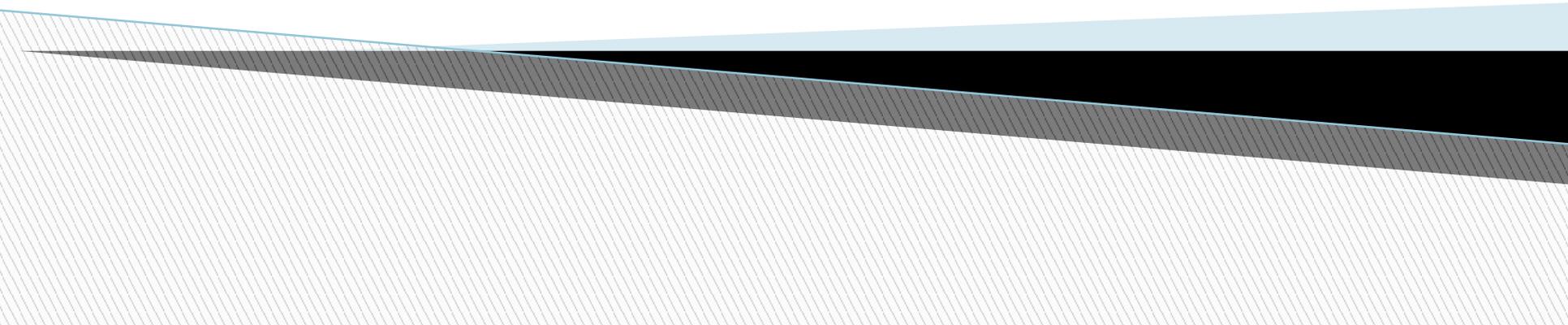


ОПиАЯ

Тема

2

АЛГОРИТМЫ. БЛОК-СХЕМЫ



План

1. Понятие алгоритма
 2. Этапы решения задачи на ПК
 3. Свойства алгоритмов
 4. Формы записи алгоритмов
 5. Блок-схемы, правила их построения
 6. Примеры алгоритмов
- 

1. Понятие алгоритма

Алгоритм

заранее заданное

понятное и точное предписание

возможному исполнителю

совершить определенную

последовательность действий

для решения задачи

за конечное число шагов.



Примеры алгоритмов

ЗЕМЛЯНИЧНЫЙ ЧАЙ

ЧАЙ ЧЁРНЫЙ
2 ч.л.

САХАР
1 ч.л.

КИПЯТОК
1 ст.

ЗЕМЛЯНИКА
50 г.

ЧАЙ ЗАЛИТЬ
КИПЯТКОМ
НАСТОЯТЬ 2 МИН.

РАСТОЛОЧЬ
ЗЕМЛЯНИКУ
С САХАРОМ

ЗЕМЛЯНИКУ
ЗАЛИТЬ ЧАЕМ
НАСТОЯТЬ 5 МИН.

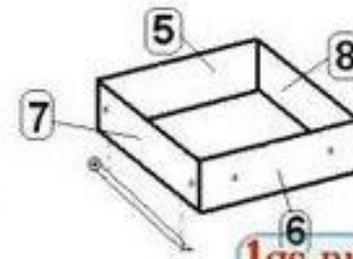
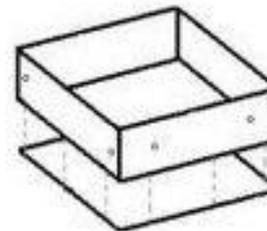
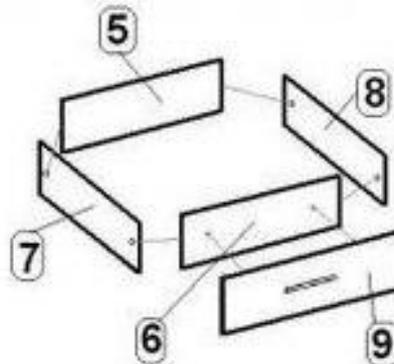
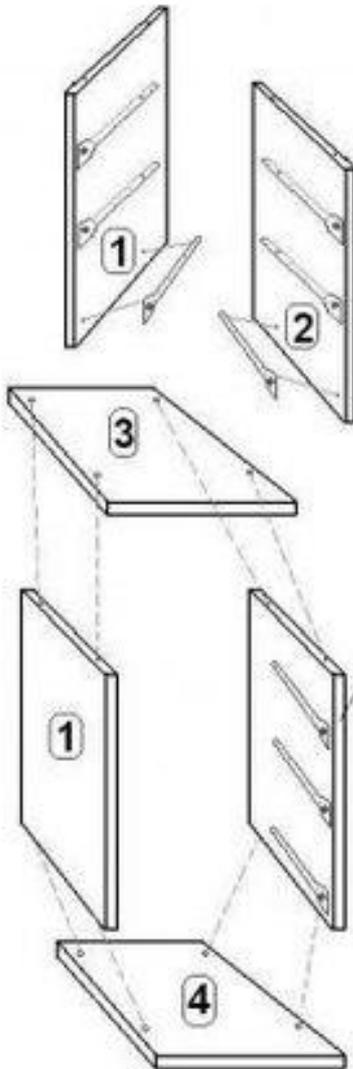
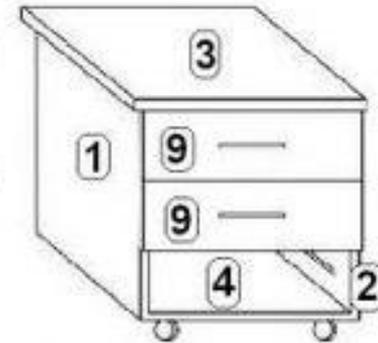
ПРИЯТНОГО
ЧАЕПИТИЯ!

