



СЕВЕРНЫЙ (АРКТИЧЕСКИЙ)  
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ  
ИМЕНИ М.В. ЛОМОНОСОВА

# Понятие алгоритма. Блок-схемы и псевдокод

Латухина Екатерина Александровна,  
старший преподаватель кафедры прикладной математики  
и высокопроизводительных вычислений

# Содержание

- Алгоритм и его основные свойства. Способы описания алгоритмов
- Блок-схемы. ГОСТ 19.701-90. Псевдокод
- Базовые конструкции структурного программирования
- Примеры

# **Алгоритм и его основные свойства. Способы описания алгоритмов**

# Понятие алгоритма

- Последовательность инструкций, описывающих порядок действий, которые необходимо выполнить для решения поставленной задачи

# Свойства алгоритма

## Детерминированность (определенность)

- Вычислительный процесс ведет к однозначному результату

## Результативность

- Результат должен быть получен через конечное число шагов

## Массовость

- Алгоритм должен быть пригоден для решения всех задач данного типа

## Дискретность

- Алгоритм можно разделить на конечное число шагов (этапов)

## Задание

- Приведите пример алгоритма из любой области или из жизни и докажите, что он обладает озвученными ранее свойствами



# Способы описания алгоритмов

- Словесный
- Словесно-стилизированный (псевдокод)
- Граф
- Таблица
- Блок-схема
- Программный код
- Другие

Основные блоки и конструкции

**Блок-схемы.**


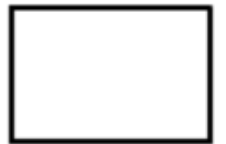


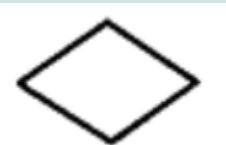
**ГОСТ 19.701-90. Псевдокод**





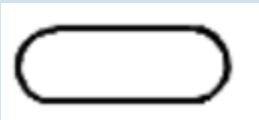
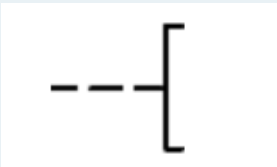
## Блок-схема

- Графическое изображение логической структуры алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций
- [ГОСТ 19.701-90](#)

## Основные блоки (1)

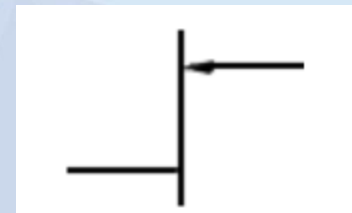
Название блока	Обозначение	Действие
Данные		Операции ввода-вывода
Процесс		Обработка данных (выполнение операции или группы операций)
Предопределенный процесс		Выполнение одной или нескольких операций, определенных в другом месте (вызов подпрограммы)
Подготовка		Циклы со счетчиком
Решение		Ветвление, выбор, циклы с предусловием и постусловием

## Основные блоки (2)

Название блока	Обозначение	Действие
Граница цикла		Любые циклы
Соединитель		Для обрыва линий и продолжения в другом месте
Терминатор		Начало или конец программы или подпрограммы
Комментарий		Добавление пояснений для объяснения или примечания

# Правила оформления блок-схем (1)

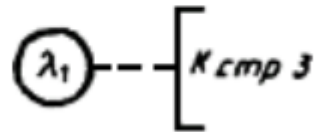
- Внутри блоков можно писать краткий пояснительный текст
- Если пояснительный текст большой, его нужно выносить в комментарий
- Направление потока слева направо и сверху вниз считается стандартным, стрелки часто не ставятся
- Если поток имеет направление справа налево или снизу вверх, стрелки обязательны!
- Следует избегать пересечения линий
- Линии изгибаются только под прямым углом
- Две или более входящие линии могут объединяться в одну исходящую линию
- Если две или более линии объединяются в одну линию, место объединения должно быть смещено



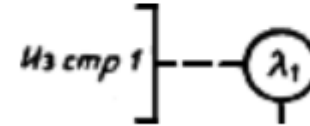
## Правила оформления блок-схем (2)

