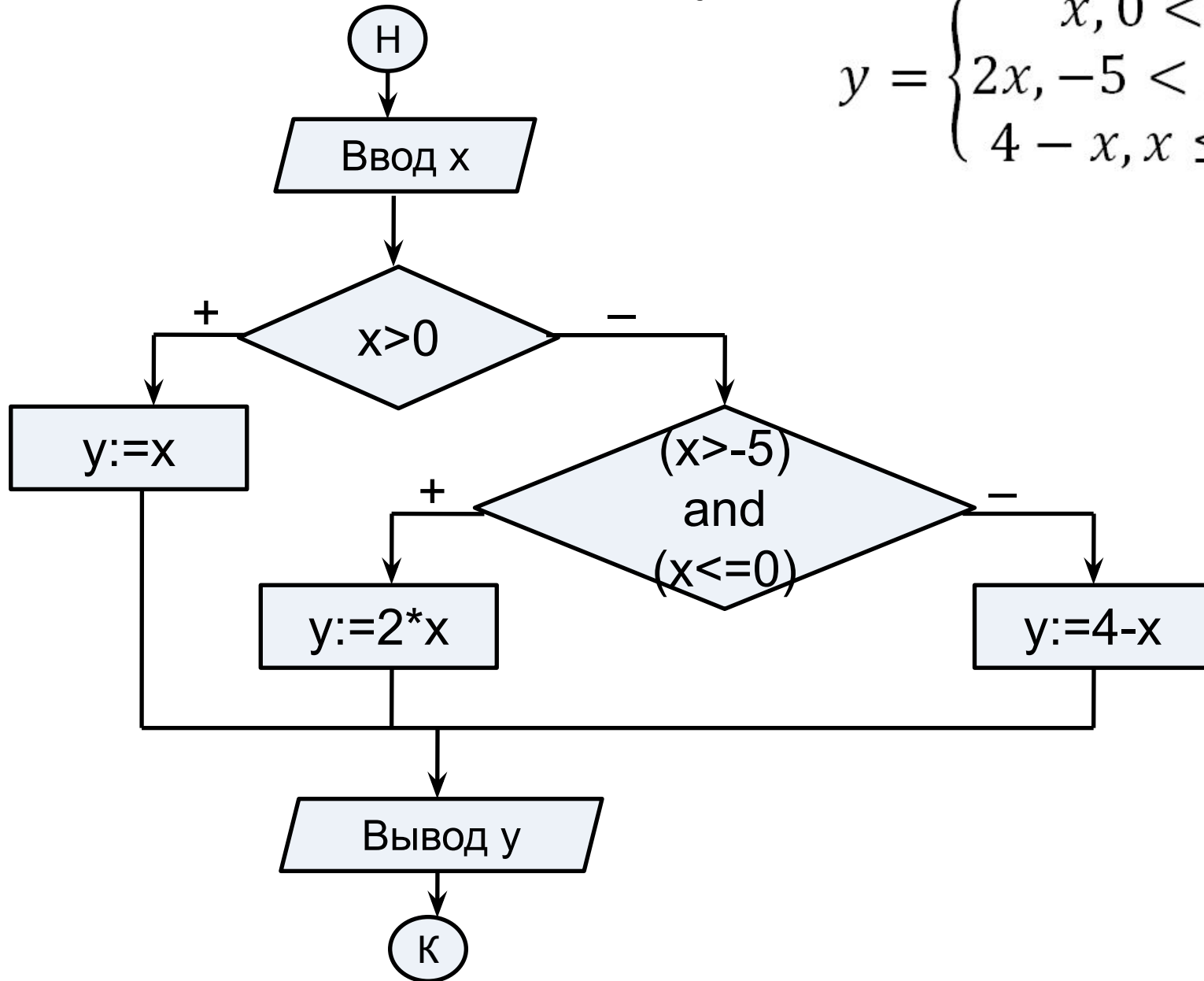
Задание 7. Найдите значение  $y$ , если

$$y = \begin{cases} 2 + x, & x \geq 0, \\ 3x, & x < 0. \end{cases}$$



Задание 8. Найдите значение  $y$ , если

$$y = \begin{cases} x, & 0 < x, \\ 2x, & -5 < x \leq 0, \\ 4 - x, & x \leq -5. \end{cases}$$



