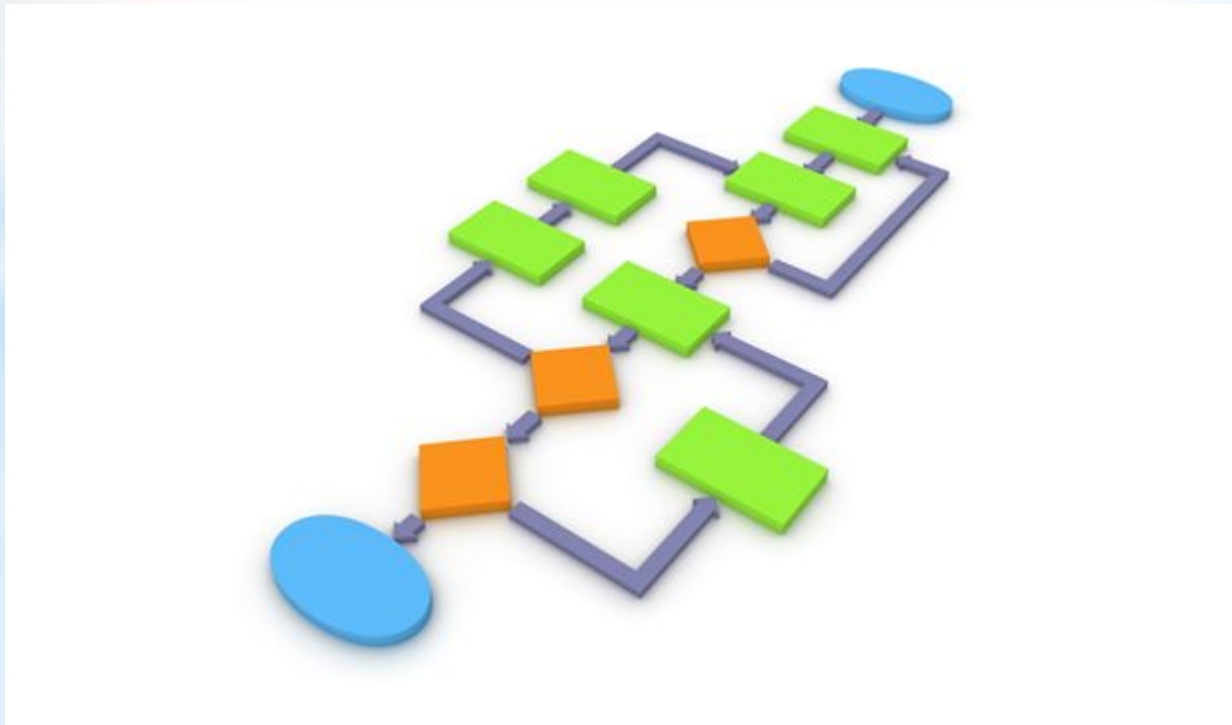
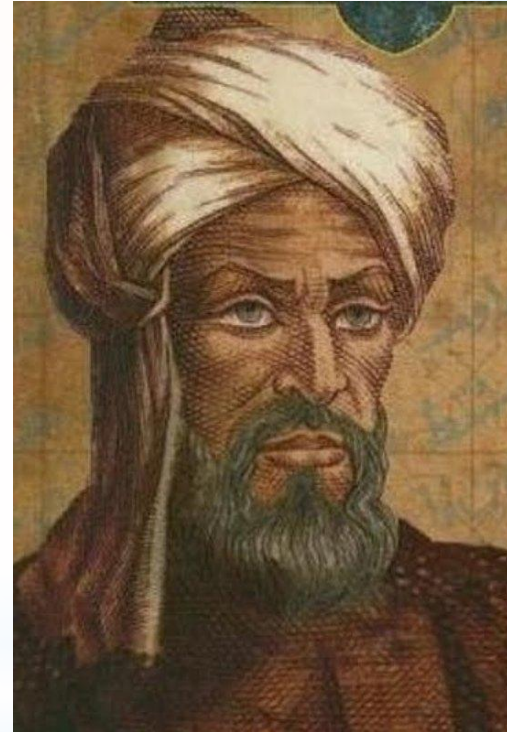


Алгоритмы



Мухаммед Аль- Хорезми (783-850)



Термин «алгоритм» произошел от имени великого математика Мухаммеда аль – Хорезми (по латыни *algoririthmus*).
В IX веке разработал правила выполнения четырех действий арифметики

Алгоритм – описание последовательности действий(план), использование которых приводит к решению поставленной задачи за конечное число шагов

Алгоритмизация - процесс разработки алгоритма (плана действий) для решения задачи

Свойства алгоритмов



Дискретность – предполагает, что любой алгоритм должен состоять из последовательности шагов, следующих друг за другом.



Детерминированность – указывает, что любое действие в алгоритме должно быть строго и не двусмысленно определено и описано для каждого случая.

1



2



3



4



5



6



Массовость – подразумевает, что один и тот же алгоритм может применяться для решения целого класса задач.



Результативность – конечный результат
любого алгоритма.



Конечность – определяет завершение каждого действия в отдельности и алгоритма в целом за конечное число шагов.



