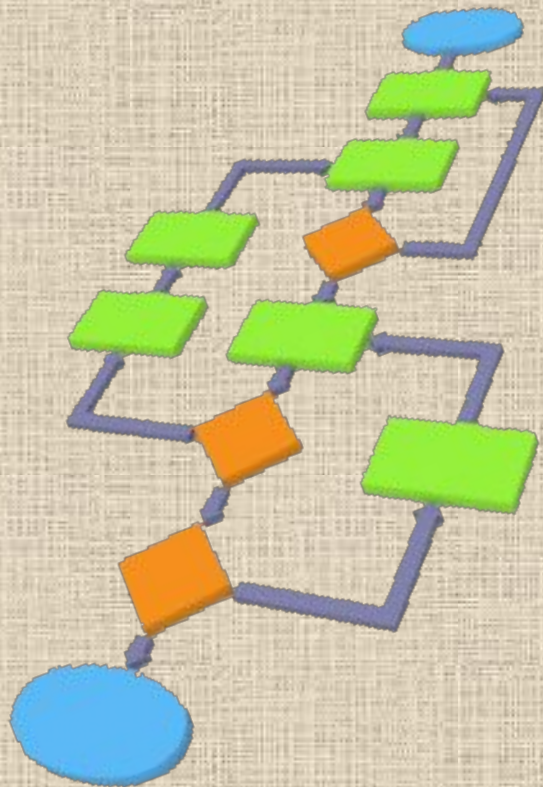


# Основы алгоритмизации и программирования



Исакова Виктория  
Викторовна

# Понятие алгоритма

Алгоритмизация – это описание очередности выполнения различных операций, необходимых для решения какой-либо задачи, в форме алгоритма.

Алгоритмизация – это разработка алгоритма.

Алгоритм – это набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

Термин “алгоритм” происходит от имени узбекского ученого IX века аль-Хорезмí, который изложил правила арифметических действий над числами в десятичной системе счисления. Эти правила и называли алгоритмами.

Т.о., правила сложения, вычитания, деления, умножения чисел, правила преобразования алгебраических выражений, правила построения геометрических фигур, правила правописания слов и предложений, различные правила и инструкции, представляющие собой подробные указания, годные в однотипных ситуациях – всё это алгоритмы.

# Пример (линейного) алгоритма



Начал  
о

Отрезать  
ломтик  
хлеба

Намазать  
хлеб  
маслом

Отрезать  
ломтик  
сыра

Положит  
ь сыр на  
хлеб

Коне  
ц

Алгоритмизация вычислительного процесса включает следующие действия:

1. Последовательная декомпозиция задачи, выделение автономных этапов вычислительного процесса и разделение каждого этапа на отдельные шаги.
2. Формализация задачи, перевод задачи на язык математических формул, уравнений, отношений.

3. Построение алгоритма, определение общего порядка выполнения этапов и/или шагов.

4. Проверка правильности алгоритма.

Далее следует программирование на определенном языке в определенной системе программирования.

Затем перед использованием программы выполняется отладка и тестирование.

# Свойства алгоритма

1. Дискретность (прерывность, раздельность) – алгоритм должен состоять из последовательности законченных действий – шагов. Переход к следующему шагу возможен лишь после завершения предыдущего.
2. Определенность – каждое правило алгоритма должно быть четким, однозначным.

3. Массовость – возможность решения по одному алгоритму множества однотипных задач.

4. Результативность – алгоритм должен обеспечивать возможность получения результата после конечного числа шагов.





# Способы описания алгоритмов

1. **Словесный** – это последовательное описание основных этапов обработки данных в произвольном изложении на естественном языке.

Пример словесного способа записи алгоритма нахождения НОД двух чисел:

- Если числа равны, то взять любое из них в качестве ответа, в противном случае – продолжить выполнение алгоритма;
- Определить большее из чисел;
- Заменить большее число разностью большего и меньшего чисел;
- Повторить алгоритм сначала.

2. Графический – это метод блок-схем. При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Для начертания схем алгоритмов используется набор символов, определяемых государственным стандартом:

 	<p><i>Начало и конец алгоритма (для функций «Вход», «Выход»)</i></p>
	<p><i>Блок обработки. Внутри блока записываются формулы, обозначения и функции</i></p>
	<p><i>Блок условия. Внутри блока записываются условия выбора направления действия алгоритма</i></p>
	<p><i>Блок предопределенного процесса (функция/подпрограмма)</i></p>
	<p><i>Блок ввода информации</i></p>
	<p><i>Блок цикла с известным количеством повторений</i></p>
	<p><i>Блок вывода информации на печатающее устройство</i></p>
	<p><i>Соединительный блок</i></p>

3. Псевдокод – представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Псевдокод занимает промежуточное место между естественным и формальным языками.

Примером псевдокода может являться алгоритмический язык:

**алг** Сумма квадратов (**арг цел**  $n$ , **рез цел**  $S$ )

**дано** |  $n > 0$

**надо** |  $S = 1*1 + 2*2 + 3*3 + \dots + n*n$

**нач цел**  $i$

**ВВОД**  $n$ ;

$S := 0$

**нц для**  $i$  **от** 1 **до**  $n$

$S := S + i*i$

**кц**

**ВЫВОД** "S = ",  $S$

**кон**

4. Программный способ представления алгоритмов – осуществляется с помощью языков программирования.

```
1  #include <iostream>
2  using namespace std;
3
4  int NOD(int x, int y)
5  {
6      if (x != 0)
7          return NOD(y%x,x);
8      else
9          return y;
10 }
11
12 int NOK(int x, int y)
13 {
14     return (x/NOD(x,y))*y;
15 }
16
17 int main()
18 {
19     setlocale(LC_ALL, "rus");
20     int x,y;
21     cin>>x>>y;
22     cout<<"НОД этих чисел = "<<NOD(x,y)<<endl;
23     cout<<"НОК этих чисел = "<<NOK(x,y)<<endl;
24     return 0;
25 }
```

# Данные и величины

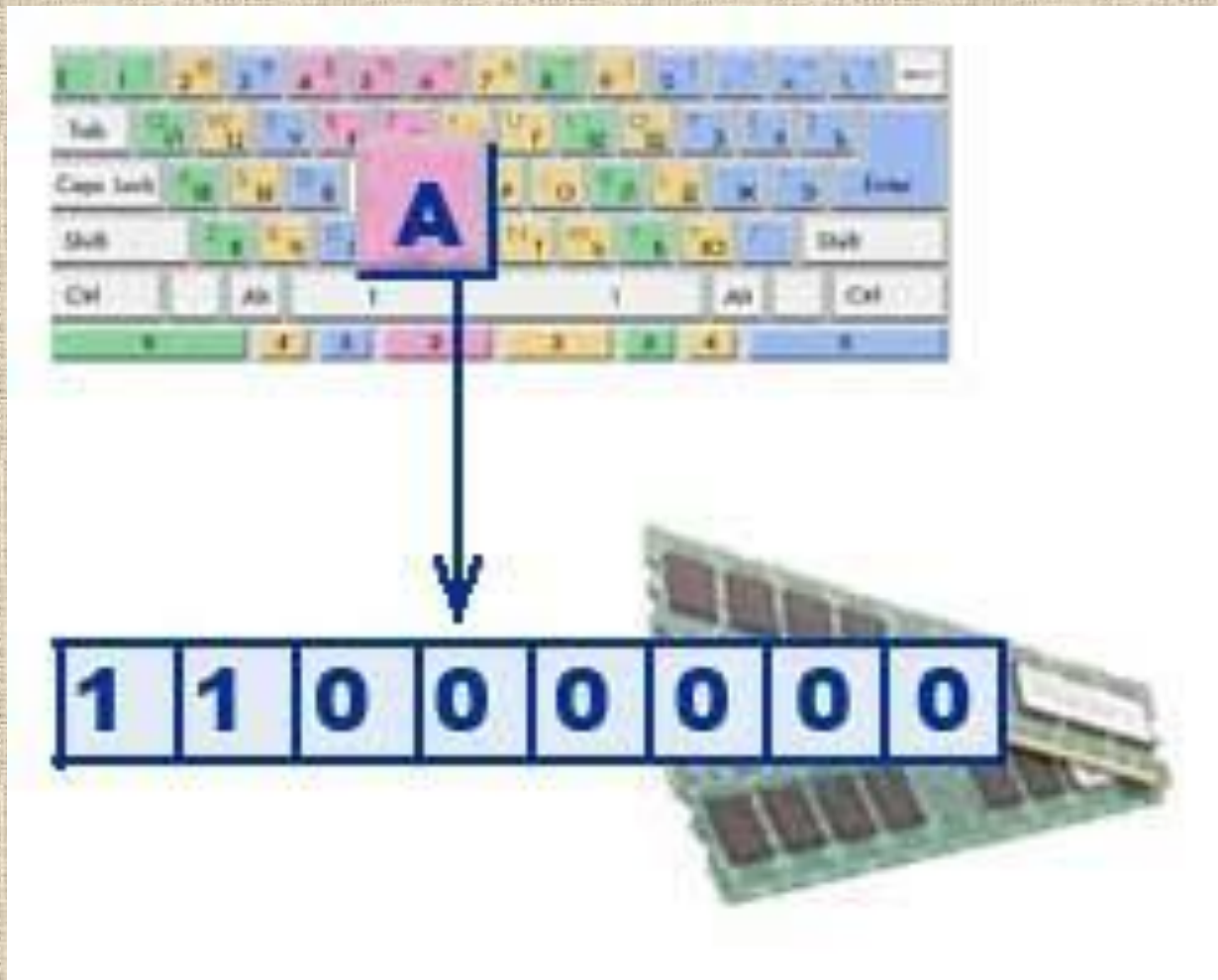
Величины делятся на константы и переменные.