# Циклы

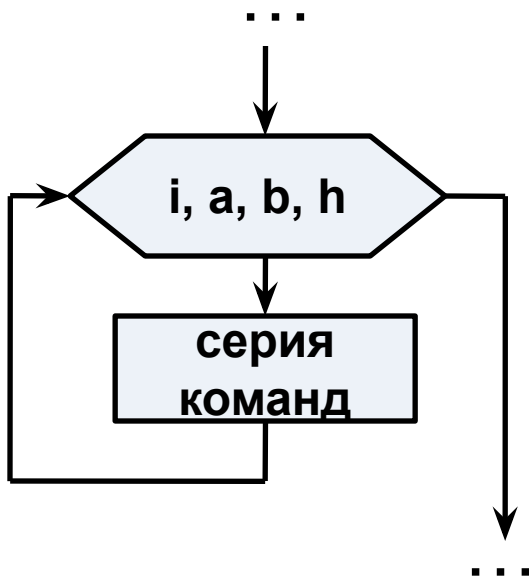
---

**Цикл** – это многократное выполнение одинаковых действий.

- **ЦИКЛ С ИЗВЕСТНЫМ ЧИСЛОМ ШАГОВ**
- **ЦИКЛ С НЕИЗВЕСТНЫМ ЧИСЛОМ ШАГОВ**  
(цикл с условием)

