



# КОНСТРУИРОВАНИЕ АЛГОРИТМОВ

# АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

**9 класс**



ИЗДАТЕЛЬСТВО

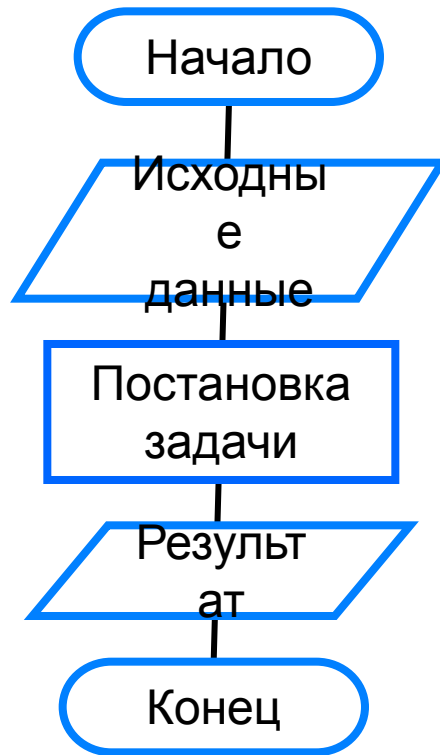
**БИНОМ**

# Ключевые слова

- последовательное построение алгоритма
- вспомогательный алгоритм
- формальные параметры
- фактические параметры
- рекурсивный алгоритм



# Последовательное построение алгоритма



Я совершенный исполнитель: всё знаю и всё умею!



# Последовательное построение алгоритма

Не могу решить поставленную задачу!?



Упрощение команд постановки задачи

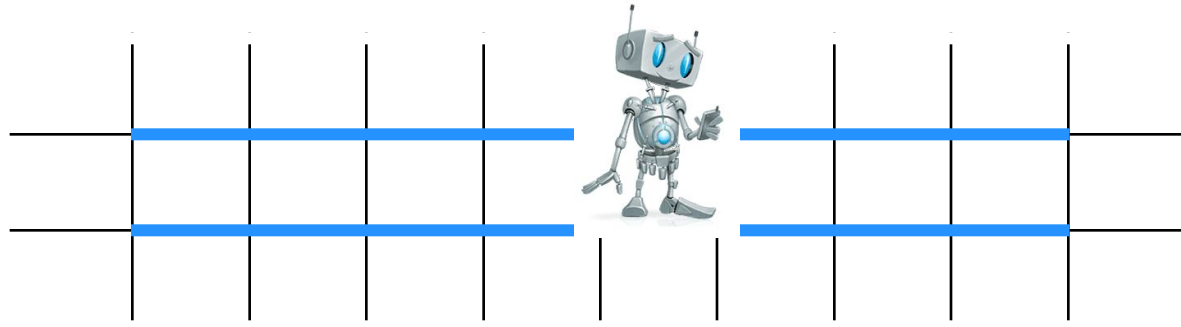
Задача разбивается на более простые части

Решение каждой части задачи формулируется в отдельной команде (предписании)

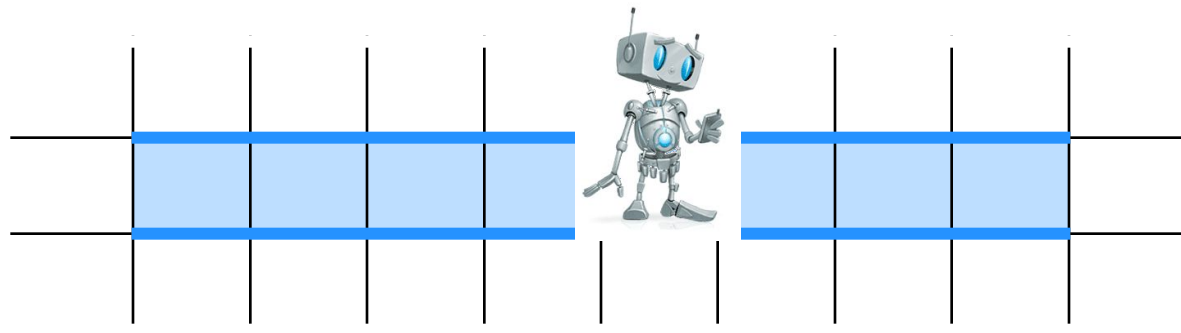
Предписания, выходящие за пределы возможностей исполнителя, представляют в виде более простых команд

# Разработка алгоритма методом последовательного уточнения для исполнителя Робот

Робот находится в некоторой клетке горизонтального коридора. Ни одна из клеток коридора не закрашена.