У каждой переменной (константы) есть три основных свойства:

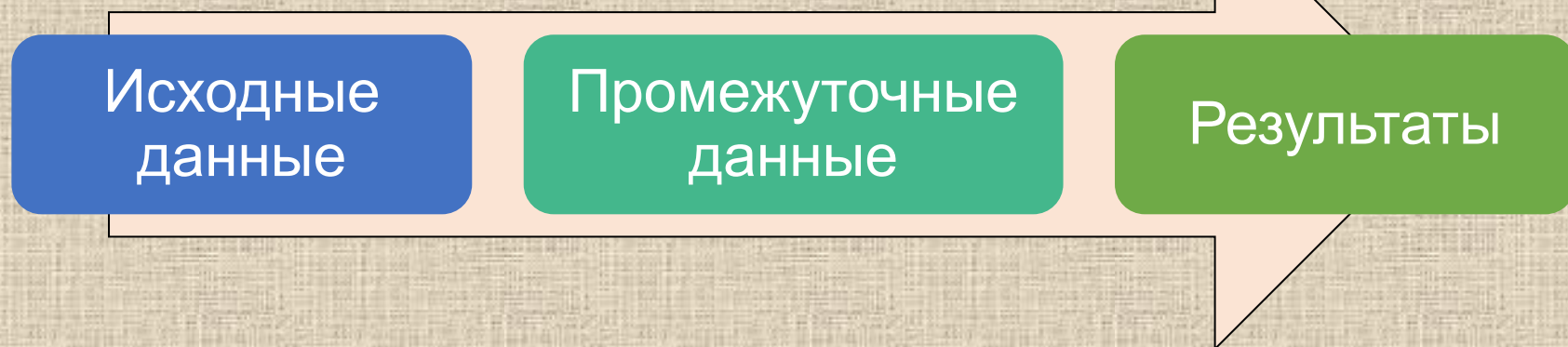
- **Имя** (например, константа может иметь имя  $X$  – икс);
- **Значение** (например, константа  $X$  может иметь значение 1);
- **Тип** (например, константа  $X$ , равная единице, имеет целый тип данных).

$$X = 1$$

Каждая величина занимает определенное место в памяти – ячейку, а значение этой величины определяется двоичным кодом в этой ячейке.



Различные величины, с которыми работает компьютер, принято называть данными. По отношению к программе данные делятся на исходные данные, промежуточные данные и результаты.



Например, при решении квадратного уравнения  $ax^2 + bx + c = 0$

**исходными данными** являются коэффициенты  $a$ ,  $b$ ,  $c$

**промежуточными данными** является дискриминант  $D$

**результатами** являются корни  $x_1$  и  $x_2$ .

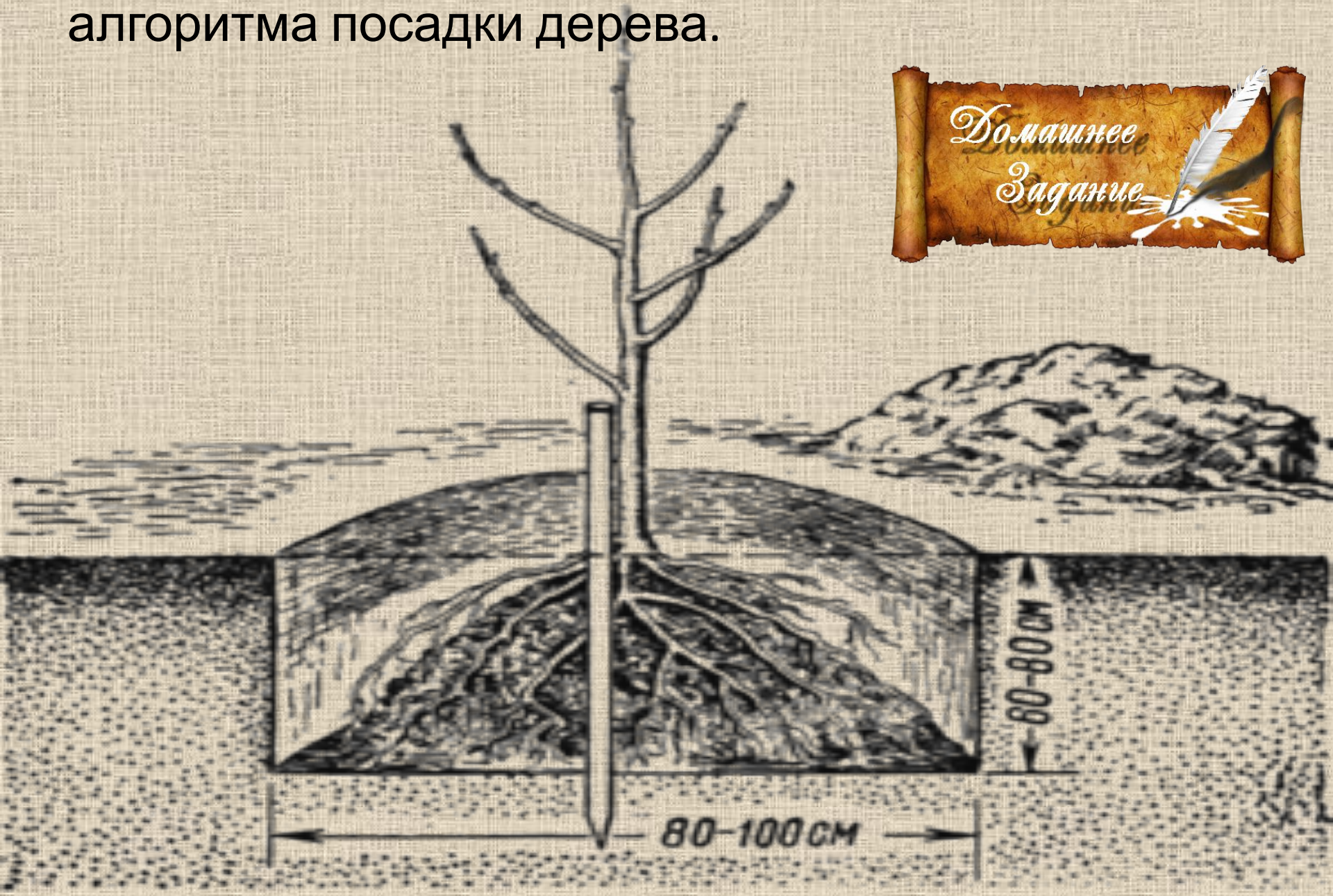


# Линейный алгоритм

Линейный алгоритм – это тип алгоритма, в котором последовательность действий не меняется в процессе его выполнения.