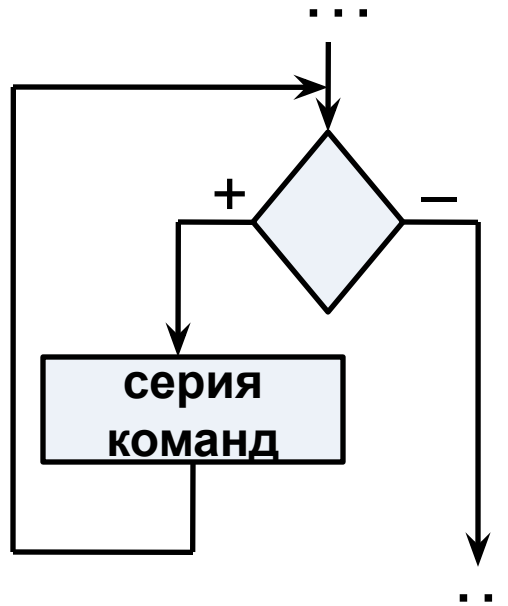
# Виды циклов

Цикл с  
параметром



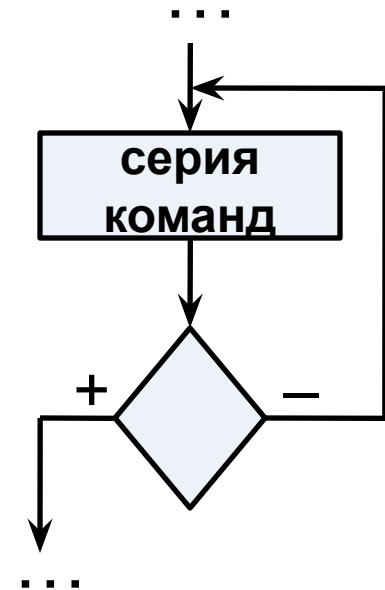
Цикл типа **для**

Цикл с  
предусловием



Цикл типа **пока**

Цикл с  
постусловием

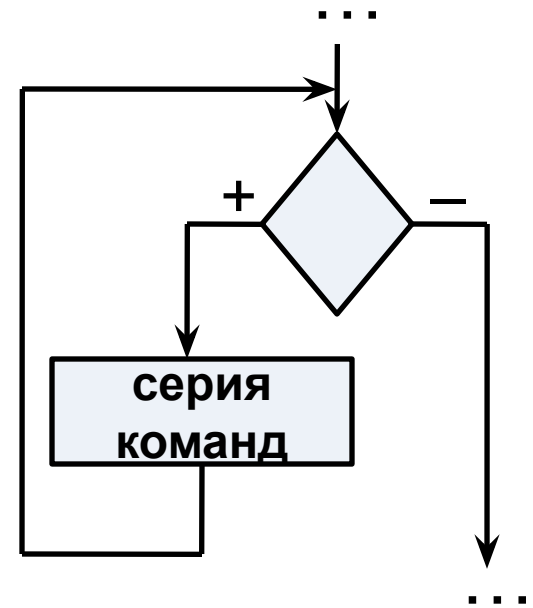


Цикл типа **до**

# Цикл с предусловием (типа Пока)

---

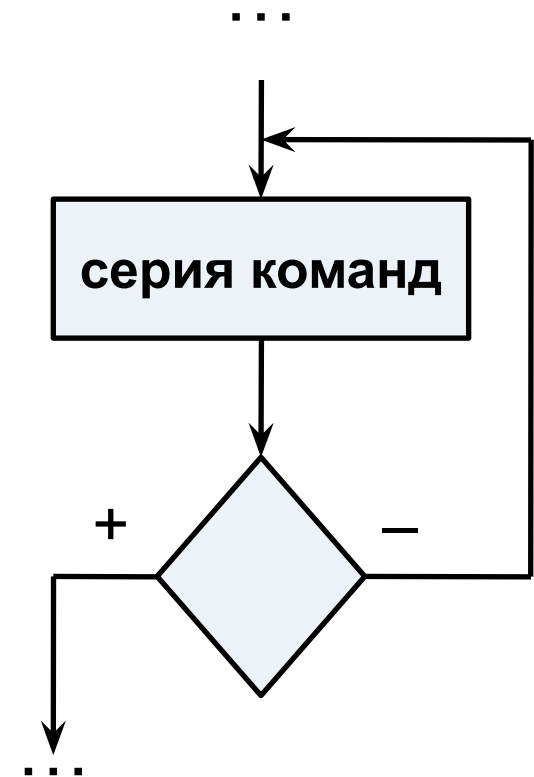
1. Цикл заканчивается, когда **условие** становится **не верным (ложным)**.
2. Если **условие** с самого начала ложно, то серия команд **не выполняется ни разу**.



# Цикл с постусловием (типа До)

---

1. Серия команд выполняется **хотя бы один раз**.
2. Цикл заканчивается, когда **условие становится верным (истинным)**.



# Задание

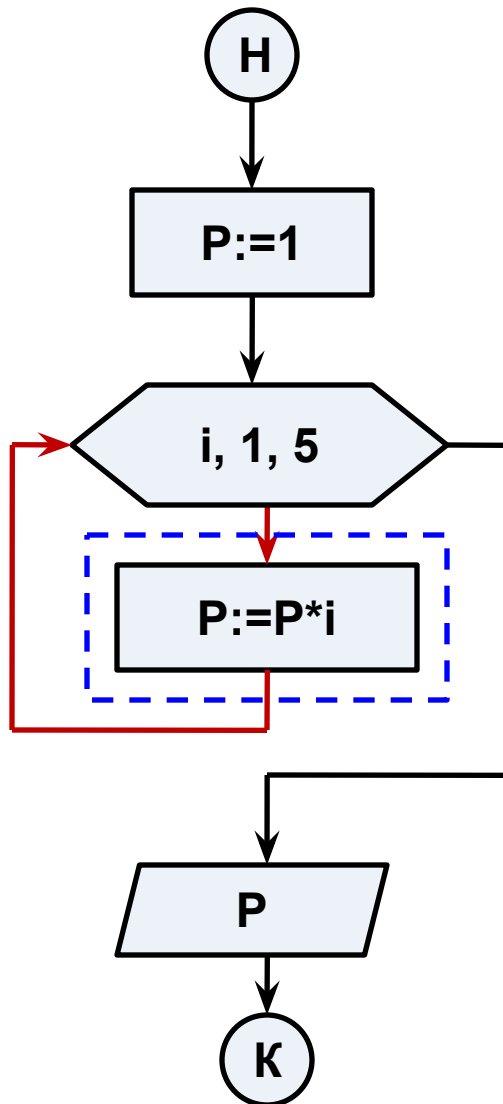
---

Вычислить произведение чисел **от 1 до 5**, используя различные варианты цикла.