Робот должен закрасить все клетки этого коридора и вернуться в исходное положение.



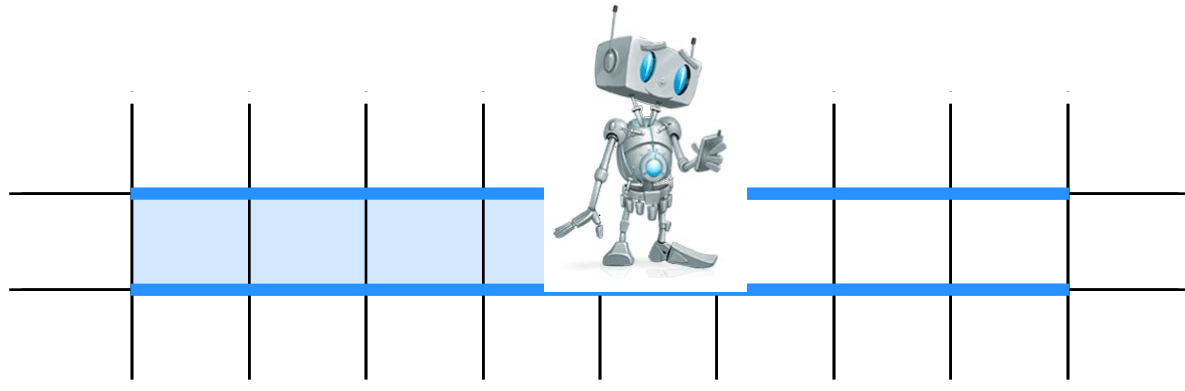
# Укрупнённый план действий Робота



# Детализация плана действий Робота

1. Закраска всех клеток коридора, находящихся левее Робота:  
влево  
**нц пока** сверху стена и снизу стена  
закрасить; влево  
**кц**

Положение Робота после выполнения этого алгоритма:



# Детализация плана действий Робота

2. Возвращение Робота в коридор в исходную точку:

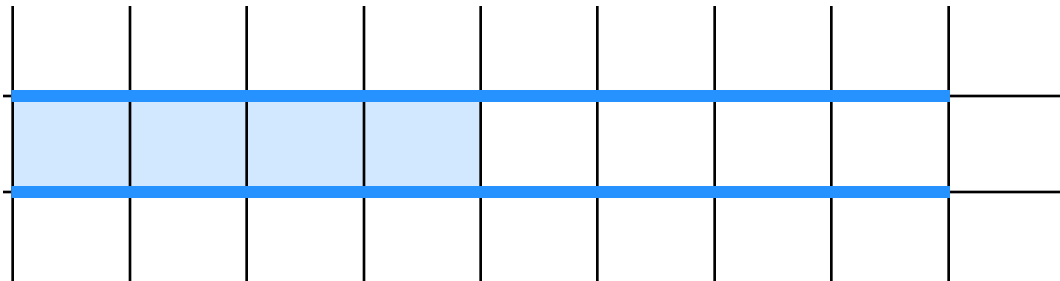
вправо

**нц пока** клетка закрашена

вправо

**кц**

Положение Робота после выполнения этого алгоритма:





# Детализация плана действий Робота

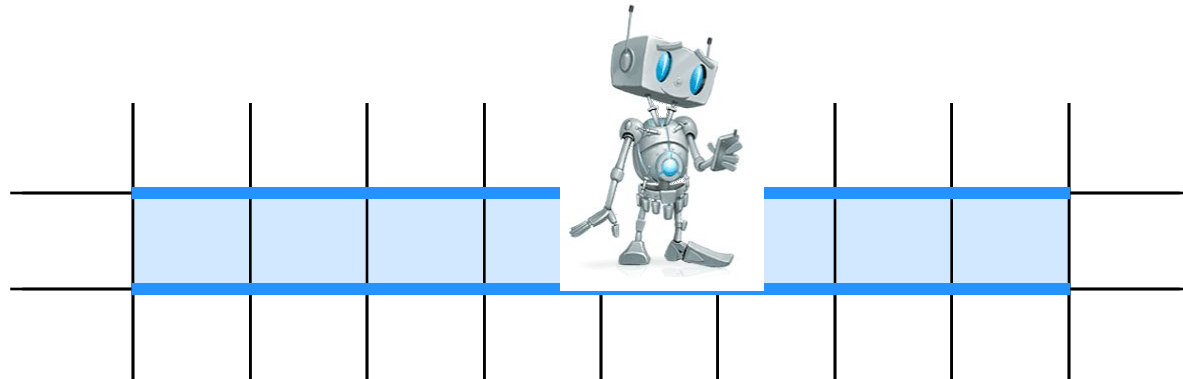
3. Закраска всех клеток коридора, находящихся правее Робота:

вправо

**нц** пока сверху стена и снизу стена  
закрасить; вправо

**кц**

Положение Робота после выполнения этого алгоритма:



# Детализация плана действий Робота

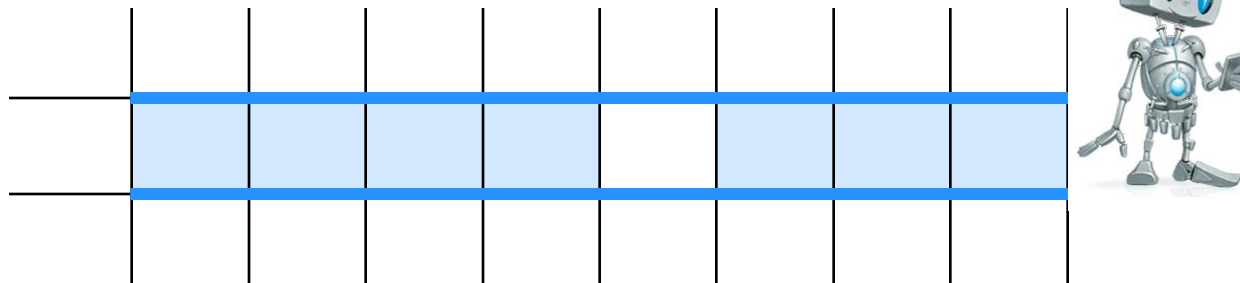
4. Возвращение Робота в коридор в исходную точку:

влево

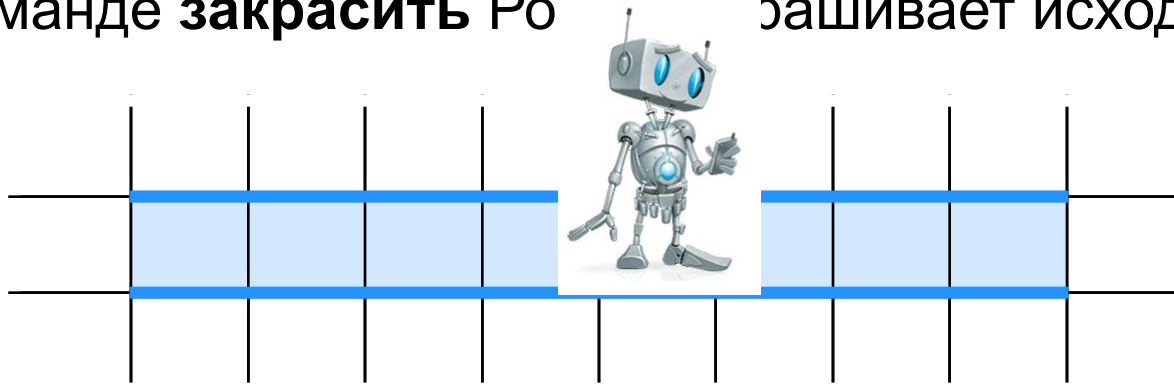
**нц пока** клетка закрашена

влево

**кц**



5. По команде закрасить Робот зашивает исходную точку.



# Программа для Робота

**алг**

**нач**

влево

**нц пока** сверху стена **и** снизу стена  
закрасить; влево

**кц**

вправо

**нц пока** клетка закрашена  
вправо

**кц**

вправо

**нц пока** сверху стена **и** снизу стена  
закрасить; вправо

**кц**

влево

**нц пока** клетка закрашена  
влево

**кц**

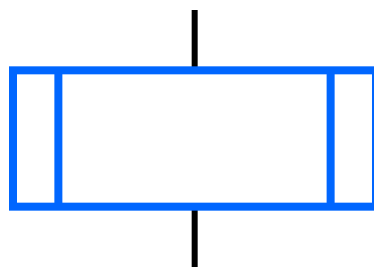
закрасить

**кон**



# Вспомогательный алгоритм

**Вспомогательный алгоритм** - алгоритм, целиком используемый в составе другого алгоритма.



Блок «предопределённый процесс»

Вспомогательный алгоритм делает структуру алгоритма более простой и понятной.