Примеры алгоритмов

ТУМБА

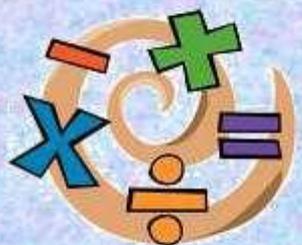
ИНСТРУКЦИЯ СБОРКИ

1. На детали 1,2 согласно разметке закрепить направляющие.
2. Детали 1,2,3,4 соединить с помощью евровинтов, проверить диагональ, прибить ДВП.
3. Детали 5,6,7,8 соединить с помощью евровинтов, проверить диагональ, прибить ДВП и прикрепить направляющие.
4. На деталь 4 согласно разметке закрепить ролики.
5. На детали 9 закрепить ручки
6. Деталь 9 (панель) закрепить на деталь 6.



Пример алгоритма, изучаемого в школьном курсе математики:

 Алгоритм выполнения арифметических операций



- 1** Произвести вычисления в скобках или раскрыть скобки в выражении (если они есть);
- 2** Выполнить все действия второй степени (умножение и деление);
- 3** Выполнить действия первой степени (сложение и вычитание).





Кто (что)
может быть
исполнителем
алгоритма?

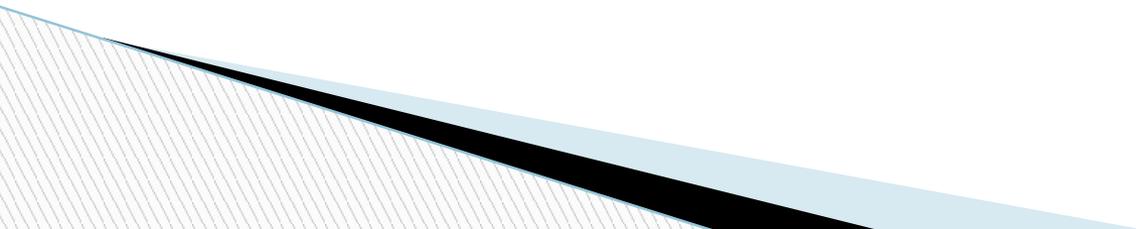
Исполнителя характеризуют

- ▣ *среда* - "место обитания" исполнителя;
 - ▣ *элементарные действия*;
 - ▣ *система команд* - команды только из некоторого строго заданного списка;
 - ▣ *отказы* - возникают, если команда вызывается при недопустимом для нее состоянии среды.
- 

Универсальный исполнитель алгоритмов



2. Этапы решения задач с использованием компьютера



Этапы решения задач

- постановка задачи;
 - формализация задачи (математическая модель);
 - выбор (или разработка) метода решения;
 - разработка алгоритма;
 - составление программы;
 - тестирование и отладка программы;
 - анализ результатов.
- 