Формы представления алгоритма

```
graph TD; A[Формы представления алгоритма] --> B[Словесное]; A --> C[Графическое]; A --> D[Программа]; A --> E[Табличное]; C --> F[Рисунки, программы]; C --> G[Графы, схемы]; C --> H[Блок - схемы];
```

Словесное

Графическое

Программа

Табличное

Рисунки,
программы

Графы, схемы

Блок -
схемы

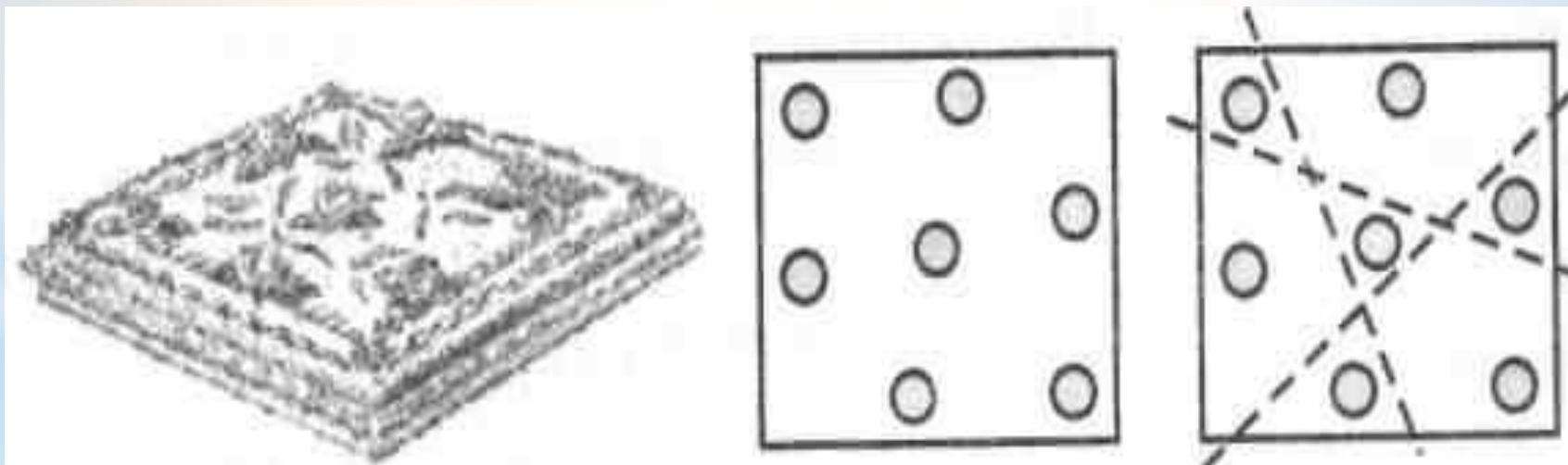
Пример словесной формы представления алгоритма

1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ 2 раза против часовой стрелки.
4. Вынуть ключ