- При необходимости разрыва линии используются соединители
- Внутри соединителей ставится отметка

Внешний  
соединитель

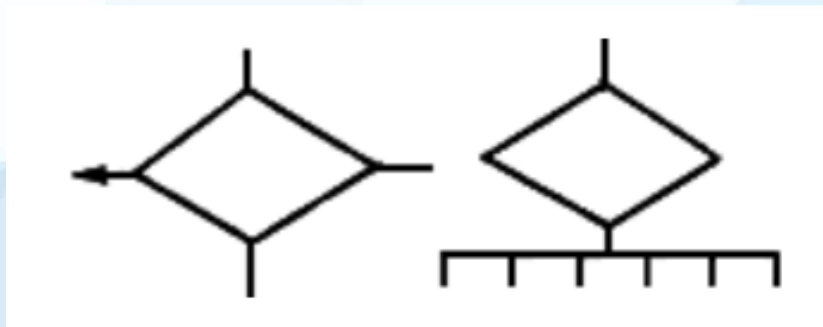


Внутренний  
соединитель



- Несколько выходов обозначаются одним из двух способов:

Сколько здесь  
выходов?



## Псевдокод

- Компактный язык описания алгоритмов, использующий ключевые слова императивных языков программирования, но опускающий несущественные подробности и специфический синтаксис
- Основные ключевые слова псевдокода:
  - Ввод
  - Вывод
  - Начало
  - Конец
  - Если
  - Иначе
  - Для
  - Пока
  - Выполнять
  - Подпрограмма
  - И др.
- Вместо операторных скобок используются отступы

Основные и дополнительные конструкции

# Базовые конструкции структурного программирования

# Структурное программирование

- Методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков
- Предложена в 1970-х годах Э. Дейкстрой



# Теорема о структурном программировании

Любая программа, заданная в виде блок-схемы, может быть представлена с помощью трех управляющих структур:

- **последовательность**
- **ветвление**
- **цикл**

# Принципы структурного программирования

1. Отказ от оператора безусловного перехода goto
2. Программа строится из трёх базовых конструкций: следование, ветвление, цикл
3. Конструкции могут быть вложены друг в друга
4. Законченные фрагменты программы оформляются в виде подпрограмм (процедур и функций)
5. Логически законченные группы инструкций оформляются в виде блоков
6. Все конструкции имеют один вход и один выход
7. Разработка программы ведётся пошагово, методом «сверху вниз»

## Блок

- Логически сгруппированная часть исходного кода
- К блоку инструкций следует обращаться как к единой инструкции
- Служат для ограничения области видимости переменных и функций
- Могут быть вложенными один в другой
- Границы блока строго определены в зависимости от языка программирования (BEGIN ... END, { ... }, отступы и др.)

# Основные конструкции структурного программирования

## Следование

- Однократное выполнение операций в том порядке, в котором они записаны в тексте программы

## Ветвление

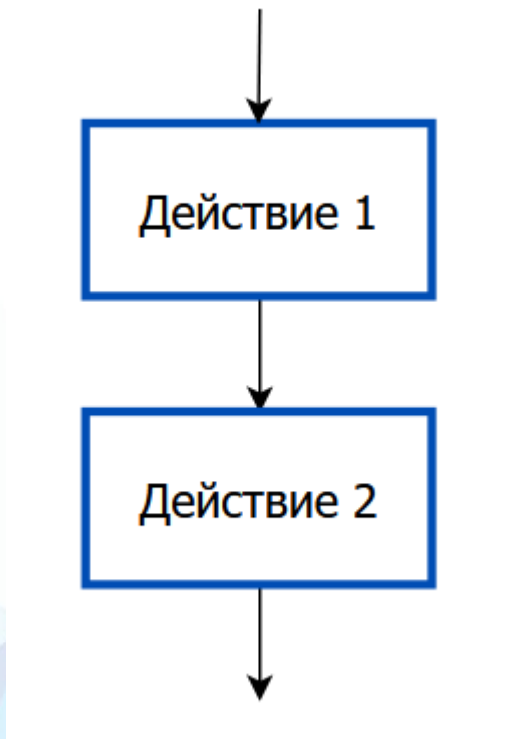
- Однократное выполнение операций в зависимости от выполнения заданного условия