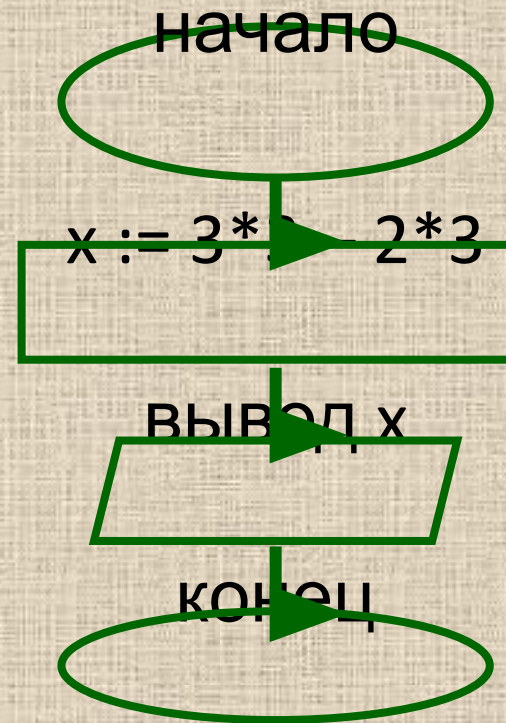
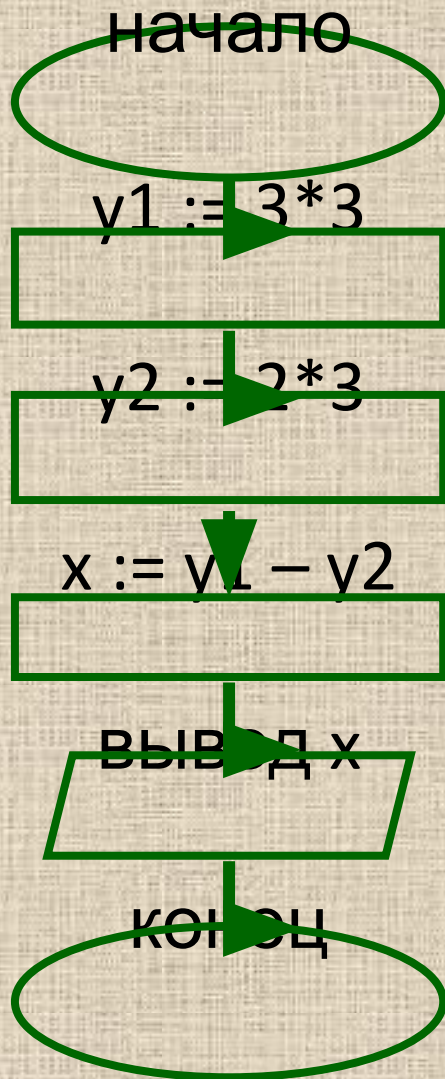


Задача 1. Составить блок-схему линейного алгоритма посадки дерева.



В алгоритмах линейной структуры инструкции выполняются последовательно, одна за другой, т. е. линейно. Алгоритмы и программы с линейной структурой являются простейшими и используются, как правило, для реализации простых вычислений по формулам.

Пример. Составить блок-схему алгоритма вычисления значения переменной  $x$  по формуле:  $x = 3^2 - 2 * 3$



# Правила записи математических выражений в блок-схемах

Название	Математическая запись	Алгоритмическая запись
Абсолютная величина (модуль)	$ x $	abs (x)
Степень числа		$x^*x$ sqr (x)
Корень квадратный		sqrt (x)
Синус	$\sin x$	sin (x)
Умножение		$x * y$
Деление	$x : y$	$x / y$
Частное от деления целого x на целое y		div (x, y) $x \text{ div } y$

# Примеры записи арифметических выражений

Математическая запись	Алгоритмическая запись
	$x * y / z$
	$x / (y * z)$ $x / y / z$
	$(-b + \text{sqrt}(b * b - 4 * a * c)) / (2 * a)$

## Типичные ошибки в записи выражений

$$2x+1$$

$$b + \sin x$$

$$((x + y)^*2 / (b+1))$$

Задача 2. Записать по правилам алгоритмического языка следующие выражения:

$$\text{а) } \frac{x+y}{x-\frac{x}{2}} - \frac{x-z}{xy}$$

$$\text{б) } (1+z) \frac{x+\frac{y}{z}}{a \frac{1}{1+x^2}}$$



Задача 3. Записать в обычной математической форме арифметические выражения:

$$\text{а) } d * c / 2 / (R + a * a)$$

$$\text{б) } \text{abs}(x) / 3 / z + x * x$$



Златопольский Д.М. Сборник задач по  
программированию.

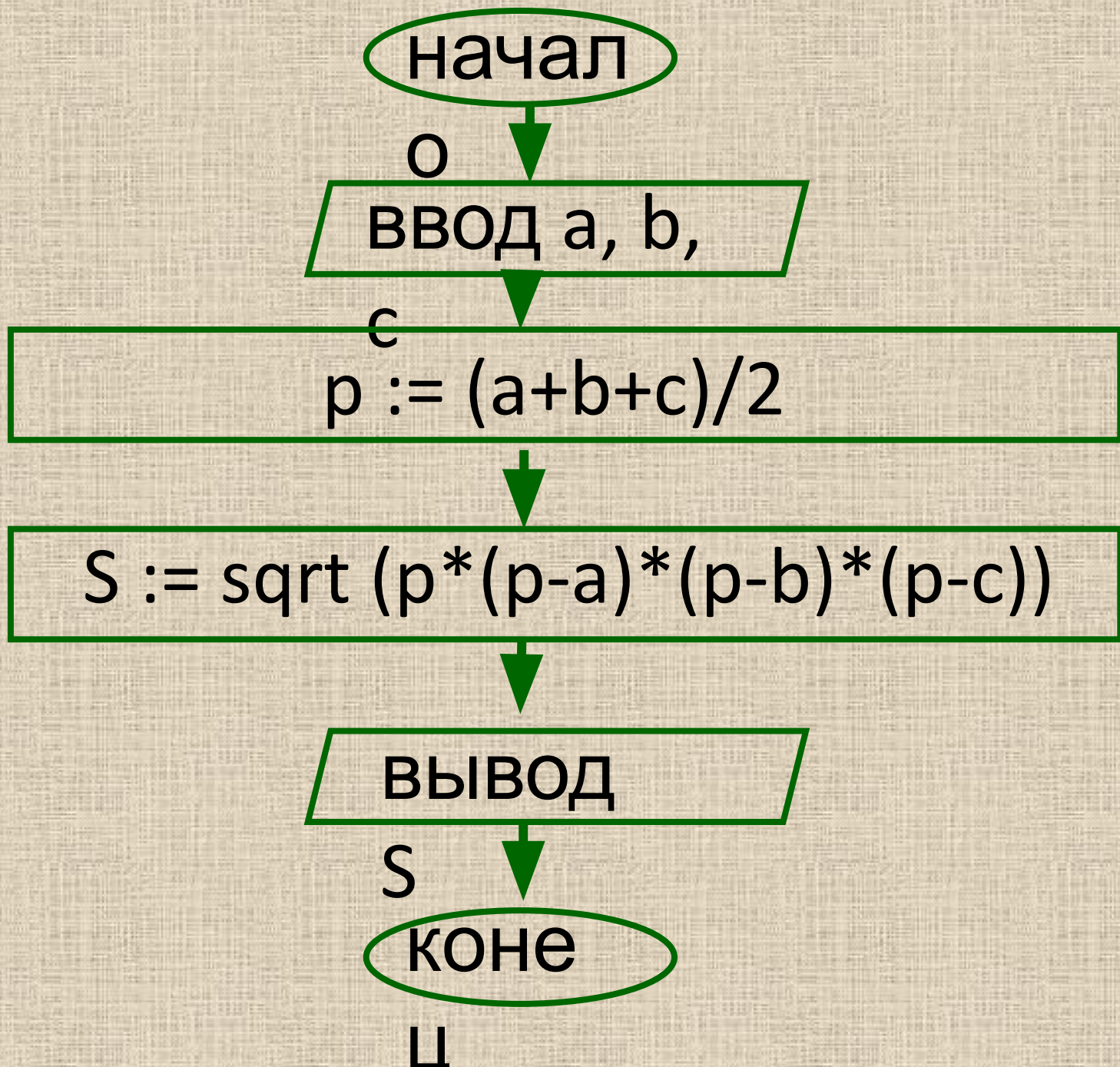
Стр. 5, № 1.14. – 1.17.