Пример графической формы представления алгоритма в виде рисунка

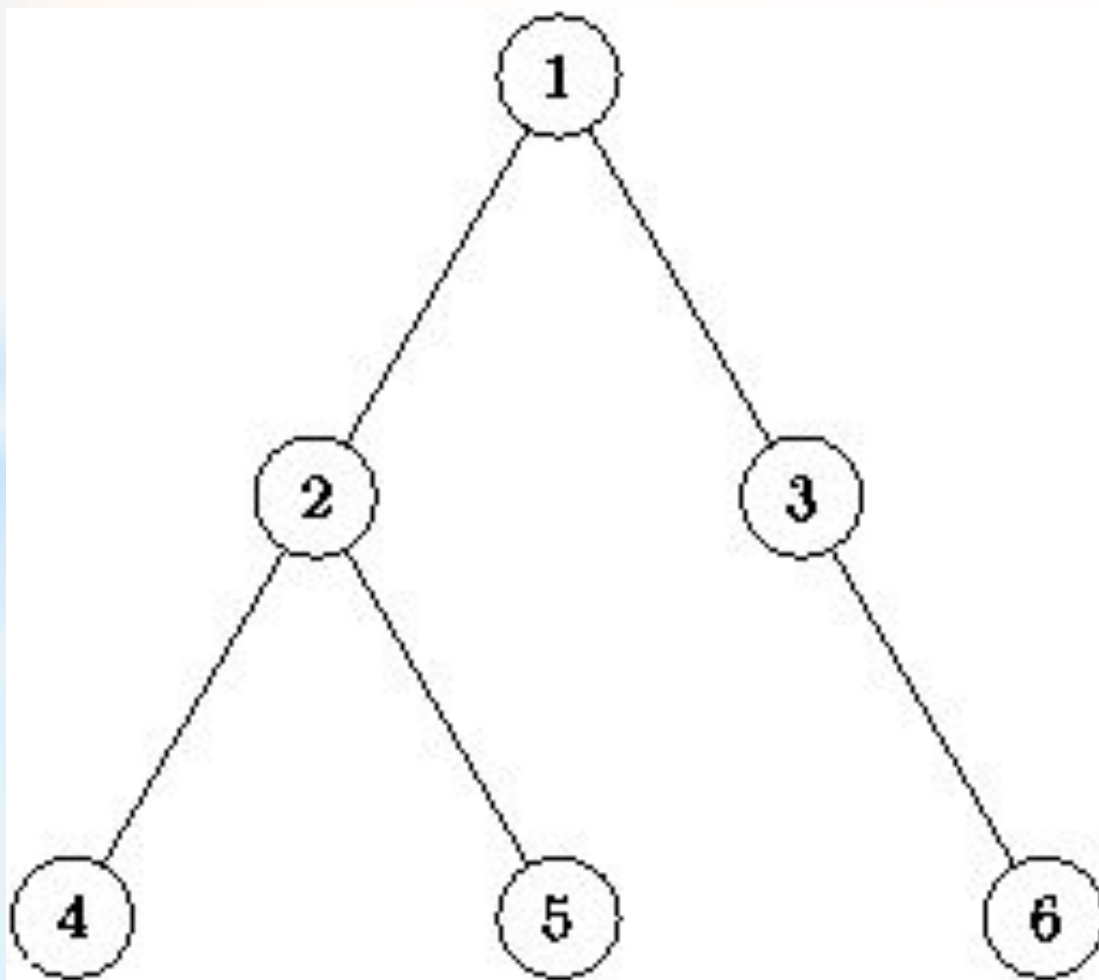


Пример графической формы представления алгоритма в виде схемы



Пример формы графа

Граф – геометрический объект, состоящий из вершин и соединяющих вершины линий-дуг.



Пример формы программы

```
Private Sub Command1_Click()
```

```
N = InputBox («введите N»)
```

```
i=1
```

```
F=1
```

```
Do Until i > N
```

```
F=F*i
```

```
i= i+1
```

```
Loop
```

```
? N; « ! = » ; F
```

```
End Sub
```



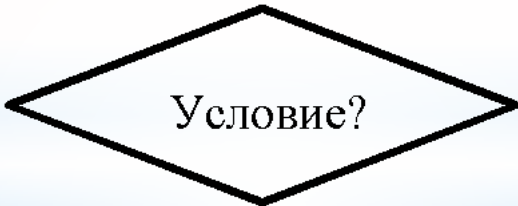


Пример табличной формы

XXII Олимпийские зимние игры

Медальный зачет

Страна				Всего
1  Норвегия	4	3	4	11
2  Канада	4	3	2	9
3  Нидерланды	3	2	3	8
4  Германия	3	1	0	4
5  Соединенные Штаты...	2	1	4	7
6  Швейцария	2	0	0	2
7  Россия	1	3	3	7
21  Украина	0	0	1	1

Стандартные графические объекты блок-схем

Название блока	Вид блока	Название блока
Начало - Конец		Указание на начало и конец алгоритма
Ввод - Вывод		Организация ввода и вывода алгоритма
Решение (условный, логический блок)		Выбор направления выполнения алгоритма в зависимости от выполнения условия
Процесс (блок действий)		Выполнение действия или группы действий
Ранее определенный процесс		Использование вспомогательных алгоритмов

Задача:

Требуется рассчитать необходимое количество рулонов обоев для оклейки комнаты. Заданы параметры: длина(a), ширина(d) и высота(h).

Заданы параметры рулона обоев: длина(l), ширина (d). Считаем, что площадь окон и дверей составляет 15% от площади стен.

Алгоритм «Оклейка обоями»

1. Рассчитать периметр комнаты:

$$p=2*(a + b).$$

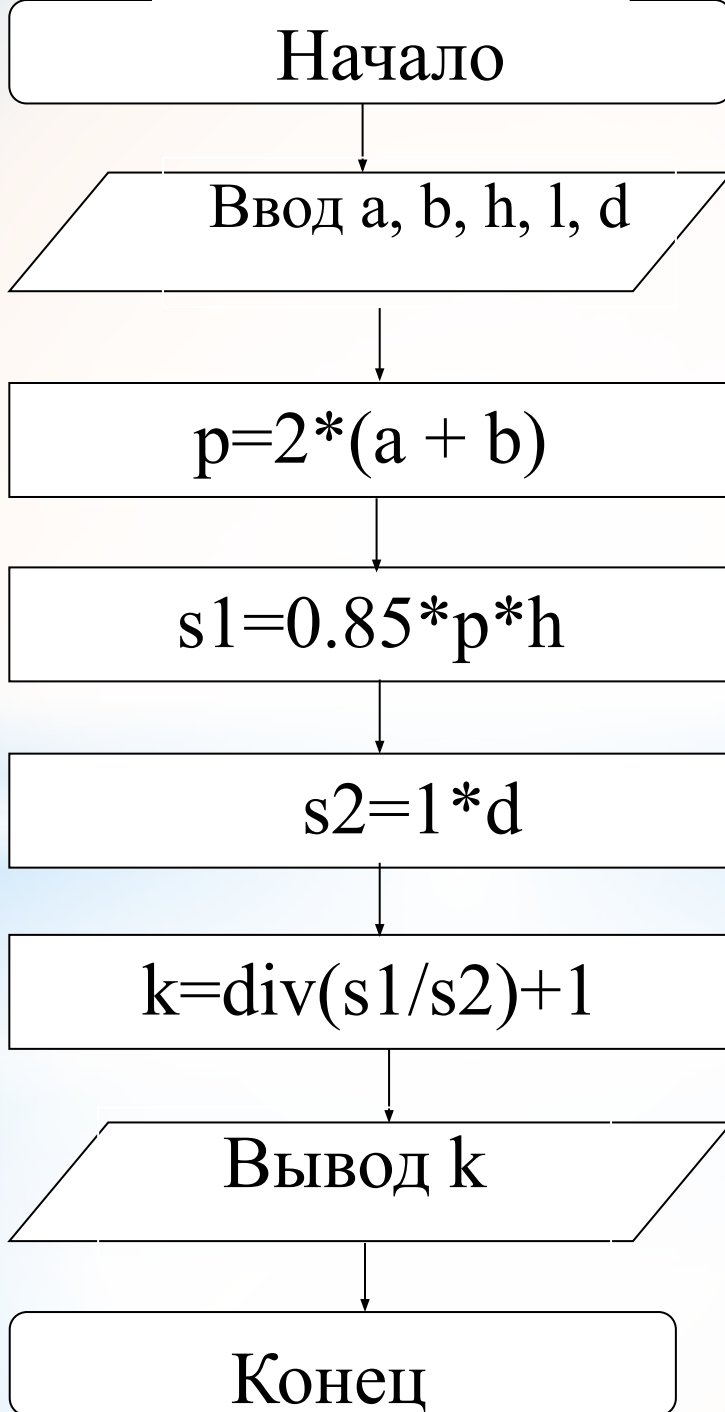
2. Рассчитать площадь стен с учетом дверей и окон: $s1=0.85*p*h$.

