Информатика 8 класс

Вопросы к зачету по теме: "Алгоритмизация и программирование"

1. Понятие алгоритма и его свойства. Способы записи алгоритма.
2. Основные блоки схемы: начало/конец, ввод/вывод, присваивание, ветвление, цикл . Уметь чертить блок-схемы для основных алгоритмических структур: линейной, циклической, ветвящейся.
3. Составление блок-схемы по словесному условию.

Примеры условий:

для линейного алгоритма: составить блок-схему для вычисления суммы, разности и произведения двух чисел, а также, частного от деления первого числа на второе. Числа вводятся с клавиатуры.

для циклического алгоритма: составить блок-схему для вычисления суммы натуральных нечетных чисел из диапазона от 15 до 25.

для ветвящегося алгоритма: составить блок-схему для поиска наибольшего из трех чисел.

4. Структура программы на Паскале для линейного алгоритма: блок описания переменных, тело программы. Синтаксис и пунктуация языка.
5. Запись на Паскале циклического алгоритма.
6. Определение значения переменной по программе циклического алгоритма (трассировка).
7. Запись на Паскале ветвящегося алгоритма.
8. Определение значения переменной по программе ветвящегося алгоритма (трассировка).

Алгоритм и его свойства, Способы представления.

1. **Алгоритм** – последовательность действий для достижения цели.

Исполнитель – тот, кто выполняет алгоритм.

Исполнитель

умеет выполнять определенные действия - **команды**

```
//Задача
//Помоги Кукараче получить слово КИСА
//-------------------------------------

ЭТО КИСА
ВВЕРХ
ВПРАВО
ВПРАВО
ВНИЗ
КОНЕЦ
// К о н е ц  т е к с т а  (?-F1) //
```

СКИ

(система команд исполнителя) – все команды исполнителя

ВВЕРХ
ВНИЗ
ВПРАВО
ВЛЕВО

2. Свойства алгоритма:

- ***понятность***

  (алгоритм составлен только из команд СКИ)

- ***дискретность***

  (деление алгоритма на простейшие шаги)

- ***точность***

  (однозначное действие исполнителя по каждой команде)

- ***конечность***

(выполнение алгоритма завершается за известное число шагов)

- ***массовость***

(по одному алгоритму можно решить много подобных задач)

Любой алгоритм выполняется **формально**

– не требуется понимание задачи, для которой сделан алгоритм

3. Полный набор исходных данных

данные необходимы и их достаточно

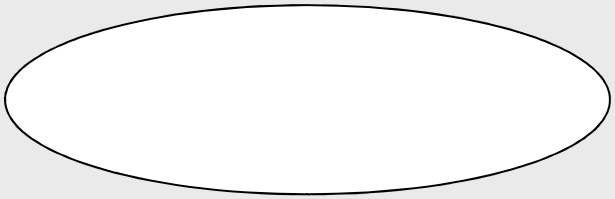
Алгоритм – понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящую от исходных данных к искомому результату

4. Способы представления алгоритмов

1. Словесный
2. Графический (блок-схема)
3. На языке программирования

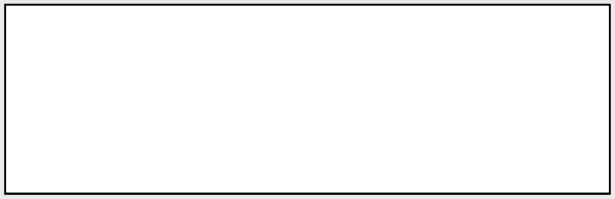
Блок-схема

Состоит из **блоков**,
соединенных линиями, показывающими
последовательность выполнения
шагов алгоритма



Начало/Конец алгоритма

Ввод/Вывод



Вычислительный блок

Команда присваивания

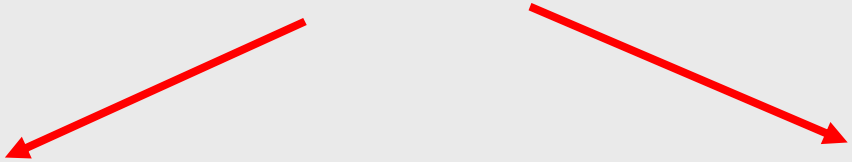
1. Данные:

информация, хранящаяся в памяти компьютера.
Величина – единица данных.