Пример. Составить блок-схему алгоритма.  
Даны три стороны треугольника  $a$ ,  $b$ ,  $c$ .  
Вычислить площадь треугольника по формуле Герона:

$$S = \sqrt{p(p - a)(p - b)(p - c)},$$

где  $p$  – полупериметр треугольника

Длины сторон треугольника будет вводить пользователь.



Составить блок-схемы алгоритмов по следующим задачам.

Задача 4. Даны два действительных числа. Получить их сумму, разность и произведение.

Задача 5. Даны два действительных числа. Получить их среднее арифметическое.



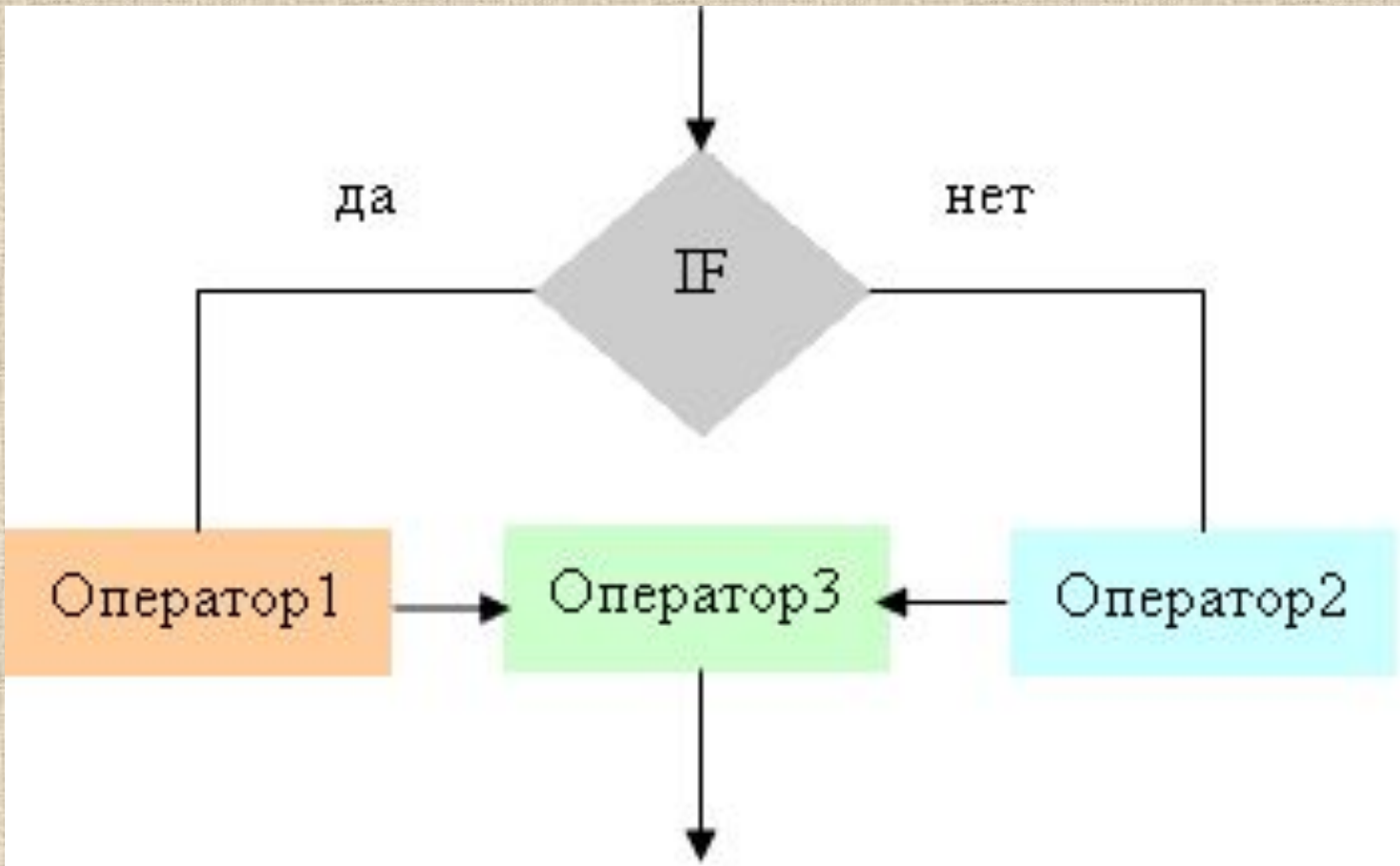
Задача 6. Даны катеты прямоугольного треугольника. Найти площадь треугольника и его гипотенузу.

Задача 7. Написать алгоритм деления одной обыкновенной дроби на другую обыкновенную дробь.

# Разветвляющийся

*Полное ветвление*





## Неполное ветвление





8. Составить блок-схему алгоритма по следующей задаче



Из трех монет одинакового достоинства одна фальшивая (более легкая). Как ее найти с помощью одного взвешивания на чашечных весах без гирь?

Алгоритм разветвляющейся структуры содержит как минимум одно условие, в зависимости от которого будет выполнено одно или другое действие.

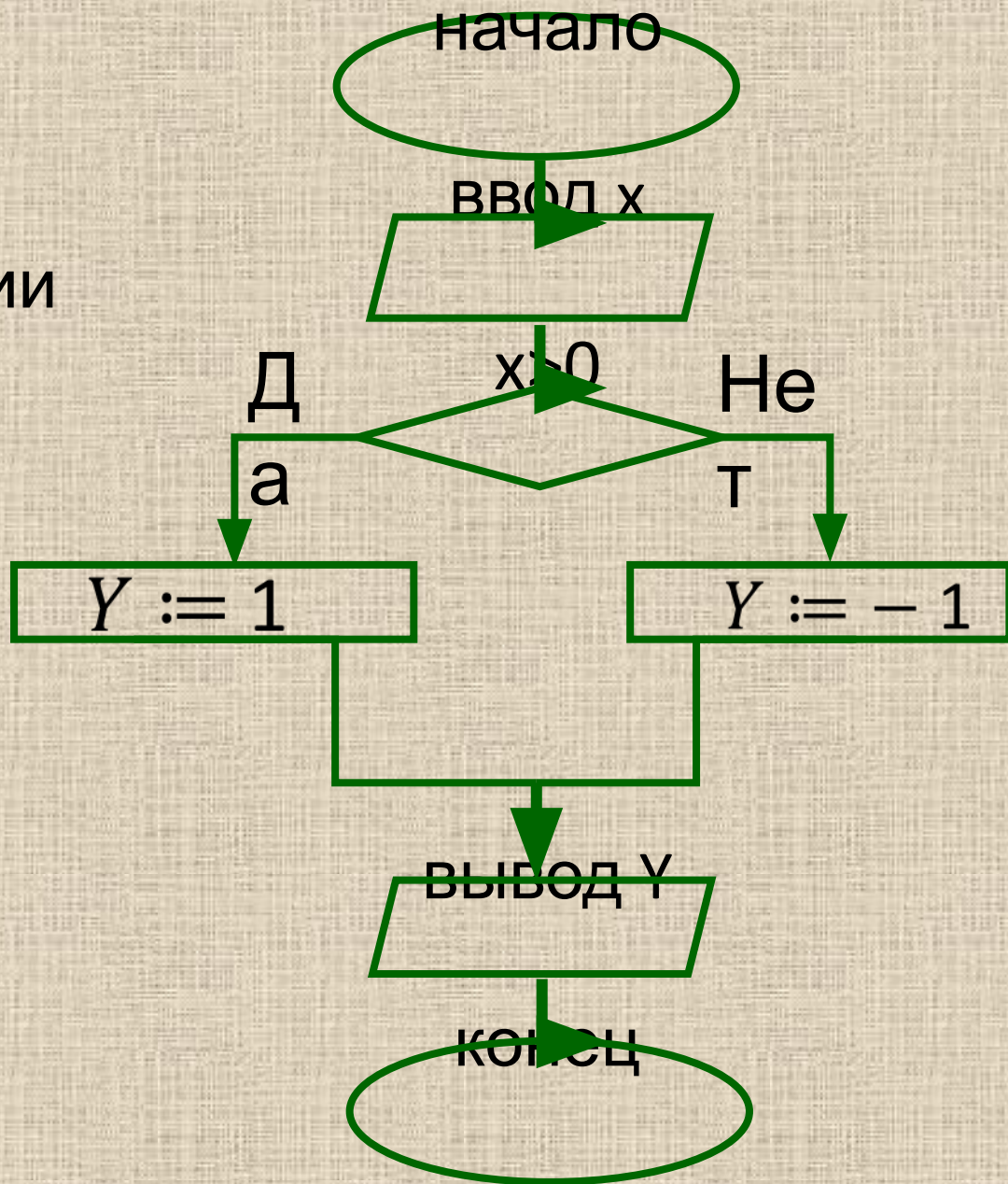
Пример.