## Цикл-пока

- Многократное исполнение одних и тех же операций пока выполняется заданное условие

# Следование (последовательный алгоритм)

## Блок-схема



## Псевдокод

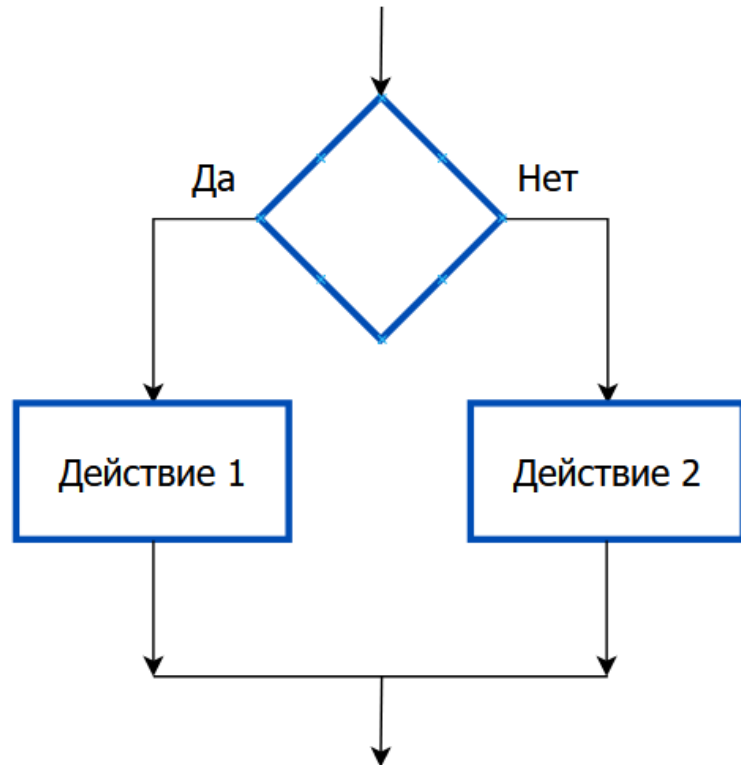
<Действие 1>

<Действие 2>

Здесь и далее то, что пишется  
**<внутри угловых скобок>**,  
в реальном примере заменяется  
на реальные значения

# Ветвление (разветвляющийся алгоритм, условный алгоритм)

## Блок-схема



## Псевдокод

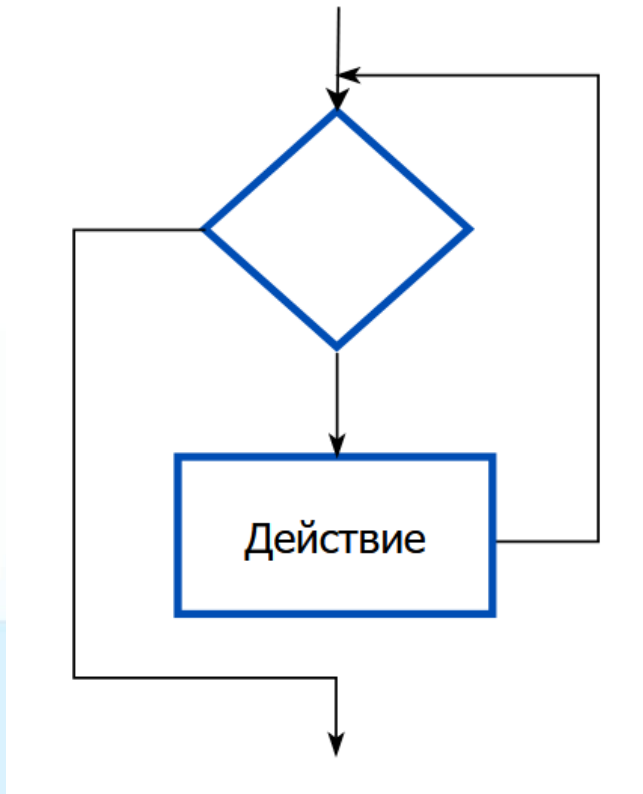
```

Если <Условие>
    <Действие 1>
[Иначе
    <Действие 2>]
    
```