2. Основные типы величин:

- **числовой**
- **символьный (строковый)**
- **логический**

3. Тип величины указывает на:



диапазон значений

разрешенные операции над величинами

4. Запись величин:

- константы – 5 3.18 59 345
- переменные – a sum s1

5. Присваивание:

Задает значение переменной.

Значение хранится до следующей команды присваивания

6. Запись присваивания:

```
x := 5

y :=(5*x – 2*y)/5

z := y
```

7. Трассировочная таблица:

содержит команды присваивания и значения переменных на каждом шаге

Задание: определите значения переменных после выполнения команд присваивания

Команда	A	B
A:=1	1	-
B:=2	1	2
A:= 3*A + B	5	2
B:=15*B - 12	5	18

значения переменных **A** и **B** после выполнения команд присваивания

Линейный алгоритм

последовательность команд, выполняемых по одному разу с первой до последней.

Содержит:
команды ввода/вывода,
присваивание

Задача № 1

Скорость первого автомобиля v_1 км/ч, второго - v_2 км/ч, расстояние между ними S км. Какое расстояние будет между ними через t ч, если автомобили движутся навстречу друг другу? Составьте алгоритм для решения этой задачи.

Решение:

1. Введем условные обозначения:

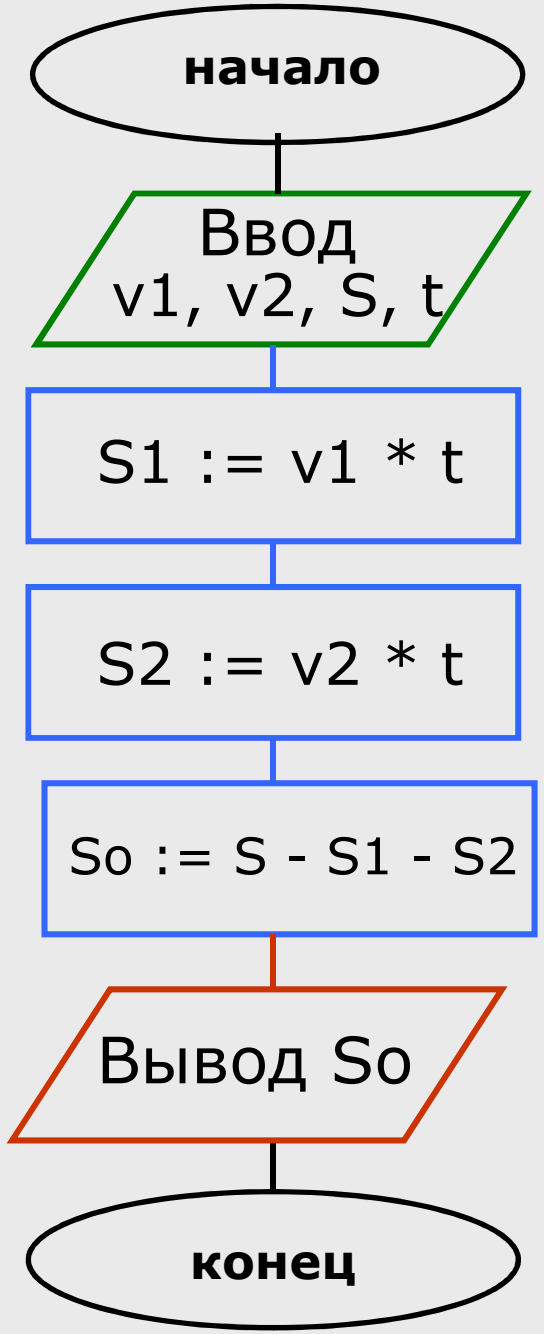
S – весь путь

S1 – путь, пройденный 1 а/м за t ч.

S2 – путь, пройденный 2 а/м за t ч.

So – оставшееся расстояние через t ч.