Написать алгоритм  
вычисления функции

$Y$ ,

используя систему:

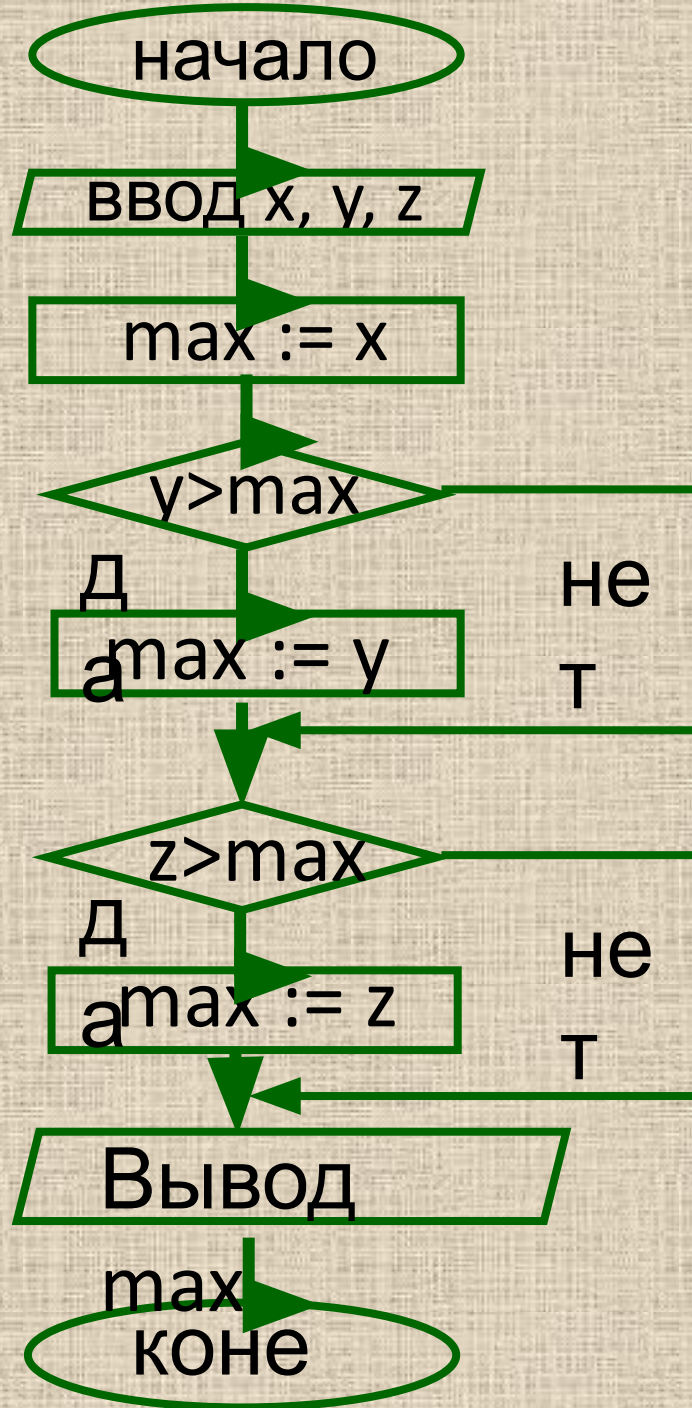
$$Y = \begin{cases} 1, & \text{если } x > 0 \\ -1, & \text{если } x \leq 0 \end{cases}$$



## Алгоритм обхода

Алгоритм обхода является разновидностью разветвляющегося алгоритма и применяется, когда одна из ветвей не содержит ни одного действия.

Пример. Даны числа  $x$ ,  $y$ ,  $z$ . Найти максимальное значение.



Составить блок-схемы алгоритмов для решения следующих задач.

9. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.

$$V = L^3$$

$$S = L^2 \cdot 4$$

10. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.

$$L = 2\pi R$$

$$S = \pi R^2$$



11. Вычислить расстояние между двумя точками с координатами  $x_1, x_2, y_1, y_2$ .

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

12. Даны действительные числа  $x, y, z$ . Вычислить  $A$  и  $B$ , если

$$A = \frac{1 + \sin(x + y)}{2 + \left| x - \frac{2x}{1 + x^2 y^2} \right|}$$

$$B = 1 + \frac{z^2}{3 + z^2/5}$$



13. Дано действительное число  $a$ . Не пользуясь никакими арифметическими операциями, кроме умножения, получить:

- a)  $a^4$  за две операции;
- b)  $a^6$  за три операции;
- c)  $a^7$  за четыре операции;
- d)  $a^9$  за четыре операции.

14. Даны действительные числа  $a, b, c$ . Удвоить числа, если  $a \geq b \geq c$ , и заменить их абсолютными значениями, если это не так.





15. Даны два действительных числа. Вывести первое число, если оно больше второго, и оба числа, если это не так.

16. Даны два действительных числа. Заменить первое число нулем, если оно меньше или равно второму, и оставить числа без изменения в противном случае.

17. Даны три действительных числа. Вывести на экране те из них, которые принадлежат интервалу  $[1..5]$ .



18. Даны два действительных числа  $x$ ,  $y$ . Меньшее из этих двух чисел заменить их полусуммой, а большее – их удвоенным произведением.

19. Даны три действительных числа. Возвести в квадрат те из них, значения которых не отрицательны.

20. Даны два действительных числа  $x$ ,  $y$ . Вычислить  $z$ :

$$z = \begin{cases} x - y, & \text{если } x > y \\ y - x + 1 & \text{в противном случае} \end{cases}$$



21. Дано действительное число  $x$ . Вычислить функцию  $F$ :

$$\text{а) } F = \begin{cases} x^2 + 4x + 5 & \text{при } x \leq 2 \\ \frac{1}{x^2 + 4x + 5} & \text{в противном случае} \end{cases}$$

$$\text{б) } F = \begin{cases} 0 & \text{при } x \leq 0 \\ x & \text{при } 0 < x \leq 1 \\ x^4 & \text{в остальных случаях} \end{cases}$$