3. Рассчитать площадь одного рулона обоев: $s2=1*d$

4. Вычислить количество рулонов:

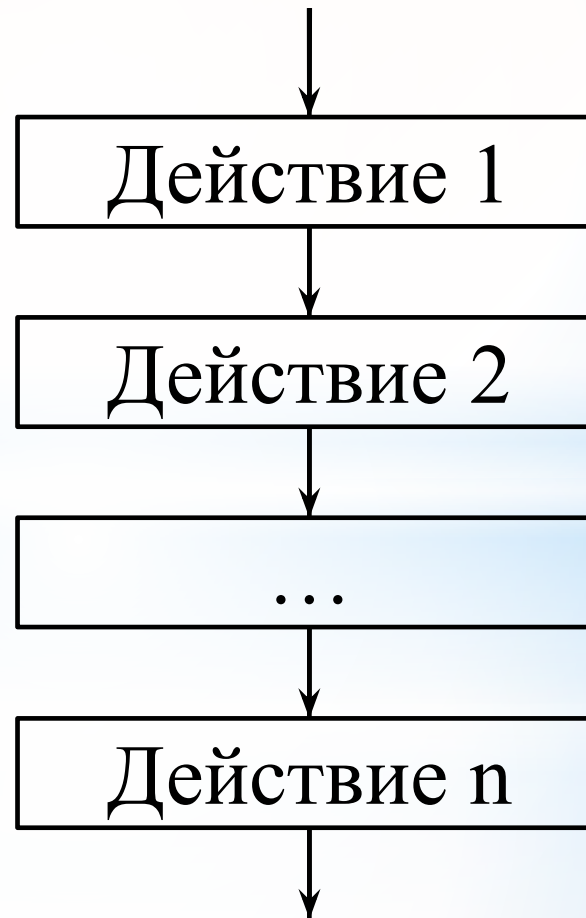
$k=\text{div}(s1/s2)+1$, где div – функция определения целой части числа.

Конец алгоритма.