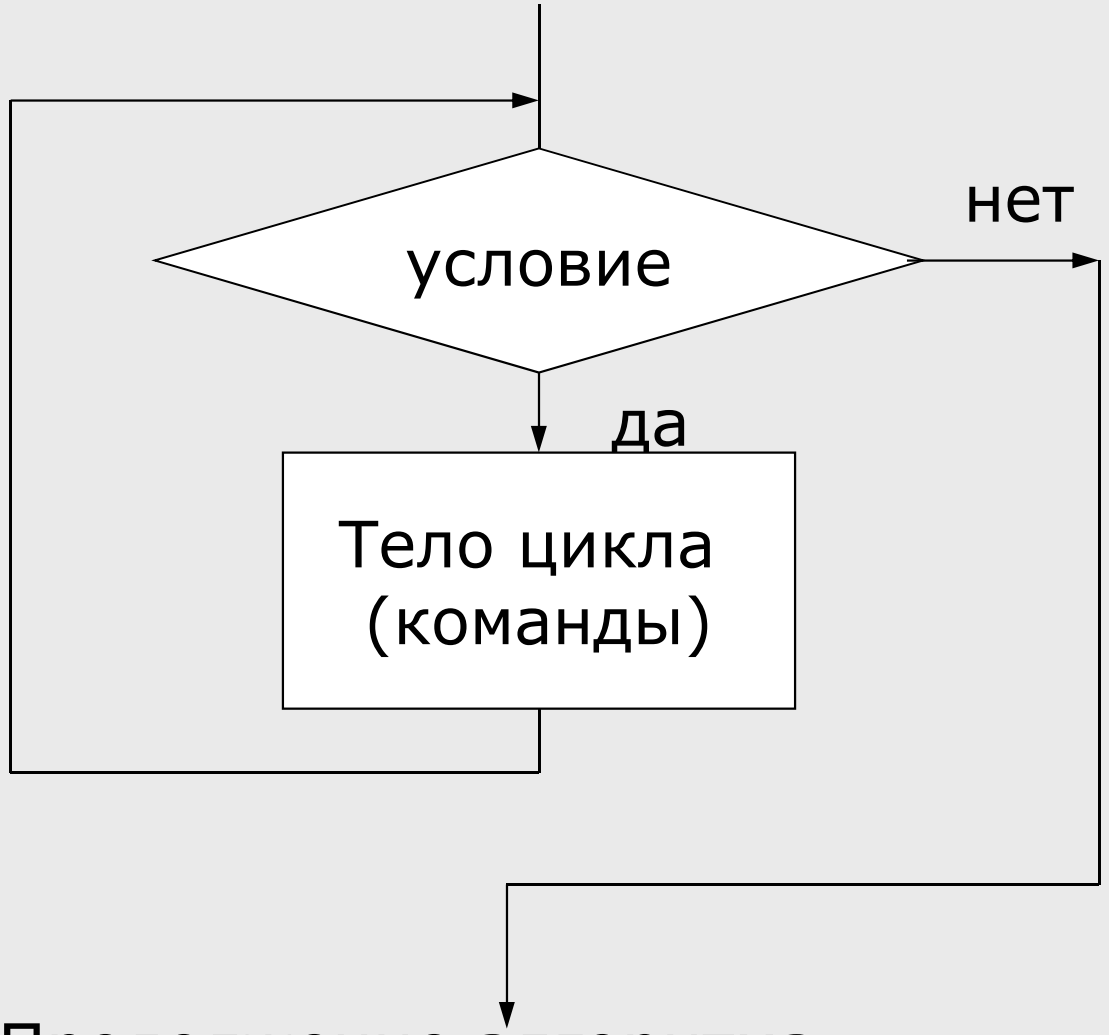
2. Составим блок-схему:



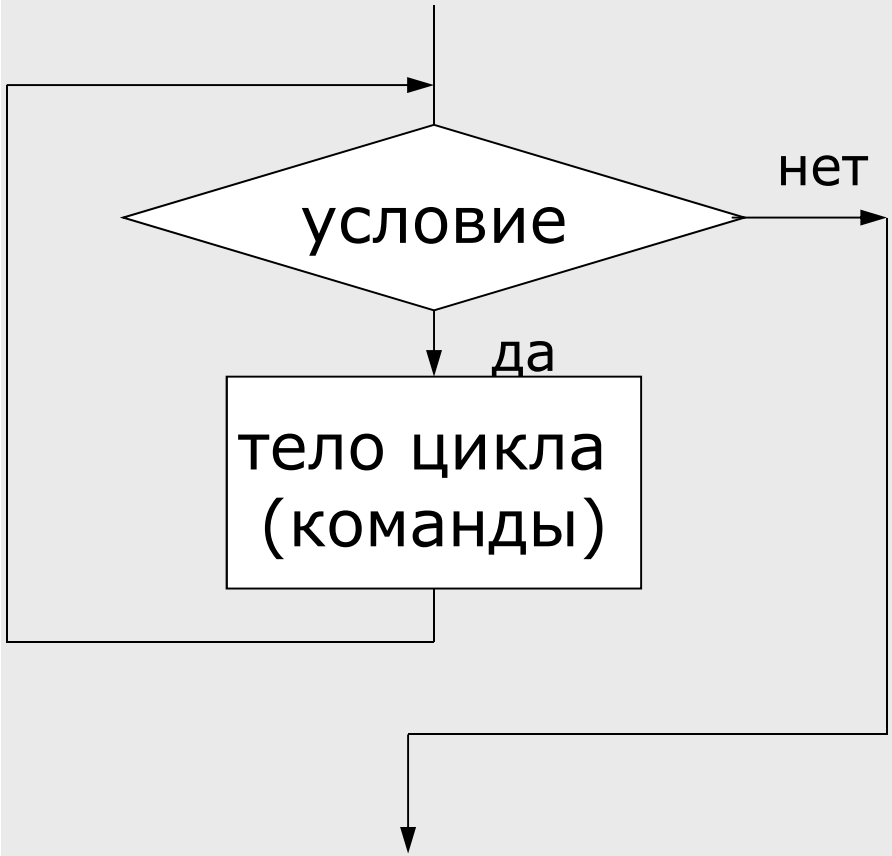
Циклический алгоритм

1. Циклический алгоритм – алгоритм, содержащий алгоритмическую структуру «цикл».
2. Цикл – многократное повторение последовательности команд (*тела цикла*) по некоторому условию.
3. Виды циклов:
   – цикл с условием (количество повторений тела цикла неизвестно);
   - цикл со счетчиком (количество повторений известно)

Графическая запись цикла с условием



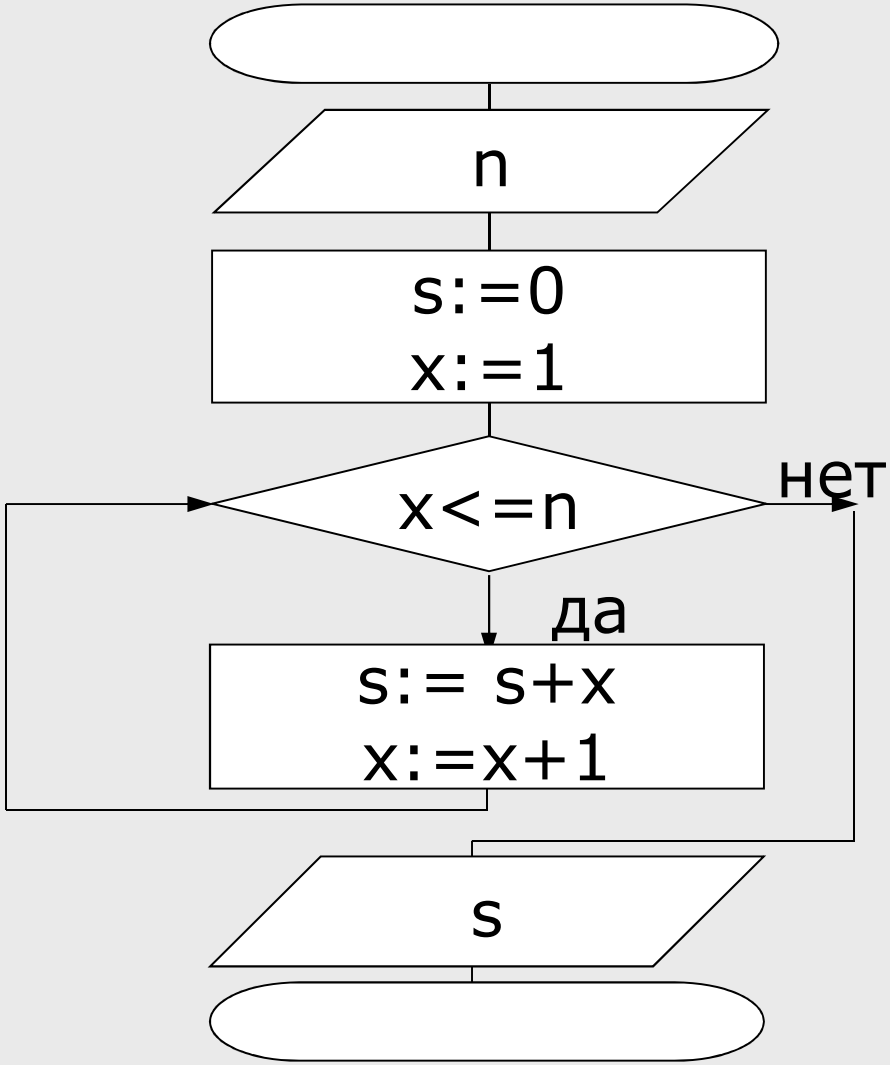
Запись на языке программирования цикла с условием




```
while условие do
begin
    тело цикла
    (команды)
end;
```