Здесь и далее **[квадратные скобки]** означают необязательность написанного внутри них

# Цикл с предусловием (цикл-пока)

## Блок-схема



## Псевдокод

Пока <Условие>  
    <Действие>

# Дополнительные конструкции структурного программирования

## Множественное ветвление

- Однократное выполнение операций в зависимости от выполнения заданных условий

## Цикл-до

- Многократное исполнение одних и тех же операций до тех пор, пока выполняется заданное условие

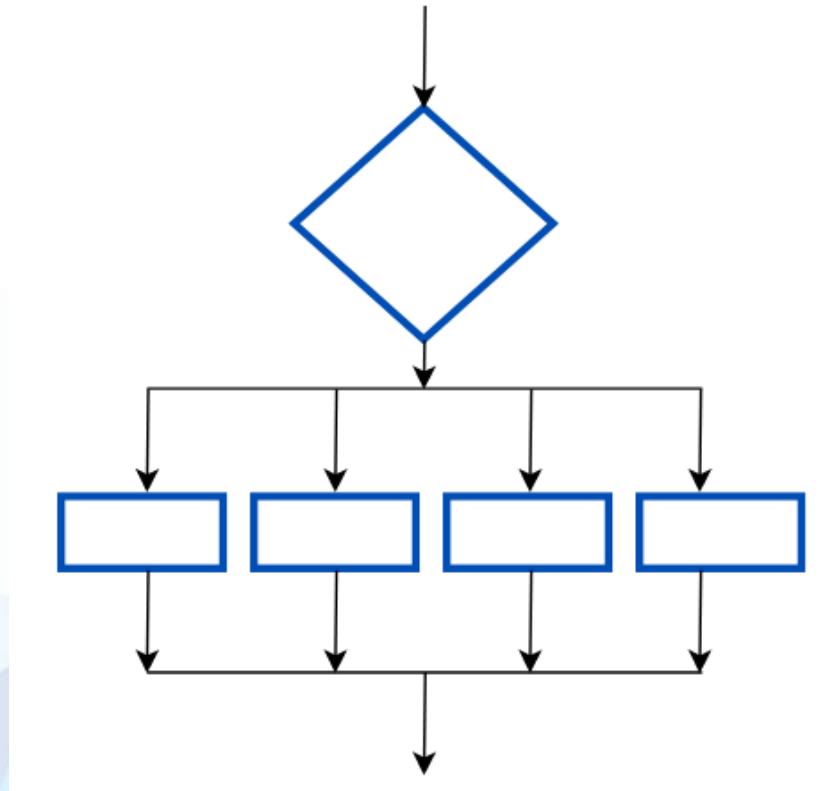
## Цикл со счетчиком

- Многократное исполнение одних и тех же операций, пока переменная-счетчик находится в заданном диапазоне



# Множественное ветвление

## Блок-схема



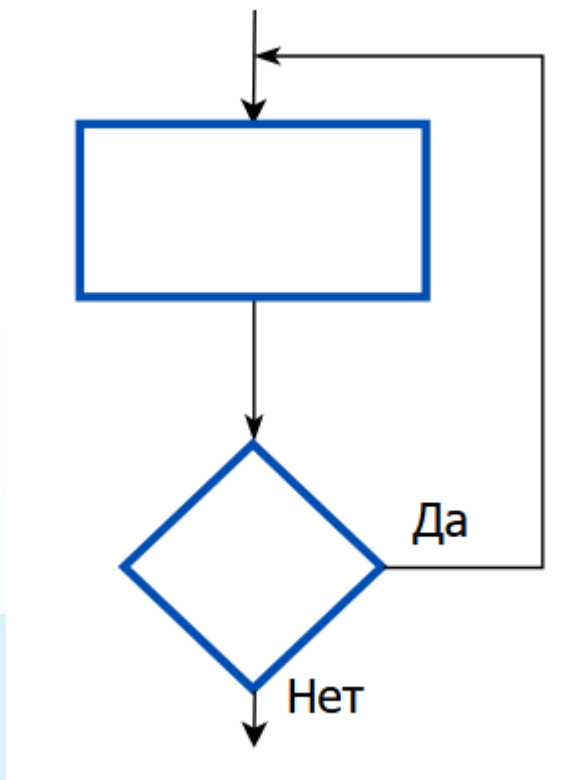
## Псевдокод

```

Если <Условие 1>
    <Действие 1>
[Иначе если <Условие 2>
    <Действие 2>]
[Иначе если <Условие 3>
    <Действие 3>]
...
[Иначе
    <Действие n>]
  