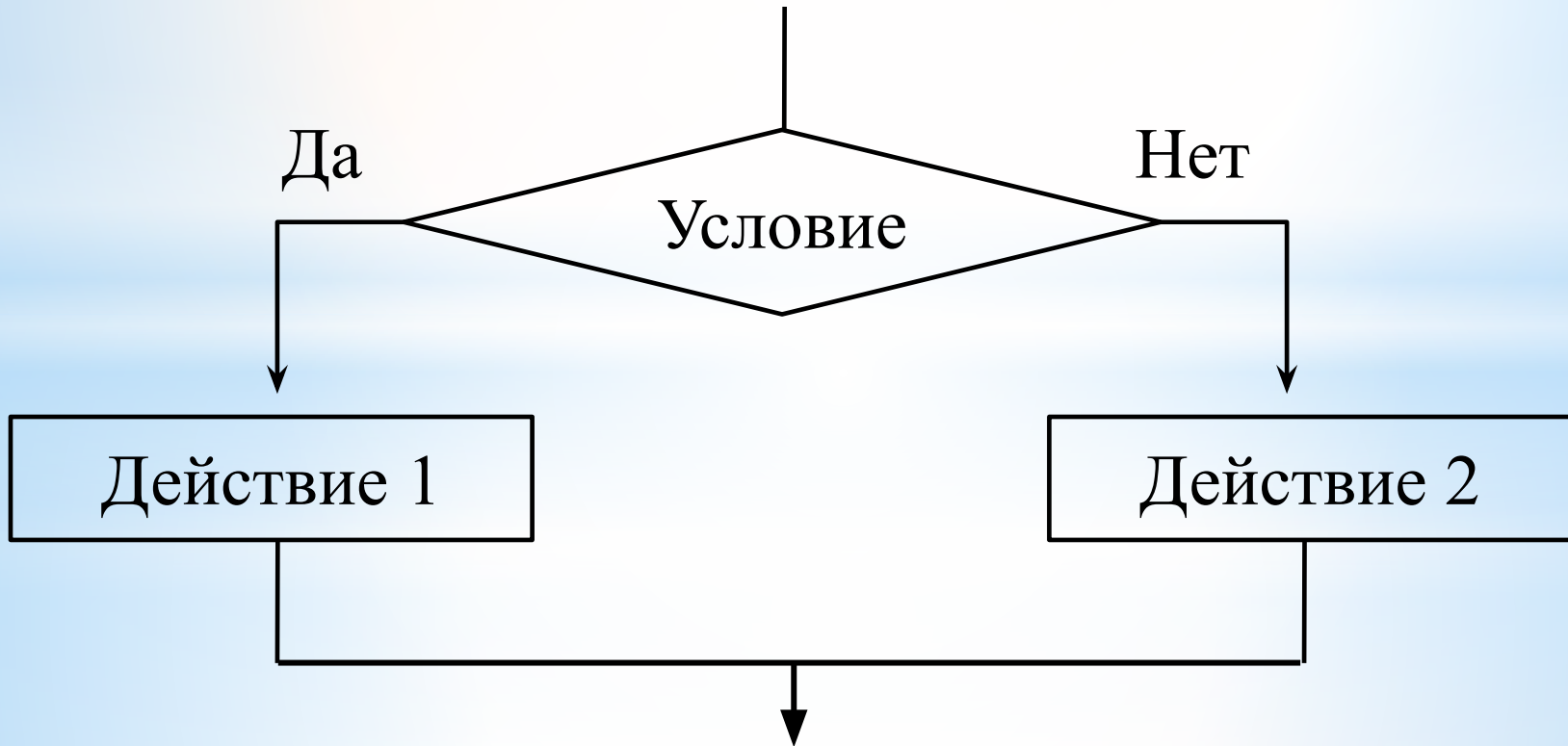


Школьный алгоритмический язык	Пояснения
алг Оклейка обоями	Начало алгоритма
нач вещ $a, b, h, d, p, s1, s2$, цел k	Описание типов переменных
ввод «Введите длину, ширину, высоту комнаты, длину, ширину обоев»	Вывод подсказки на экран
ввод a, b, h, l, d	Ввод информации с клавиатуры
$p := 2 * (a + b)$	Вычисление периметра комнаты
$s1 := 0.85 * p * h$	Вычисление площади стен
$s2 := l * d$	Вычисление площади рулона
$k := \text{div}(s1, s2) + 1$	Вычисление количества рулонов
вывод k	Вывод ответа на экран
кон	Конец алгоритма

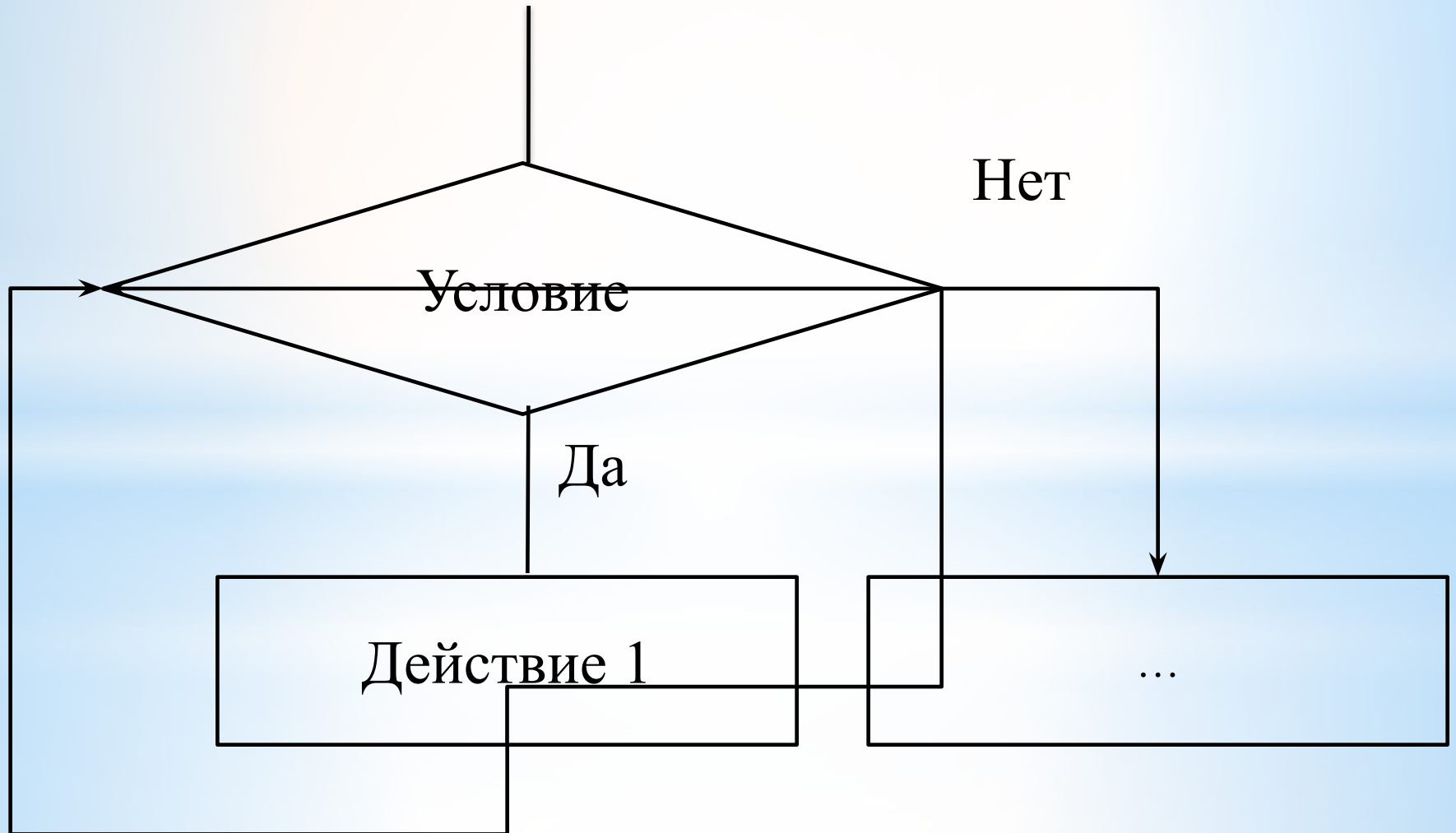
Последовательный алгоритм -
выполняется последовательно, сверху вниз, без
возвратов