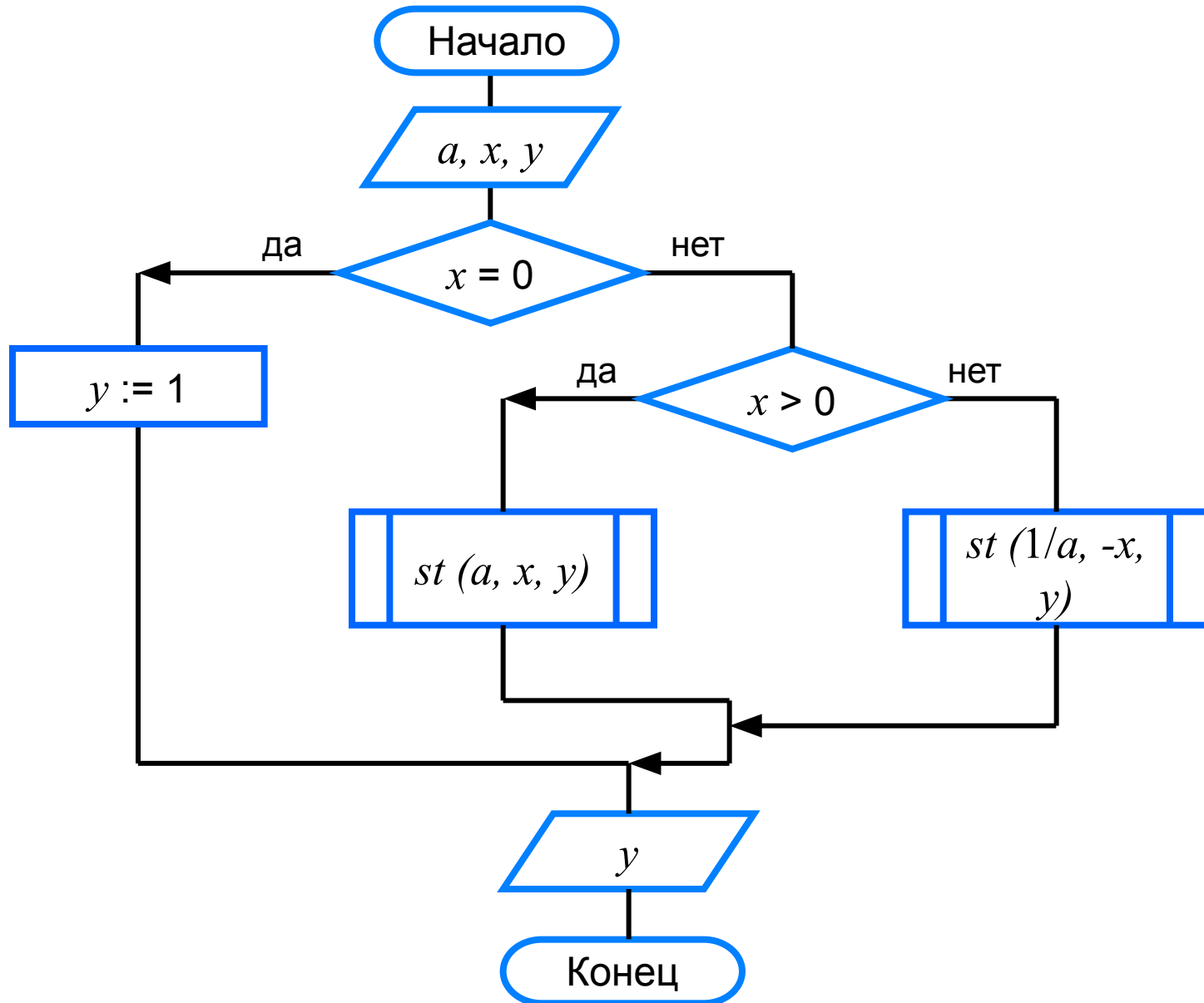
# Алгоритм вычисления степени

$y = a^x$ , где  $x$  - целое число,  $a \neq 0$ .

$$y = \begin{cases} 1, & \text{при } x = 0 \\ a^x, & \text{при } x > 0, \\ \left(\frac{1}{a}\right)^{-x}, & \text{при } x < 0. \end{cases}$$

Обозначим алгоритм возведения числа в степень  $st(a, n, y)$ .  
Это вспомогательный алгоритм.