# Циклические

## Циклические действия при подготовке домашнего задания



# Алгоритм поиска Золушки

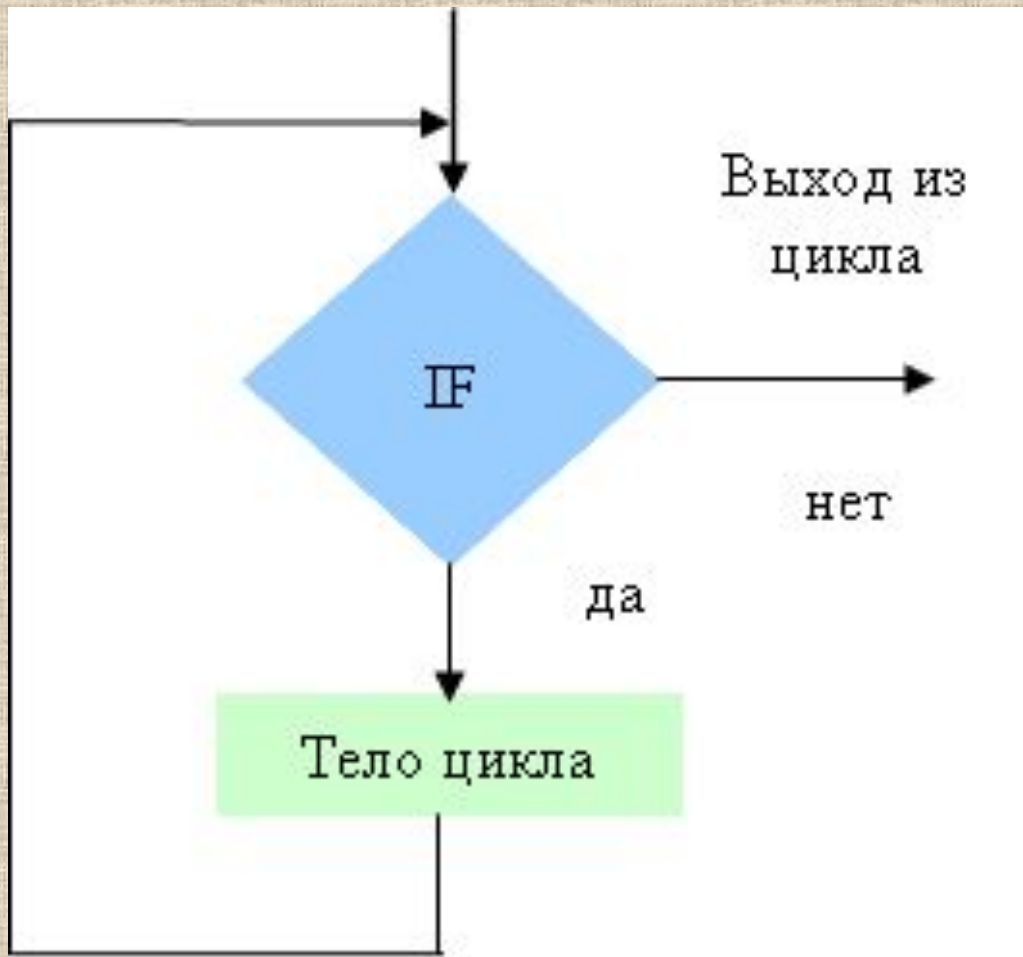


Алгоритм циклической структуры – это такой алгоритм, в котором повторяются одни и те же действия.

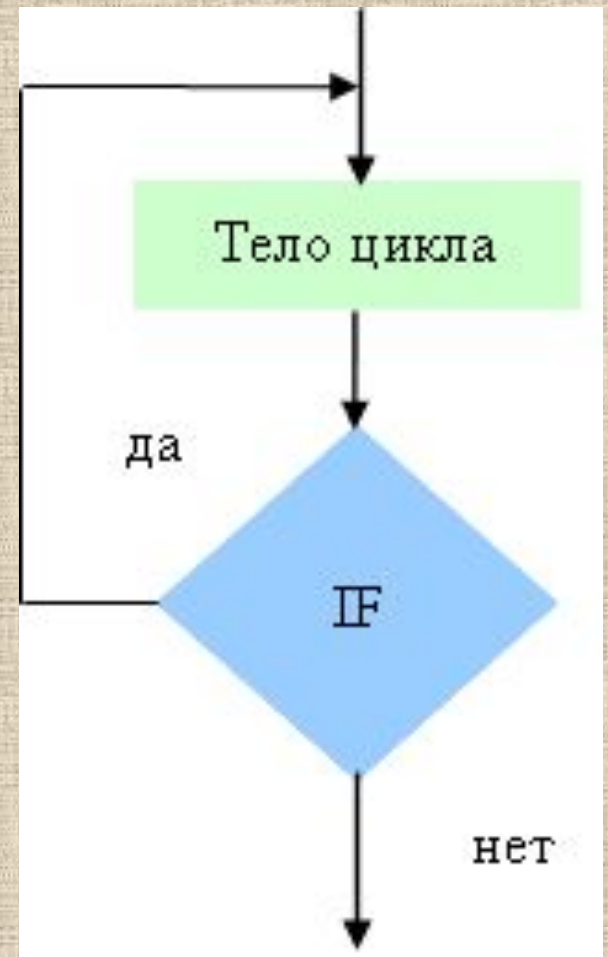
Существует два типа циклических алгоритмов: цикл с предусловием и цикл с постусловием.

Алгоритм подготовки домашнего задания – цикл с предусловием: сначала задается условие, затем выполняются действия. Таким образом, **в цикле с предусловием тело цикла может не выполниться ни разу.**

Алгоритм поиска Золушки включает в себя цикл с постусловием: сначала выполняются 2 действия (встретить и примерить), а затем уже проверяется условие. Таким образом, **действия в цикле с постусловием всегда будут выполнены хотя бы 1 раз.**



Цикл с  
предусловием



Цикл с  
постусловием

Пример. Дано целое положительное число  $N$ . Найти  $N!$  (факториал).

Факториал числа  $N$  – это произведение всех натуральных чисел от 1 до  $N$  включительно. Факториал числа  $0! = 1$ .

Если  $N=5$ , тогда  $N!=1*2*3*4*5=120$

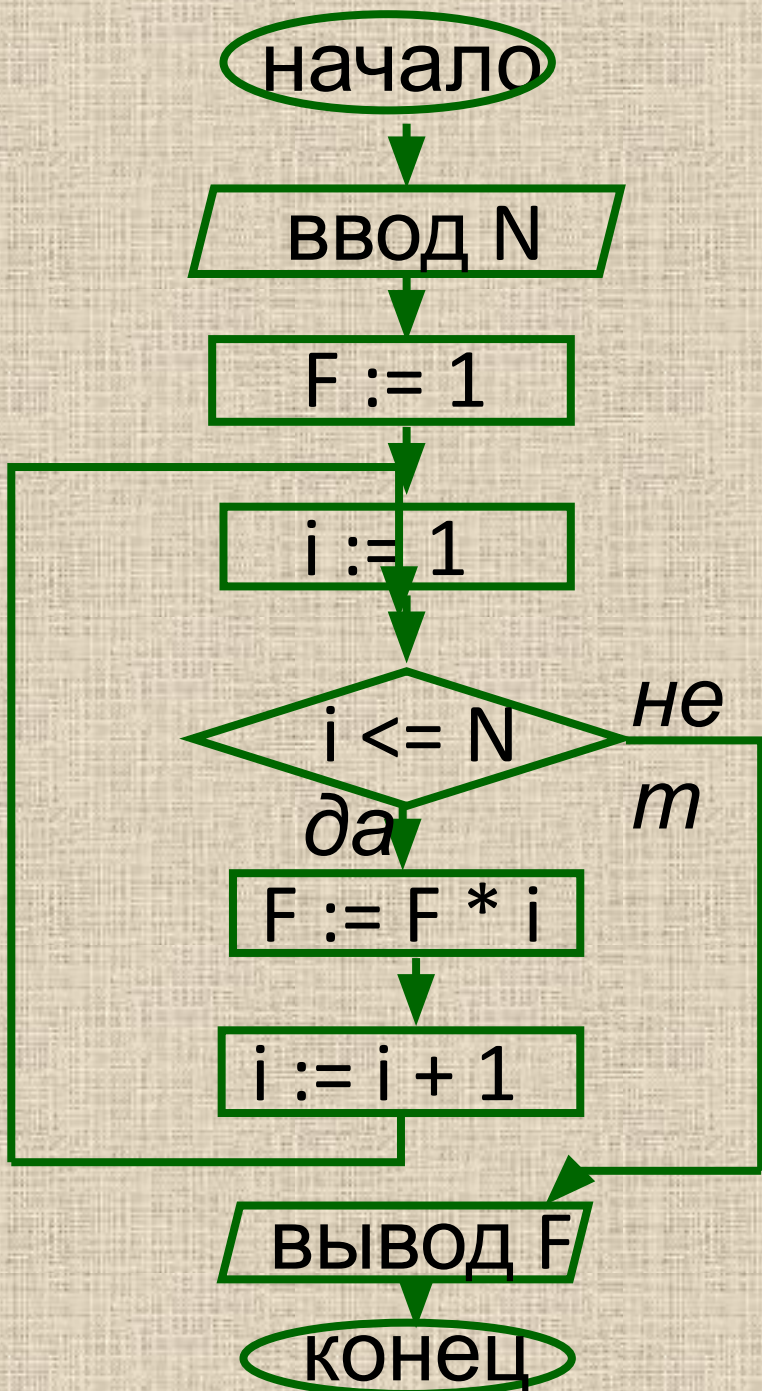
Для составления алгоритма понадобятся три переменные целого типа:

$N$  – аргумент,

$i$  – промежуточная переменная,

$F$  – факториал числа, т.е. результат.