Задача 1:

Подсчитайте сумму **n** первых натуральных чисел




```
var
     x,s,n : integer;
begin
    cls;
    write('n=');
    readLn (n);
    s:=0;
    x:=1;
    while x <=n do
        begin
            s:=s+x;
            x:=x+1;
        end;
     writeln ('Сумма=', s);
end.
```

Ветвящийся алгоритм

1. **Ветвление** – разделение алгоритма на два пути (ветки) по некоторому условию с последующим выходом на общее продолжение алгоритма.

   Ветка «Да» - условие выполняется.

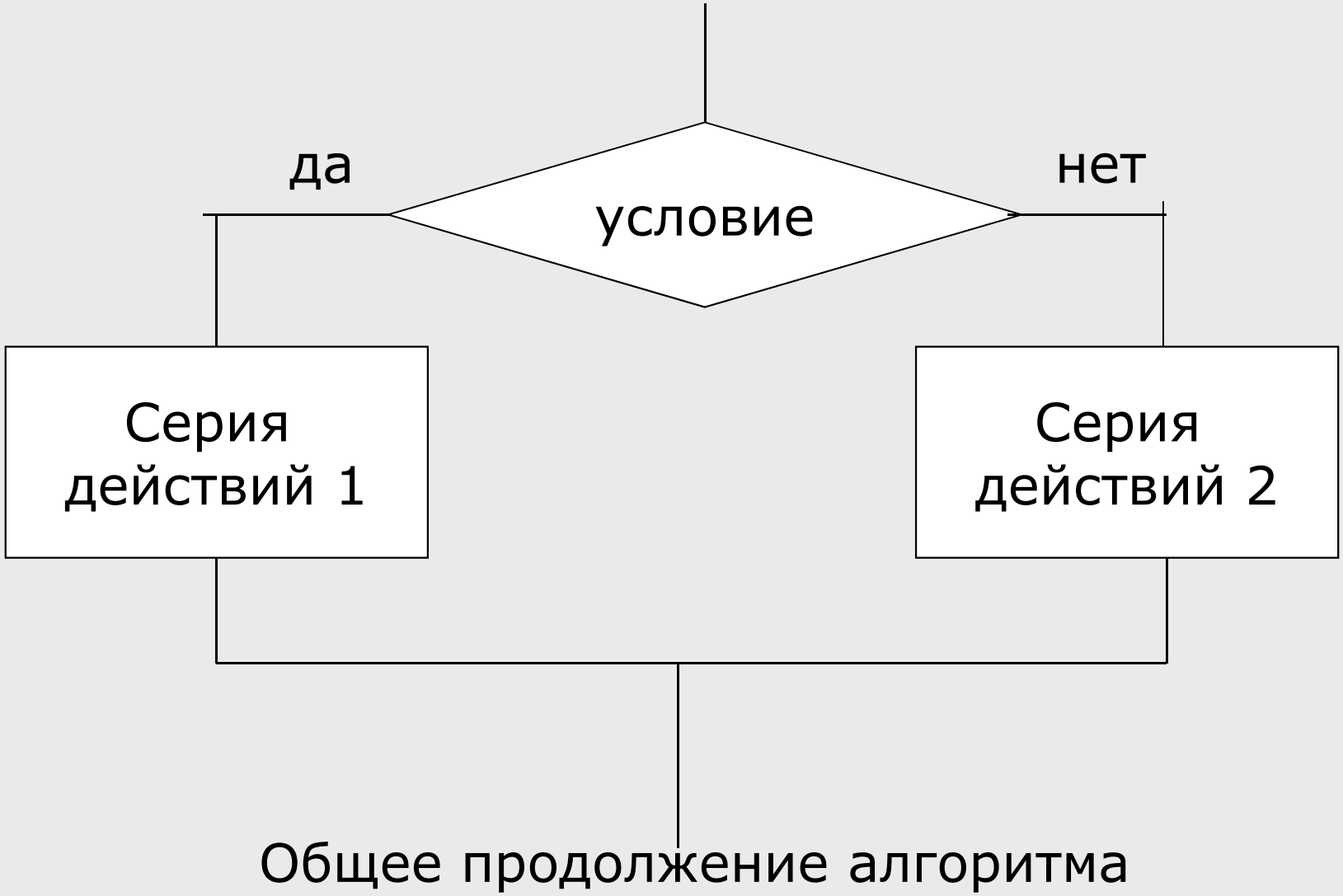
   Ветка «нет» - условие не выполняется.