```

## Цикл с постусловием (цикл-до)

### Блок-схема



### Псевдокод

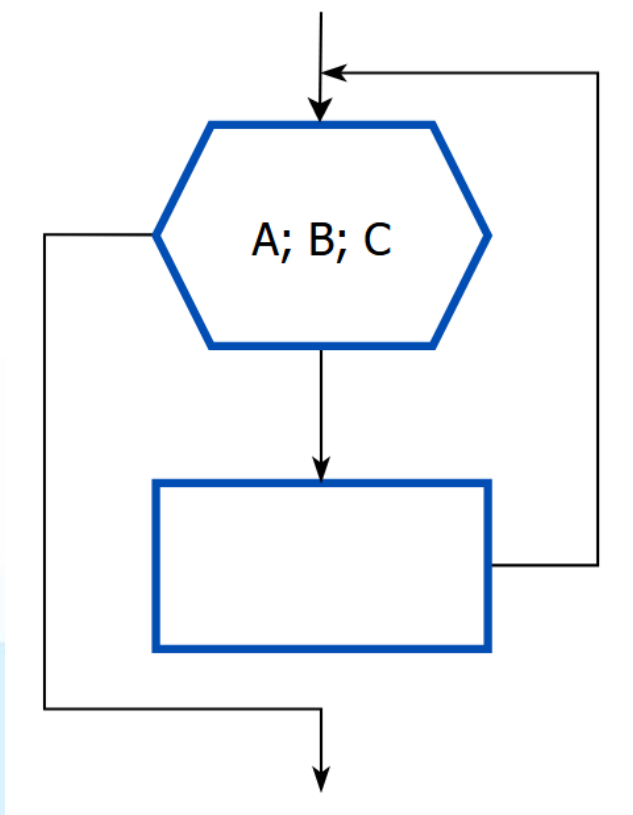
Выполнять

<Действие>

Пока <Условие>

# Цикл со счетчиком

## Блок-схема



## Псевдокод

Для  $x$  от  $A$  до  $B$  [с шагом  $C$ ]

<Действие>

где  $x$  — переменная-счетчик цикла

$A$  — начало интервала

$B$  — конец интервала

$C$  — шаг перемещения по интервалу  
(если не указан, то равен 1)

# Сравнение циклических алгоритмов

	С предусловием	С постусловием	Со счетчиком
Выполняется минимум ? раз			
Выполняется максимум ? раз			
Может ли быть бесконечным?			
Число шагов известно заранее?			

**Таблицу нужно заполнить самостоятельно!**

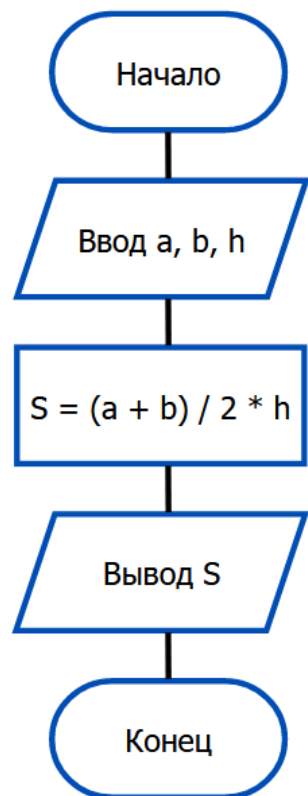
Линейных, разветвляющихся и циклических алгоритмов

# Примеры

# Следование

Найти площадь трапеции.  
Длины оснований и высота  
вводятся с клавиатуры.