шаг	N	F	i	Условие
1	3			
2		1		
3			1	
4				1 ≤ 3, да
5		1		
6			2	
7				2 ≤ 3, да
8		2		
9			3	
10				3 ≤ 3, да
11		6		
12			4	
13				4 ≤ 3, нет
14		ВЫВОД		

22. Одна штука некоторого товара стоит 20,4 рубля. Напечатать таблицу стоимости от 2 до 30 штук товара.

23. Спортсмен в первый день пробежал 10 км. Каждый следующий день он увеличивал норму на 10% от нормы предыдущего дня. Какой путь пробежит спортсмен на 7 день.



24. Спортсмен в первый день пробежал 10 км. Каждый следующий день он увеличивал норму на 10% от нормы предыдущего дня. Определить, через сколько дней спортсмен будет пробегать более 20 км.

25. Вычислить значения выражения и вывести таблицу результатов функции

$$Z = \frac{3a+4}{a^2-5a-9},$$

где  $a$  изменяется от 1 до 40 с шагом 4.

26. Цилиндр объема единица имеет высоту  $h$ . Определить радиус основания цилиндра для значений  $h$ , равных 0.5, 1, 1.5, ..., 5. Вывести таблицу результатов найденных значений.



27. Получить таблицу температур по Цельсию от 0 до 30 градусов и их эквивалентов по шкале Фаренгейта, используя для перевода формулу  $t_F = \frac{9}{5} \cdot t_C + 32$ . Вывести таблицу результатов.

28. Вычислить последовательность значений и вывести таблицу результатов функций  $p1 = x$

$$p2 = \frac{3x^2 - 1}{2}$$

$$p3 = \frac{5x^2 - 3x}{2}$$

для значений аргумента  $x=0, 0.05, 0.1, \dots, 3$ .



29. Вычислить значения функции

$$y = 4x^3 - 2x^2 + 5$$

для значений  $x$ , изменяющихся от  $-3$  до  $1$  с шагом  $0.1$ . Вывести таблицу результатов найденных значений.

30. Дано натуральное число  $n$ . Вычислить значения функции

$$y = \frac{x^2 - 3x + 2}{\sqrt{2x^3 - 1}}$$

для  $x = 1, 1.1, 1.2, \dots, 1 + 0.1n$ .

Вывести таблицу результатов найденных значений.



31. Вывести таблицу соответствия между весом в фунтах и весом в килограммах для значений 1, 2, 3, ..., 22 фунта. 1 фунт = 453 грамма.

32. Вывести таблицу перевода расстояний дюймов в сантиметры для значений 10, 12, 14, ..., 30 дюймов. 1 дюйм = 2,54 см.

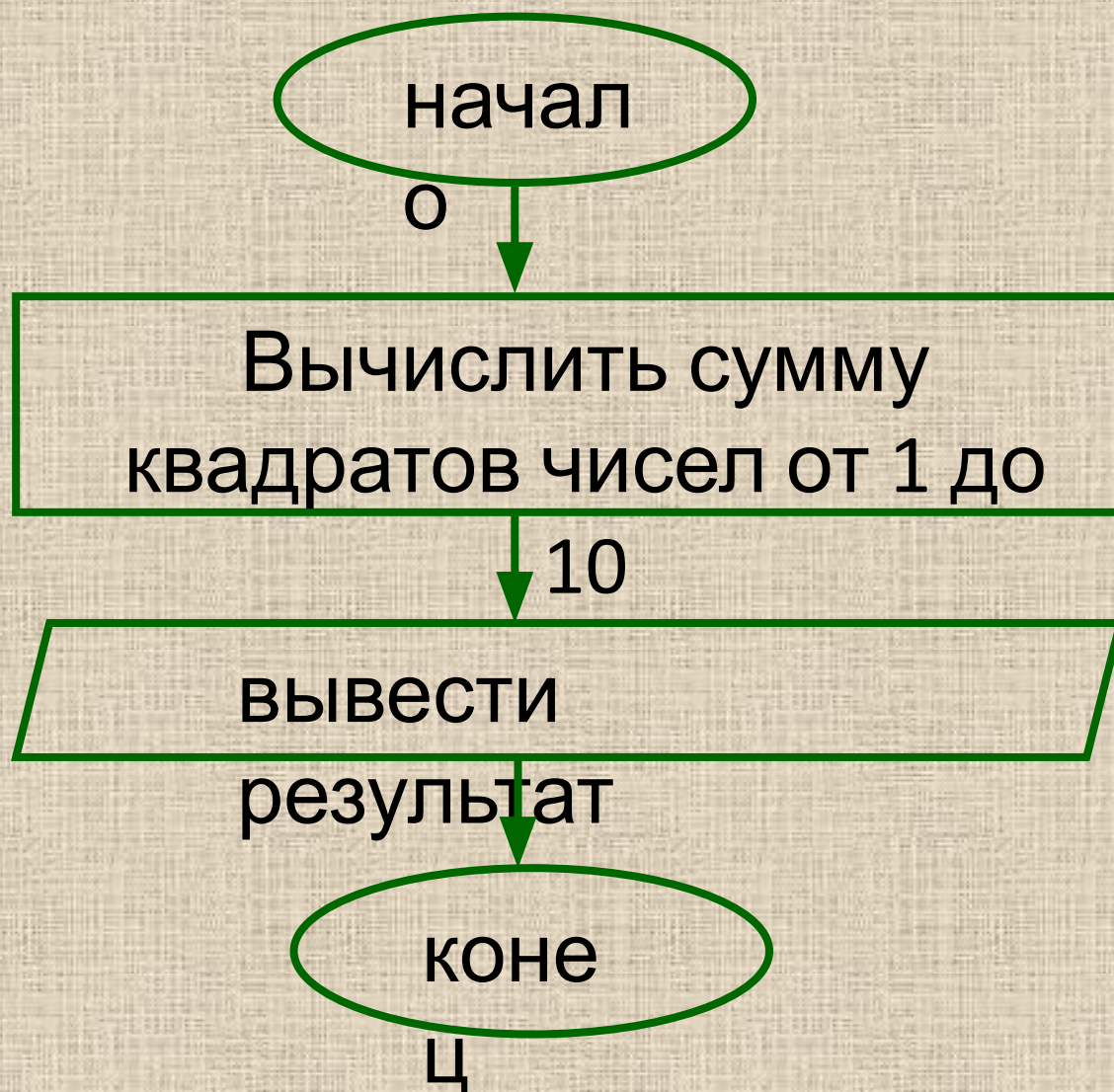
# Пошаговая детализация

Пошаговая детализация (программирование сверху вниз, нисходящая разработка) представляет собой процесс поэтапного решения сложной задачи.

На первом этапе алгоритм описывает решение задачи в самых общих чертах.