2. Условие – выражение, принимающее истинное или ложное значение и записанное с помощью знаков =, <,>,<=,>=, <>

Примеры:

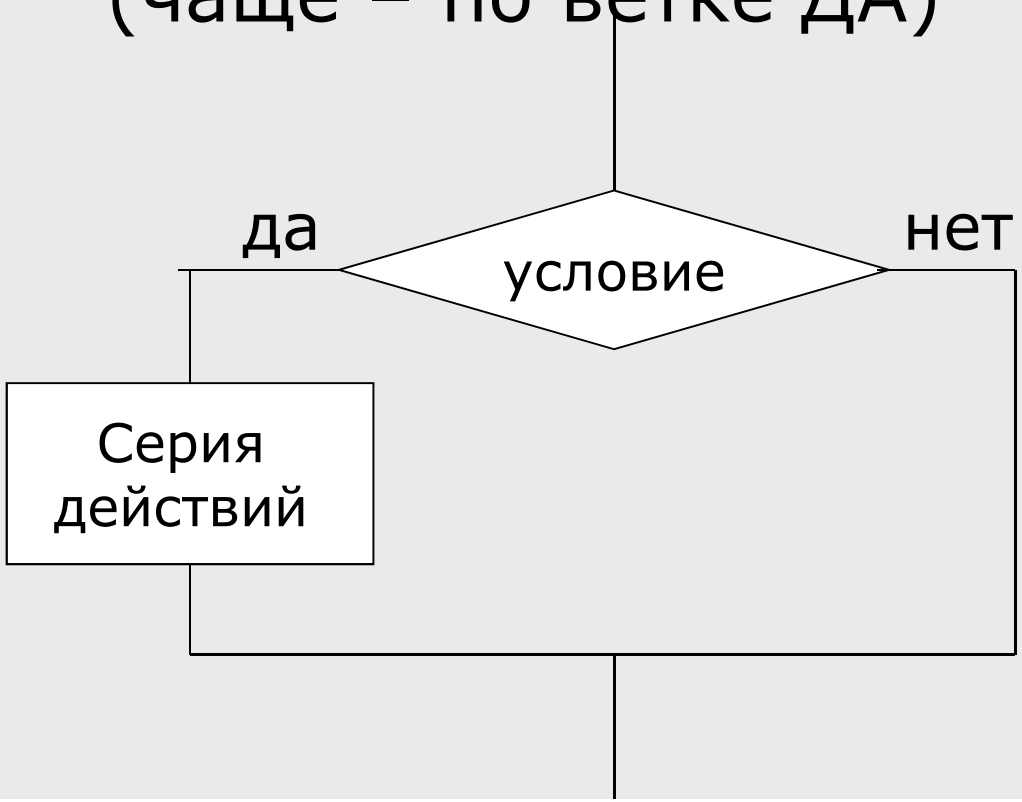
X > 8; Sum=100; y<>z;

Графическая запись ветвления (полного)