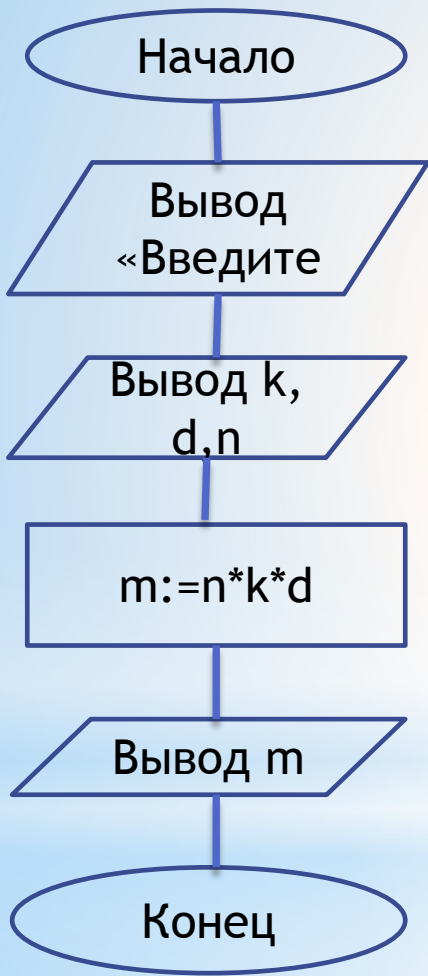


Ветвление – выполняется либо одна, либо другая группа действий в зависимости от истинности(выполнения) или ложности (невыполнения) условия