# Блок-схема решения задачи:



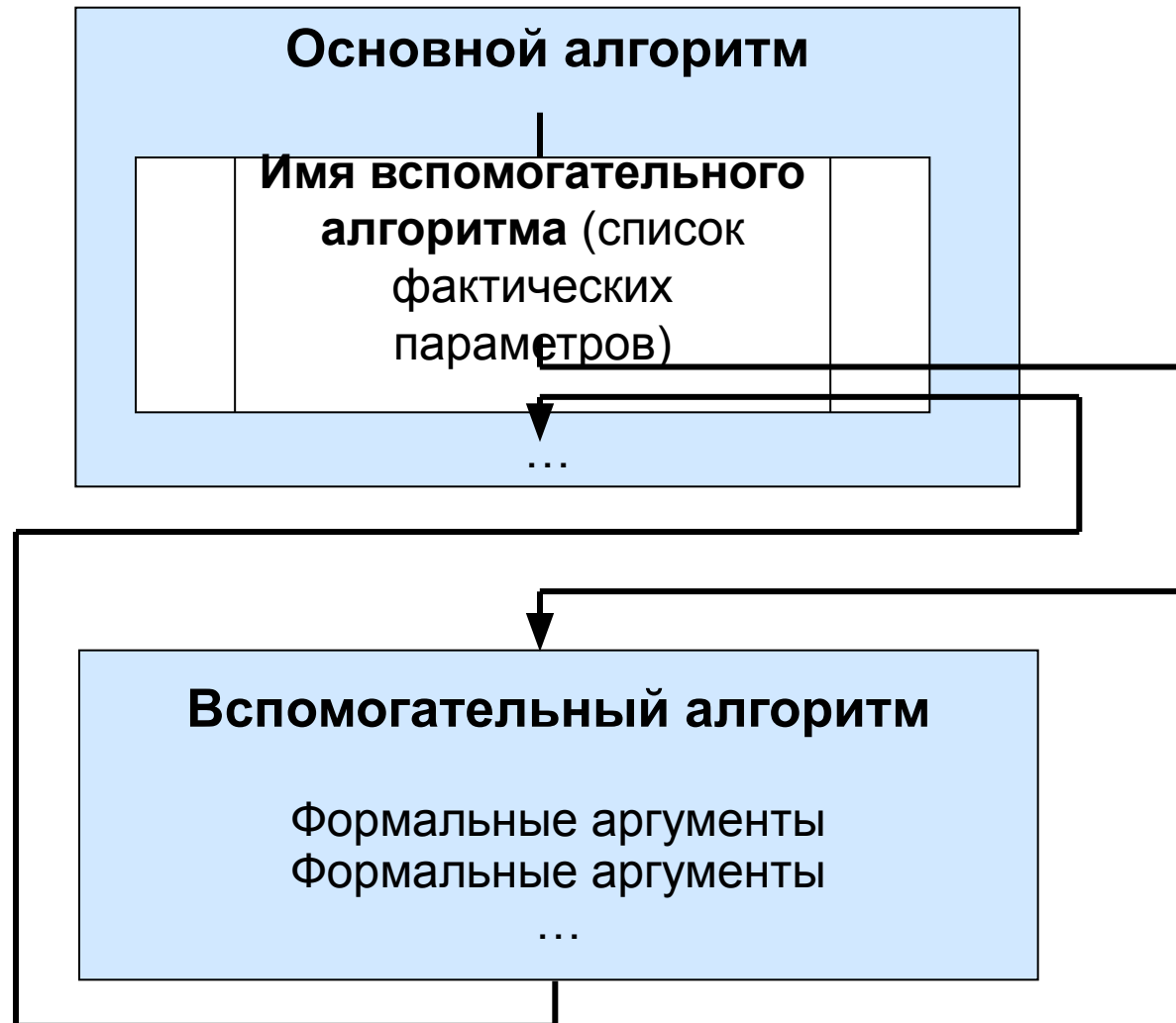
# Формальные и фактические параметры

**Формальные параметры** используются при описании алгоритма.

**Фактические параметры** - те величины, для которых будет исполнен вспомогательный алгоритм.

Типы, количество и порядок следования формальных и фактических параметров должны совпадать.