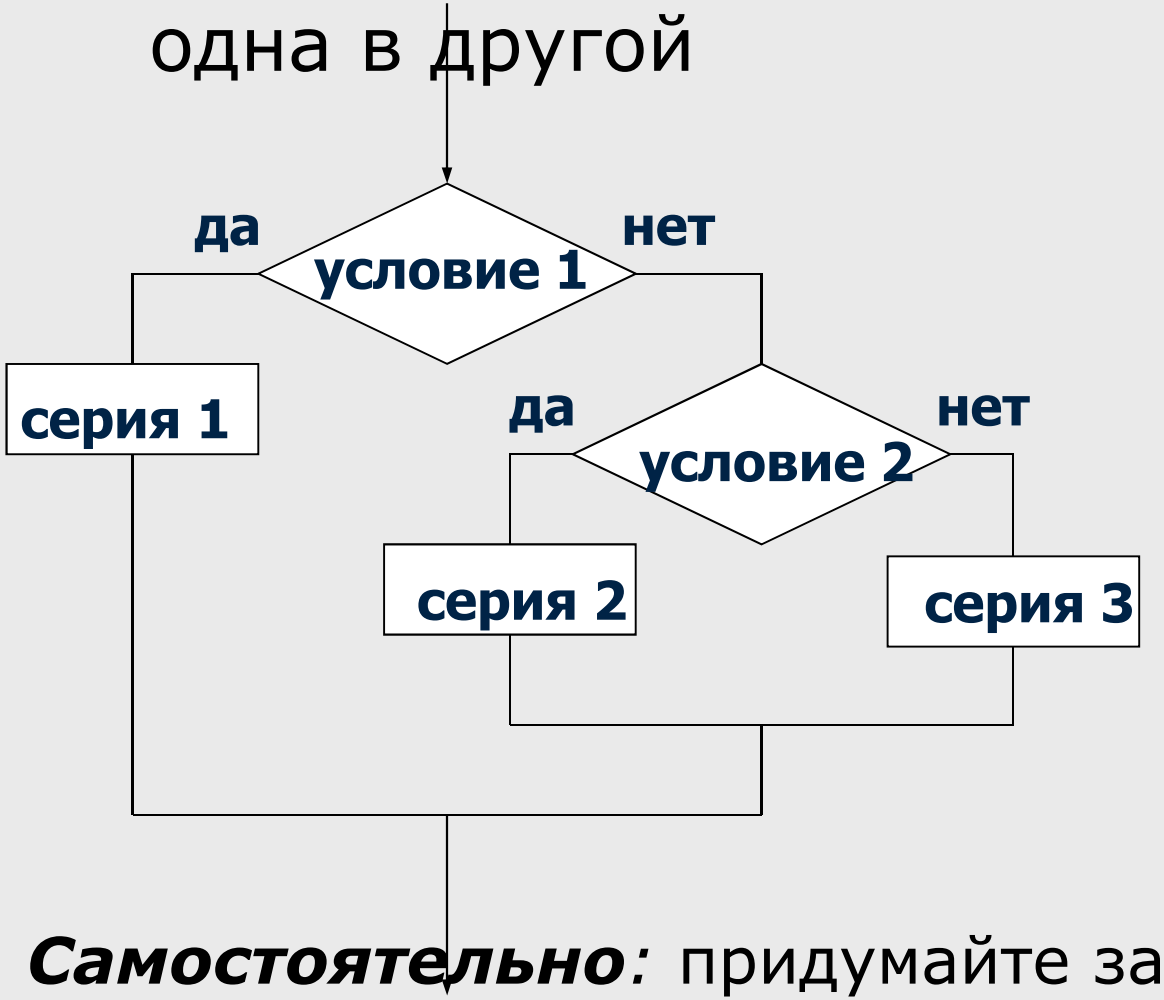
Неполное ветвление -

ветвление, в котором действия выполняются только по одной ветке (чаще – по ветке ДА)



Вложенное ветвление -

структуры ветвления могут находиться одна в другой



Самостоятельно: придумайте задачу, для решения которой потребовался бы алгоритм с вложенным ветвлением. Запишите её условие в тетради