## Блок-схема



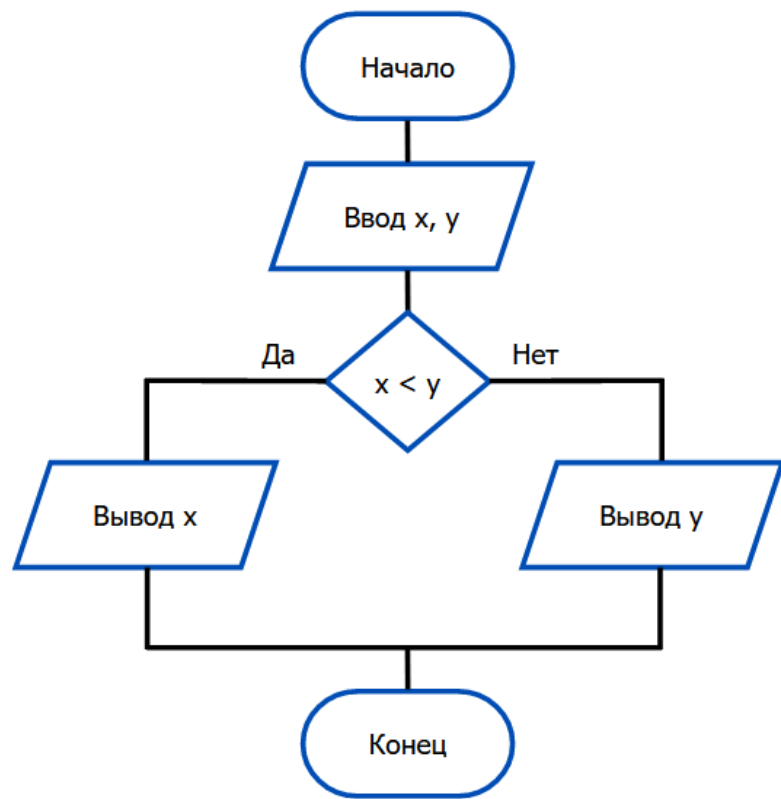
## Псевдокод

**Начало**  
**Ввод а, b, h**  
 $S = (a + b) / 2 * h$   
**Вывод S**  
**Конец**

# Полное ветвление

Вывести минимальное  
из двух введенных  
с клавиатуры чисел.  
Решить задачу  
с использованием  
полного ветвления

## Блок-схема



## Псевдокод

Начало

Ввод  $x, y$

Если  $x < y$

Вывод  $x$

Иначе

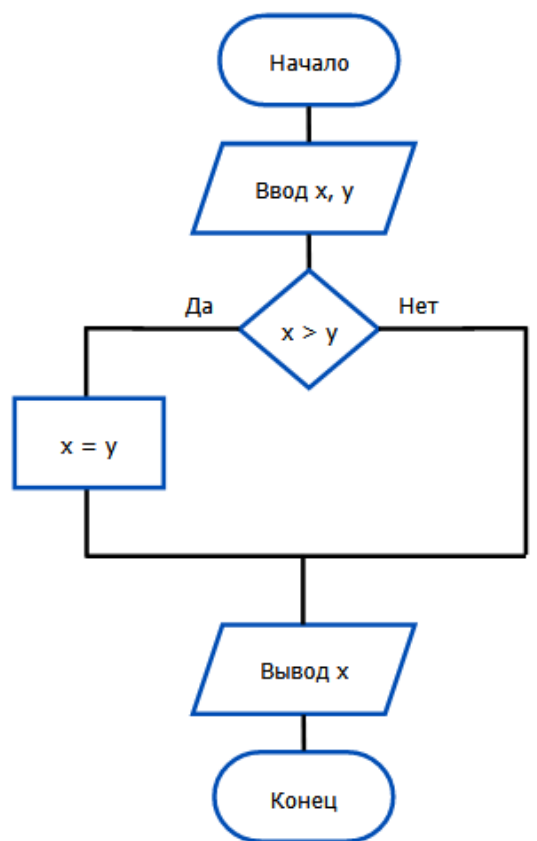
Вывод  $y$

Конец

# Неполное ветвление

Вывести минимальное  
из двух введенных  
с клавиатуры чисел.  
Решить задачу  
с использованием  
неполного ветвления