# Схема вызова вспомогательного алгоритма

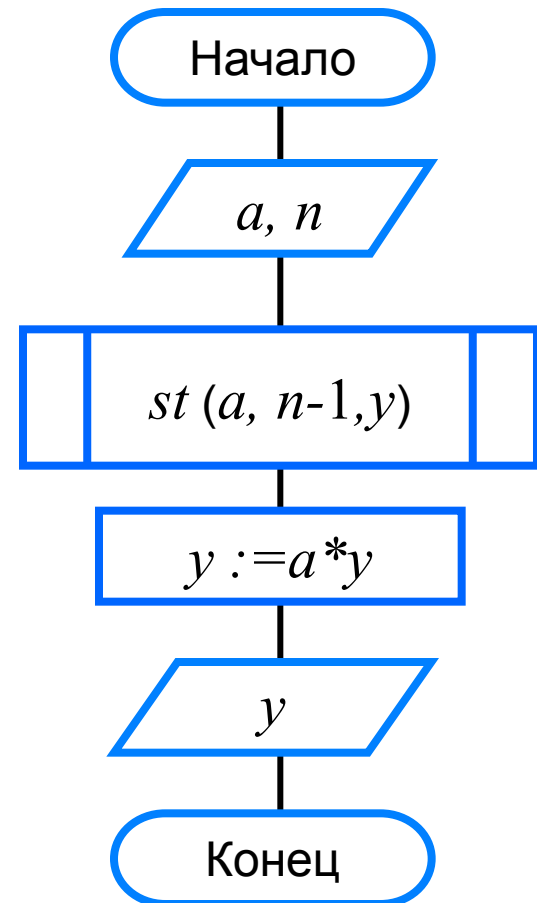




# Рекурсивный алгоритм

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют **рекурсивным**.

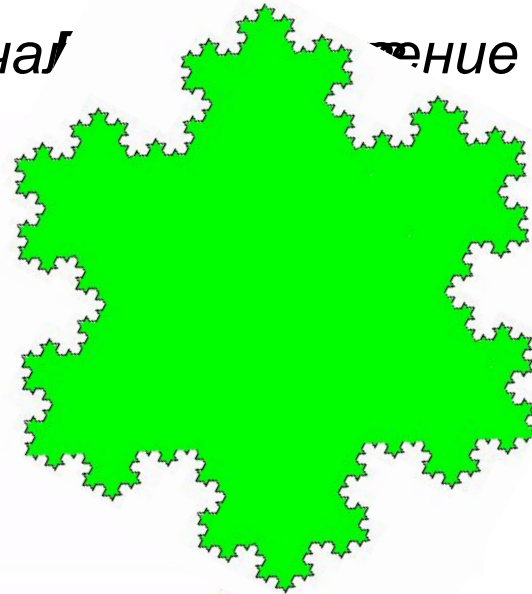
**Пример.** Алгоритм вычисления степени с натуральным показателем  $n$  для любого вещественного числа  $a$ , представленный в виде рекурсивного алгоритма



# Снежинка Коха

**Пример.** Рассмотрим алгоритм построения геометрической фигуры, которая называется снежинкой Коха. Шаг процедуры построения состоит в замене средней трети каждого из имеющихся отрезков двумя новыми той же длины.

*Начало*  *Конец*



С каждым шагом фигура становится всё причудливее. Граница снежинки Коха - положение кривой после выполнения бесконечного числа шагов.

# Самое главное

## ***Метод последовательного построения алгоритма:***

- исходная задача разбивается на несколько частей, каждая из которых проще всей задачи, и решение каждой части формулируется в отдельной команде;
- если получаются команды, выходящие за пределы возможностей исполнителя, то они представляются в виде совокупности ещё более простых предписаний;
- процесс продолжается до тех пор, пока все предписания не будут понятны исполнителю.

***Вспомогательный алгоритм*** - алгоритм, целиком используемый в составе другого алгоритма.

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют ***рекурсивным***.





# Опорный конспект

**Метод последовательного построения алгоритма** - один из основных методов конструирования алгоритмов.



**Вспомогательный алгоритм** - алгоритм, целиком используемый в составе другого алгоритма.