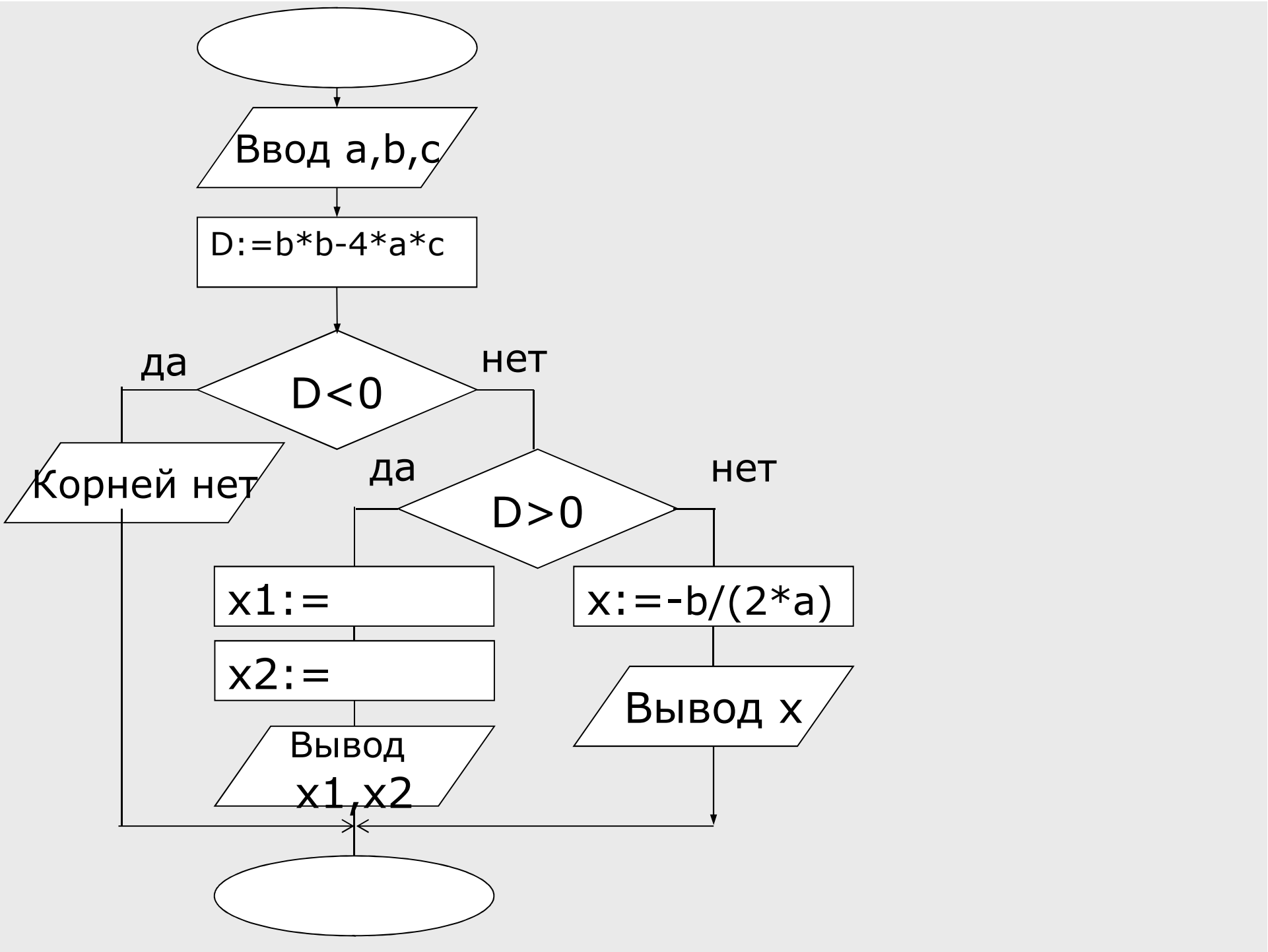
Исследование квадратного уравнения $ax^2 + bx + c = 0$

Решение:

1. Введем условные обозначения:

a, b, c – вводимые с клавиатуры коэффициенты
D – дискриминант уравнения
x1, x2 – корни уравнения при D>0
x – корень уравнения при D = 0

2. Составим блок-схему:


Ввод a,b,c
D:=b*b-4*a*c
да
D<0
нет
Корней нет
да
D>0
нет
x1:=
x2:=
Вывод
x1,x2
x:=-b/(2*a)
Вывод x