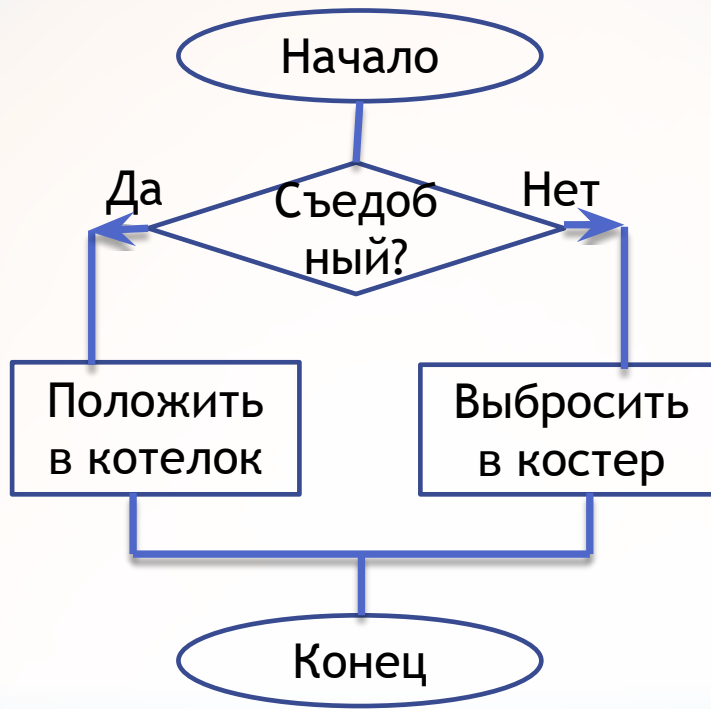


Цикл - действие повторяется до тех пор,
пока выполняется заданное условие

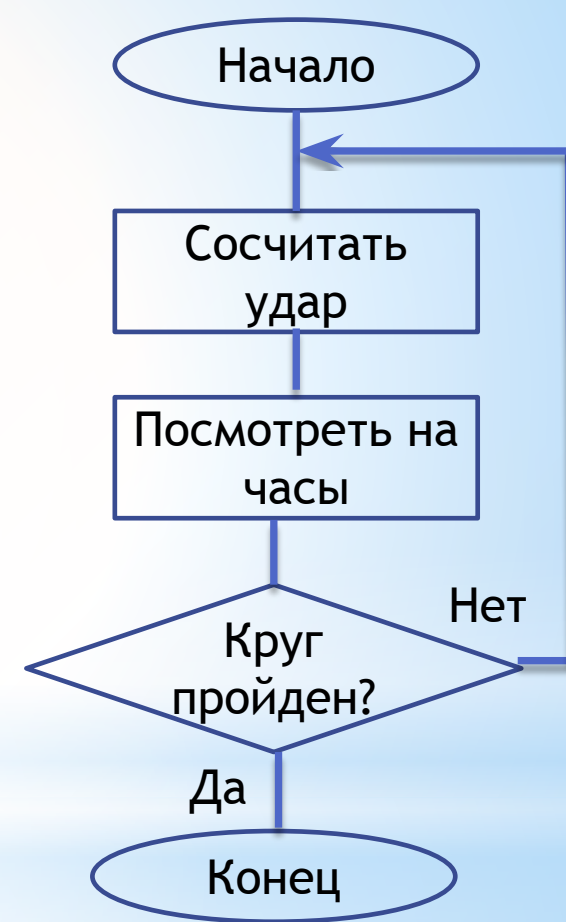




Линейный алгоритм

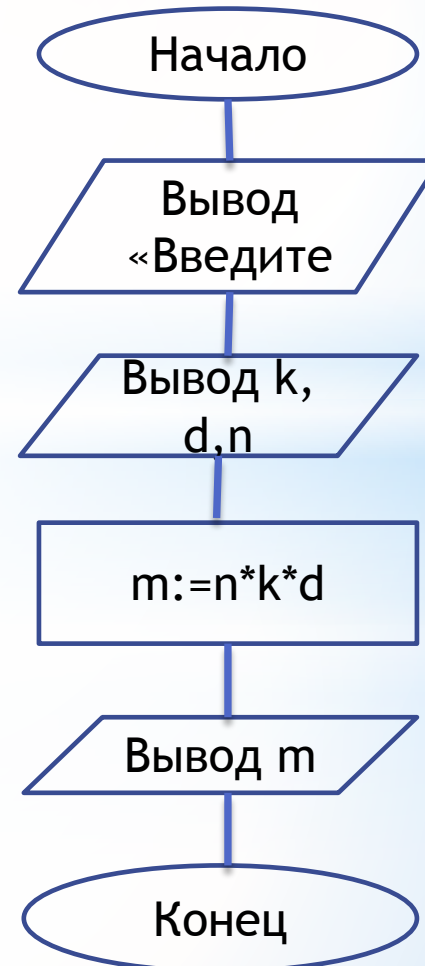


Разветвляющийся алгоритм



Циклический алгоритм

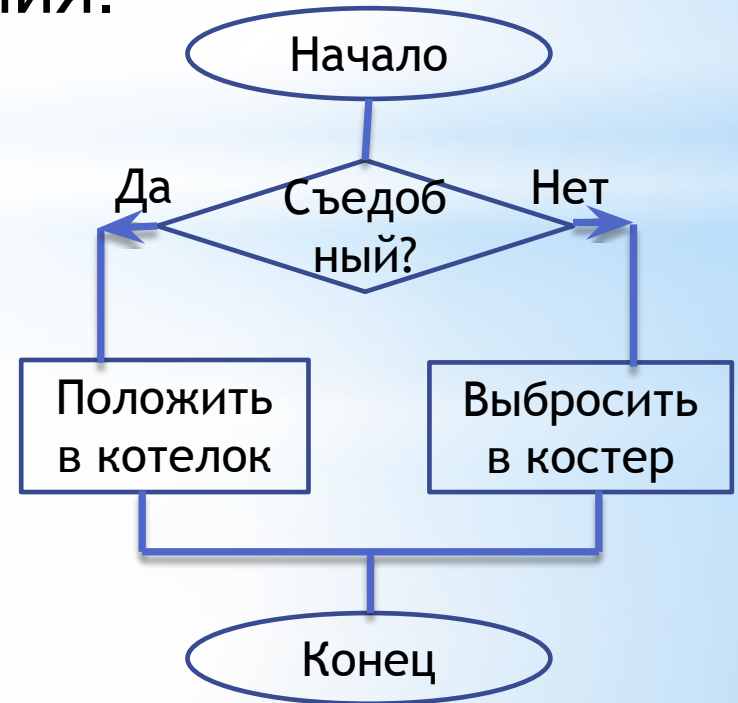
Линейный алгоритм – алгоритм, в котором действия выполняются последовательно одно за другим.