Математическая модель:

$$P=1 \cdot 2 \cdot 3 \cdot 4 \cdot 5=120$$

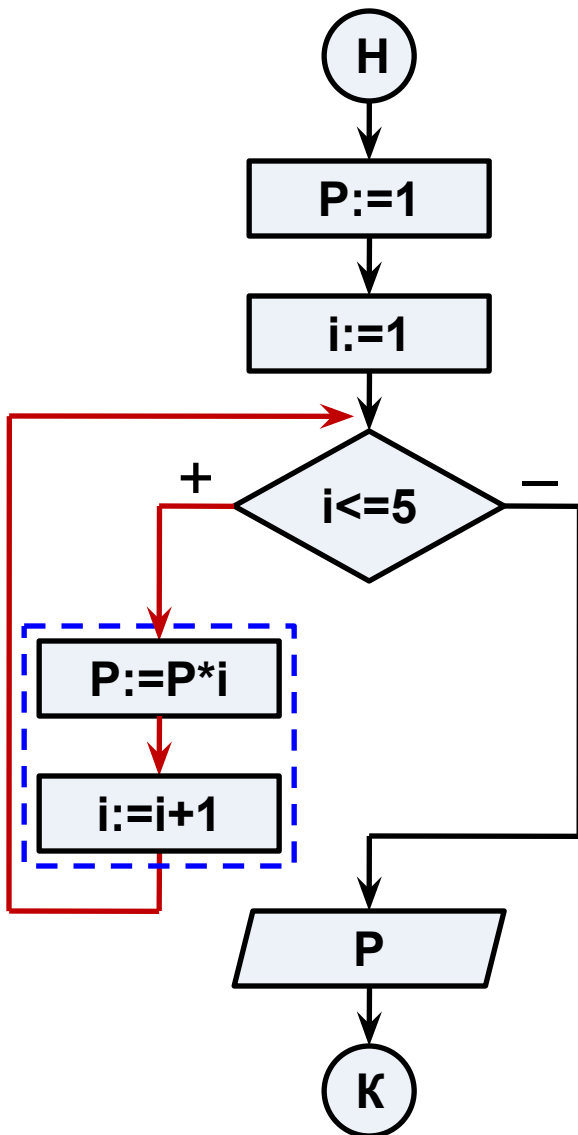
# Решение. Цикл с параметром



Шаг	Оп-ция	P	i	Проверка условия
1	P:=1			
2	i:=1 P:=P*i			
3	i:=2 P:=P*i			
4	i:=3 P:=P*i			
5	i:=4 P:=P*i			
6	i:=5 P:=P*i			