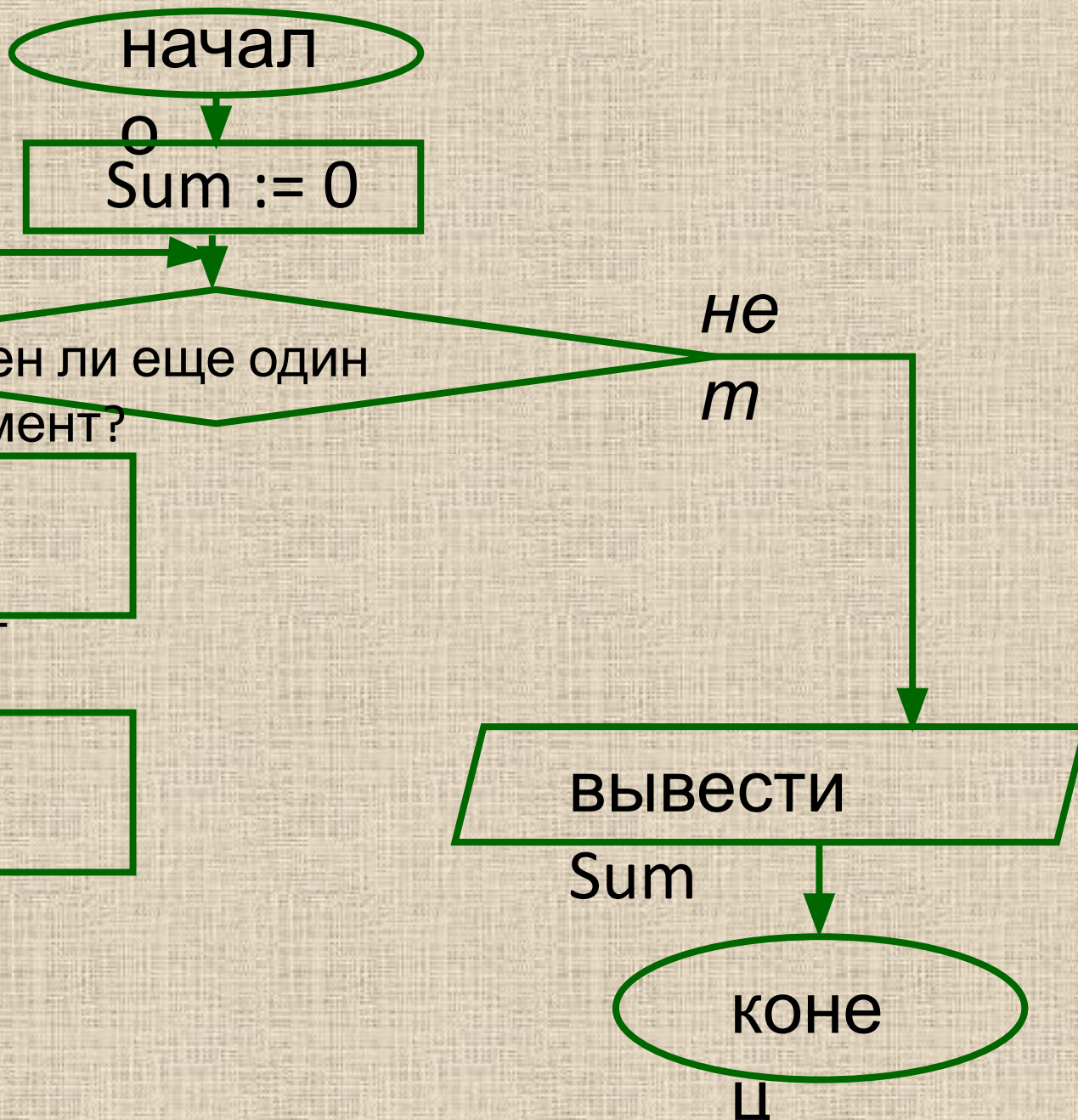
Далее процесс детализации повторяется по отношению к новым вариантам схемы алгоритма до тех пор, пока не будет достигнут такой уровень ясности решения, который позволит приступить непосредственно к программированию.

Пример. Составить алгоритм нахождения суммы квадратов чисел от 1 до 10.  
Исходная схема алгоритма:

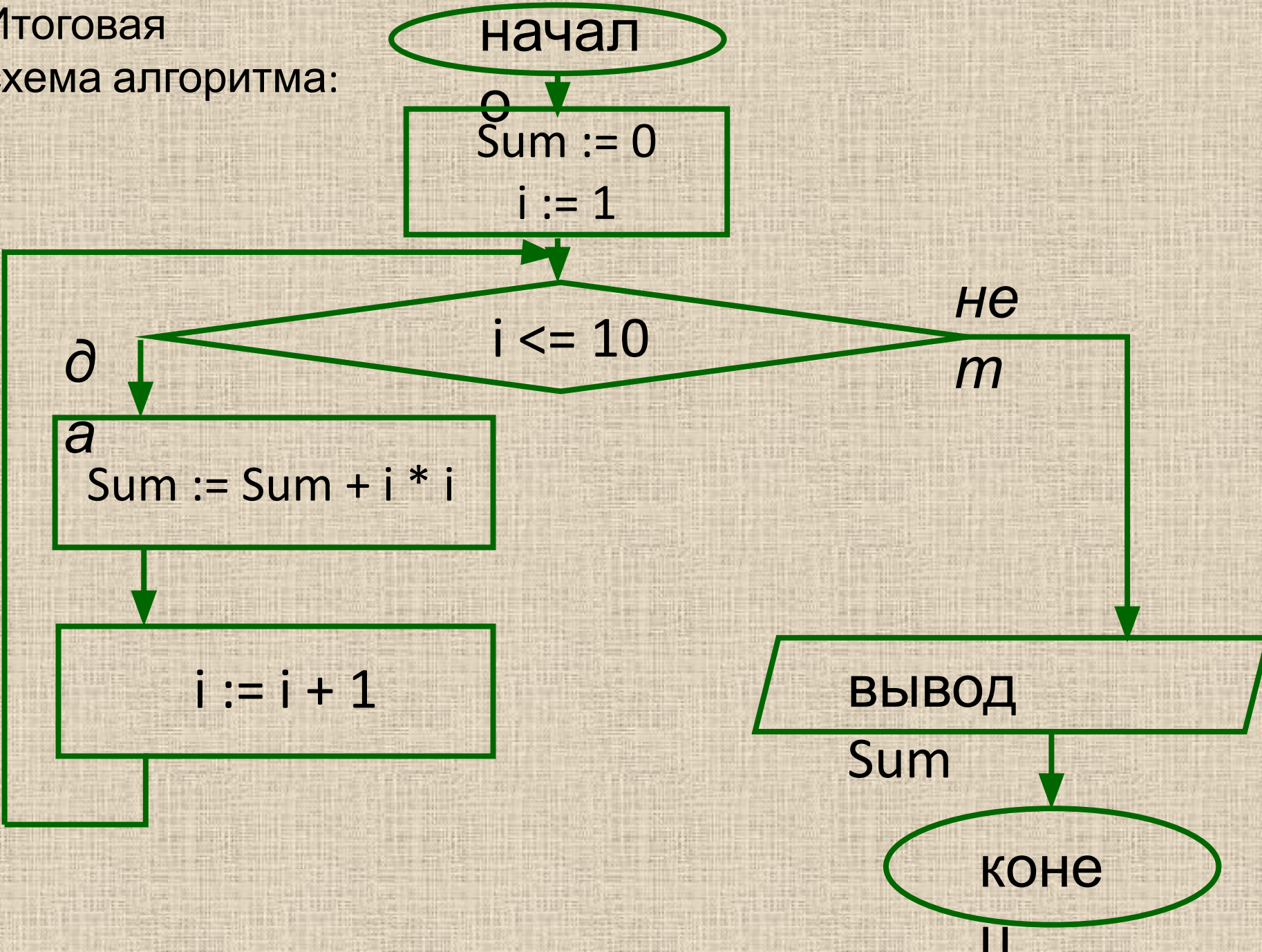




Детализированная  
схема алгоритма:



Итоговая  
схема алгоритма:



33. Найти сумму всех целых чисел от 1 до 10.

36. Найти среднее арифметическое всех целых чисел от 30 до 100.

38. Известна масса каждого из 12 предметов. Определить общую массу всего набора предметов.



34. Найти сумму всех целых чисел от 20 до 50.
35. Найти сумму всех целых чисел от значения А до значения В.
37. Найти произведение всех целых чисел от 3 до 10.
39. Найти сумму всех четных чисел от 1 до 50.
40. Найти сумму всех чисел, кратных трем, от значения А до значения В.
41. Вывести на экран все целые числа от 100 до 200, кратные трем.