Полная форма: *если <условие>, то <действие 1>, иначе <действие 2>*

Неполная форма: *если <условие>, то <действия>*

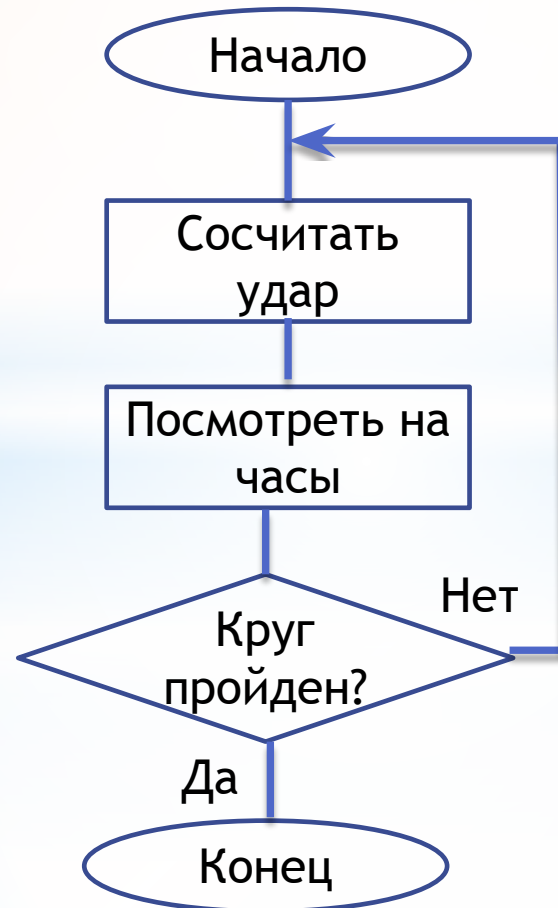
Разветвляющийся алгоритм – алгоритм, содержащий структуру ветвления.



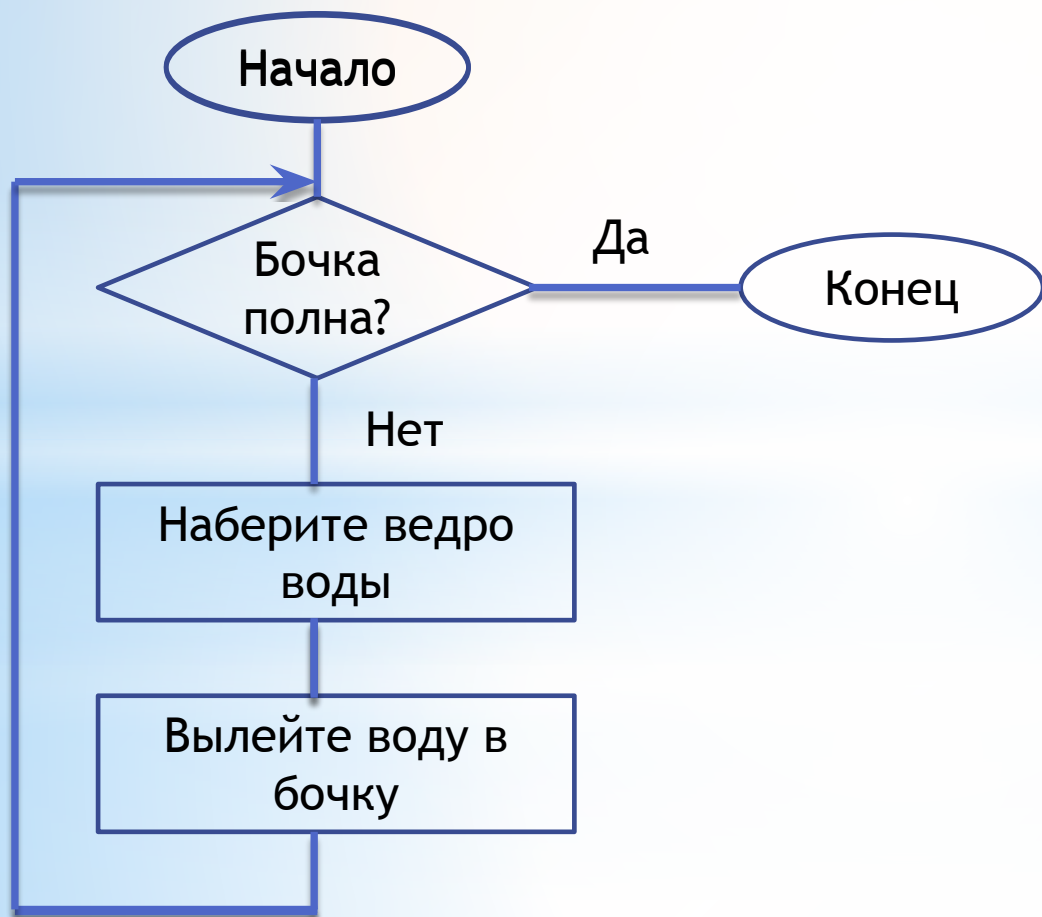
Циклический алгоритм – алгоритм, содержащий типовую конструкцию «цикл»



Цикл с постусловием – это цикл с неизвестным числом повторений, в котором выход из цикла осуществляется при выполнении условия.



Цикл с предусловием – это цикл с неизвестным числом повторений, в котором цикл продолжается, пока выполняется условие.



Вспомогательный алгоритм – это алгоритм, который можно использовать в других алгоритмах, указав его имя и, если имеются, значения параметров.

Первая стадия – алгоритм должен быть представлен в формате, понятной человеку, который его разрабатывает.

Вторая стадия - алгоритм должен быть представлен в форме, понятной тому объекту который будет выполнять описанные в алгоритме действия.