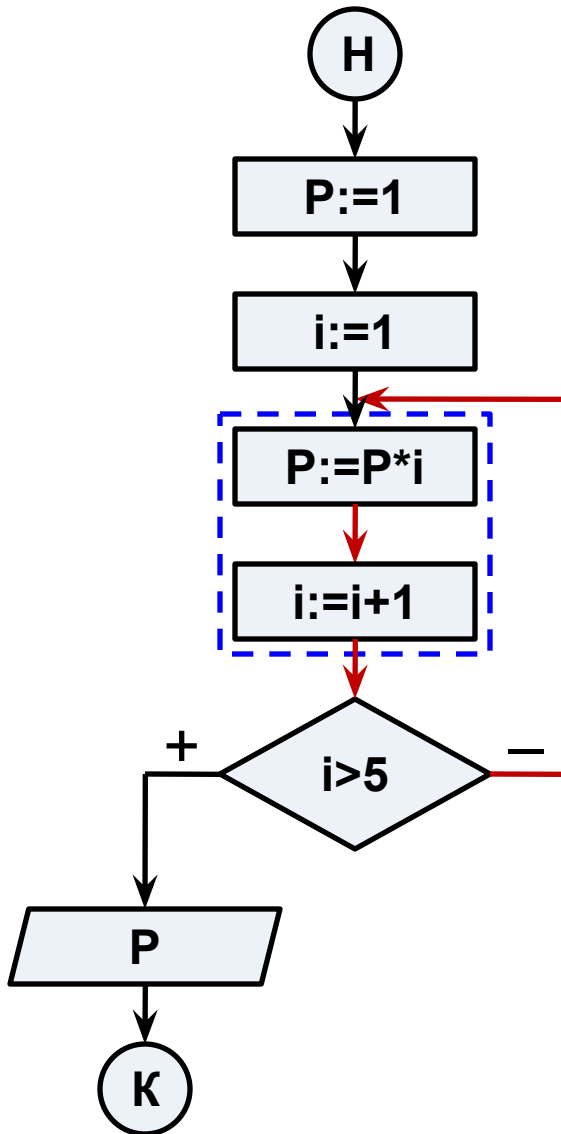


# Решение. Цикл с предусловием



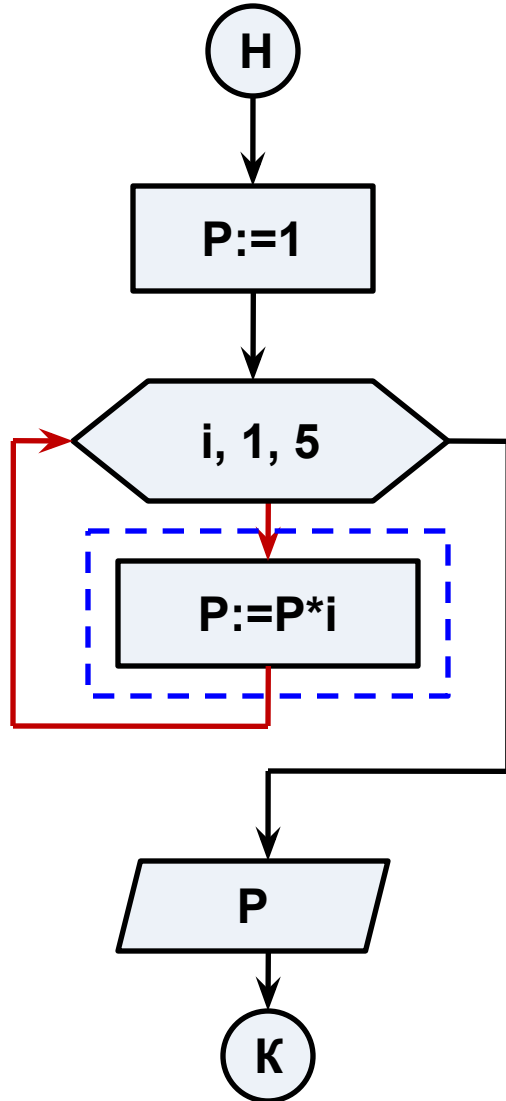
Шаг	Оп-ция	P	i	Проверка условия
1	P:=1			
2	i:=1			
3	i<=5 P:=P*i i:=i+1			
4	i<=5 P:=P*i i:=i+1			
5	i<=5 P:=P*i i:=i+1			
6	i<=5 P:=P*i i:=i+1			
7	i<=5 P:=P*i i:=i+1			
8	i<=5			