## Блок-схема



## Псевдокод

Начало

Ввод  $x, y$

Если  $x < y$   
     $x = y$

Вывод  $x$

Конец



# Цикл с предусловием

Написать программу, которая ждет от пользователя ввода загаданного числа. Пока пользователь вводит неправильные числа, программа выводит разность загаданного и введенного чисел. Когда пользователь угадывает число, программа завершается.

## Псевдокод

Начало

Ввод  $x$

Пока  $x \neq 42$

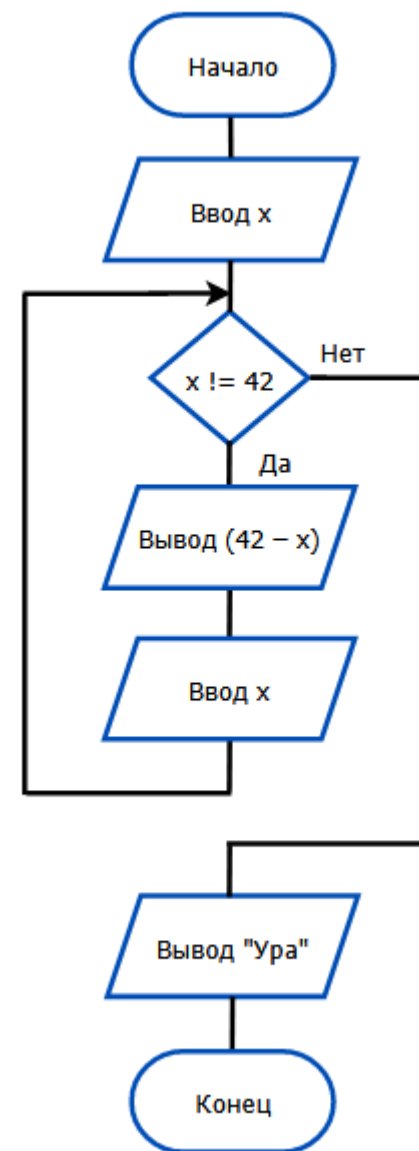
Вывод  $(42 - x)$

Ввод  $x$

Вывод "Ура!"

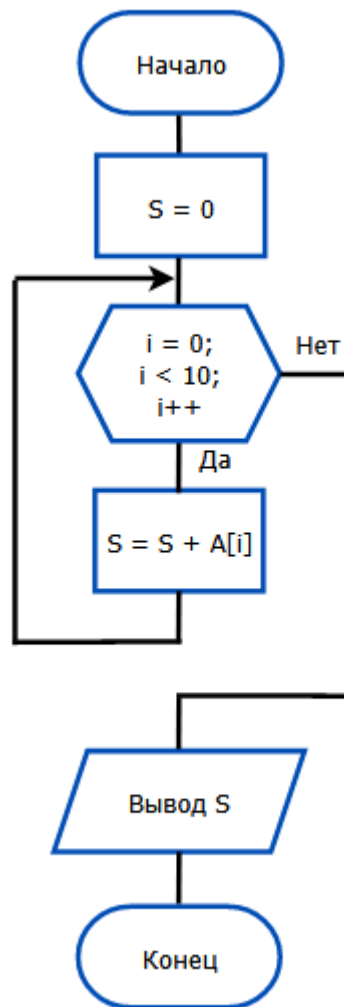
Конец

## Блок-схема



## Цикл со счетчиком

Найти сумму элементов массива, состоящего из 10 элементов. Считать, что массив уже хранится в памяти и вводить его не нужно



### Псевдокод

Начало

$S = 0$

Для  $i$  от 0 до 10

$S = S + A[i]$

Вывод  $S$

Конец