# Решение. Цикл с постусловием

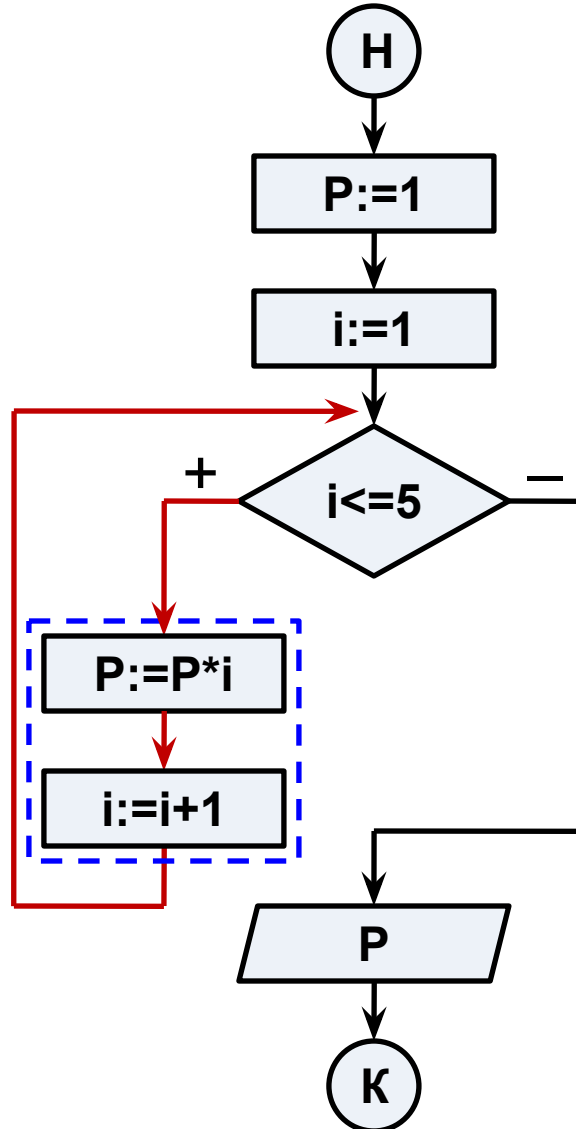


Шаг	Оп-ция	P	i	Проверка условия
1	P:=1			
2	i:=1			
3	P:=P*i i:=i+1 i>5			
4	P:=P*i i:=i+1 i>5			
5	P:=P*i i:=i+1 i>5			
6	P:=P*i i:=i+1 i>5			
7	P:=P*i i:=i+1 i>5			

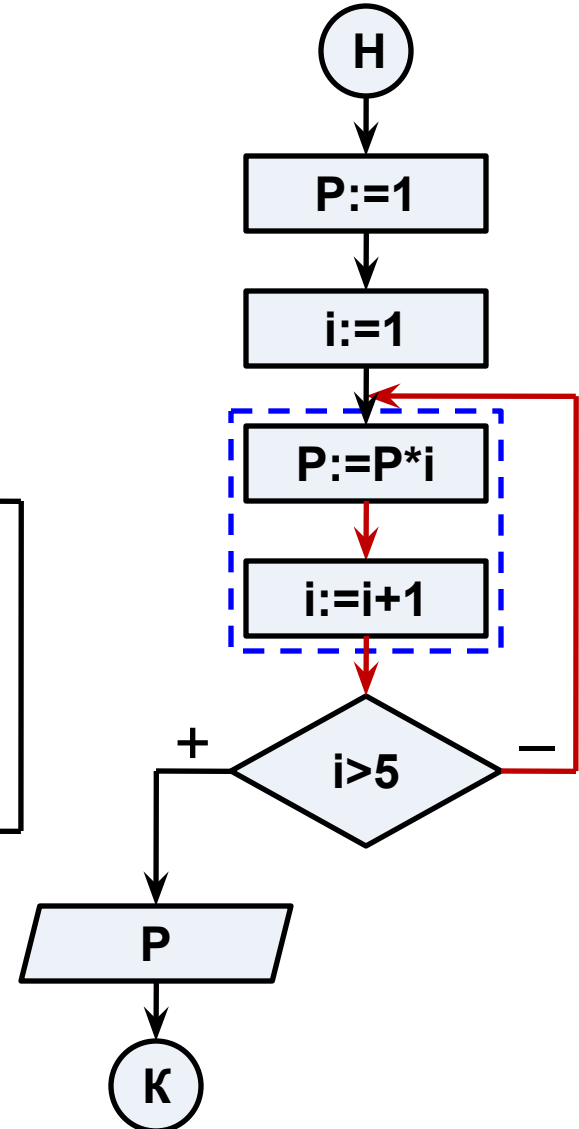
## Цикл с параметром



## Цикл с предусловием



## Цикл с постусловием



# Задания

---

1. Вычислить сумму чисел от 1 до 5, используя различные варианты цикла.

Математическая модель:

$$S=1+2+3+4+5=15$$

2. Вывести на экран числа от 1 до 5 в:

а) прямом порядке;

б) обратном порядке.

Математическая модель:

а) 1 2 3 4 5

б) 5 4 3 2 1

## Задания

---

3. Вычислить произведение 5 первых положительных четных чисел, используя различные варианты цикла.
4. Вычислить сумму всех нечетных целых чисел из промежутка  $[-7; 11]$ , используя различные варианты цикла.
5. Вывести на экран квадраты чисел от 1 до 15, используя различные варианты цикла.