Классы алгоритмов

- ВЫЧИСЛИТЕЛЬНЫЕ;
 - информационные;
 - управляющие.
- 

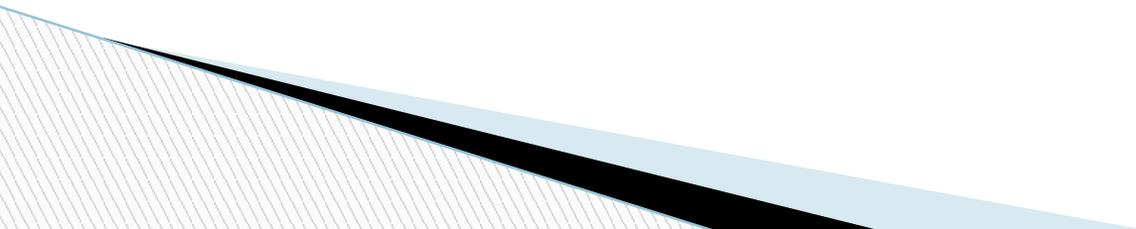
Аргументы
(дано)

Схема
работы
алгоритма

АЛГОРИТМ

Результаты
(найти)

3. Свойства алгоритмов



Понятность для исполнителя

Исполнитель алгоритма должен понимать, как его выполнять.

Алгоритм должен включать только те команды, которые входят в его систему команд.



Ничего не понимаю...

Дискретность

Discretus (лат.) –
разделенный, прерывистый.

Алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов (этапов).



Определенность

Каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвольной трактовки.

Благодаря этому свойству выполнение алгоритма носит механический (*формальный*) характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

*Что
это?*



Рецепт: «Всыпьте немного МУКИ»



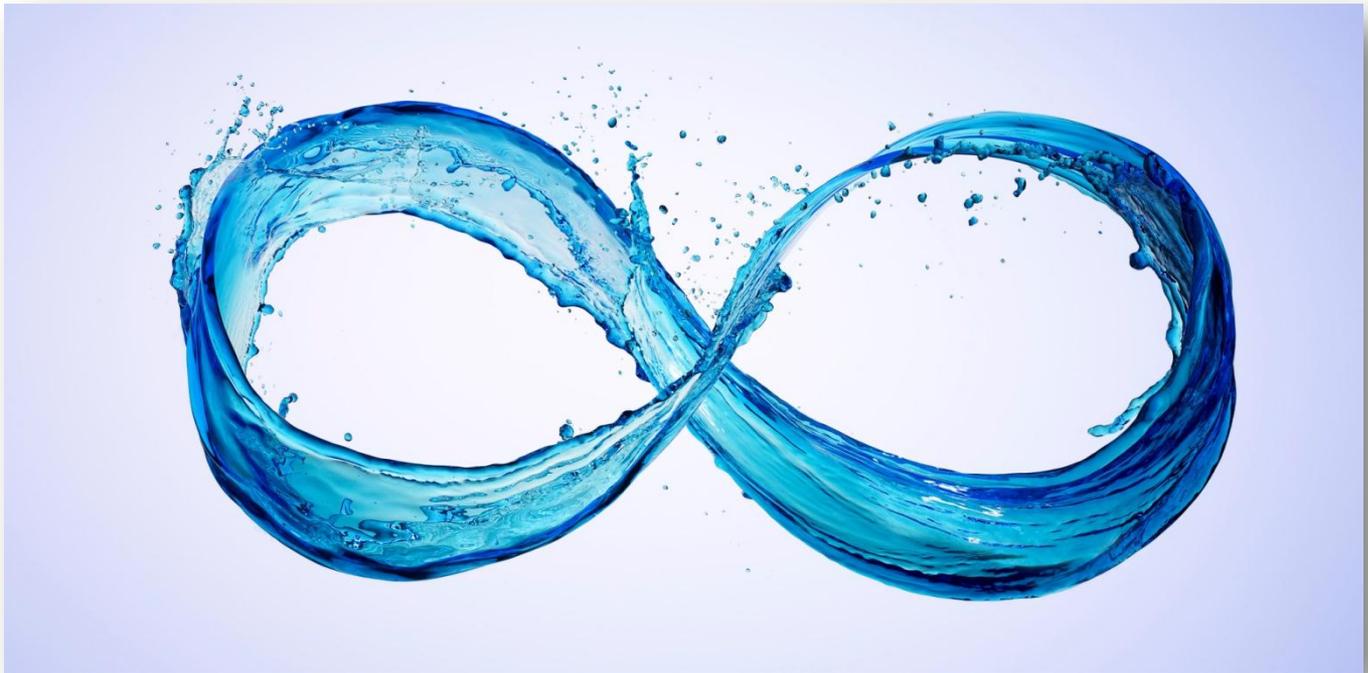
Результативность

(или *конечность*)

За конечное число шагов алгоритм либо должен приводить к решению задачи, останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

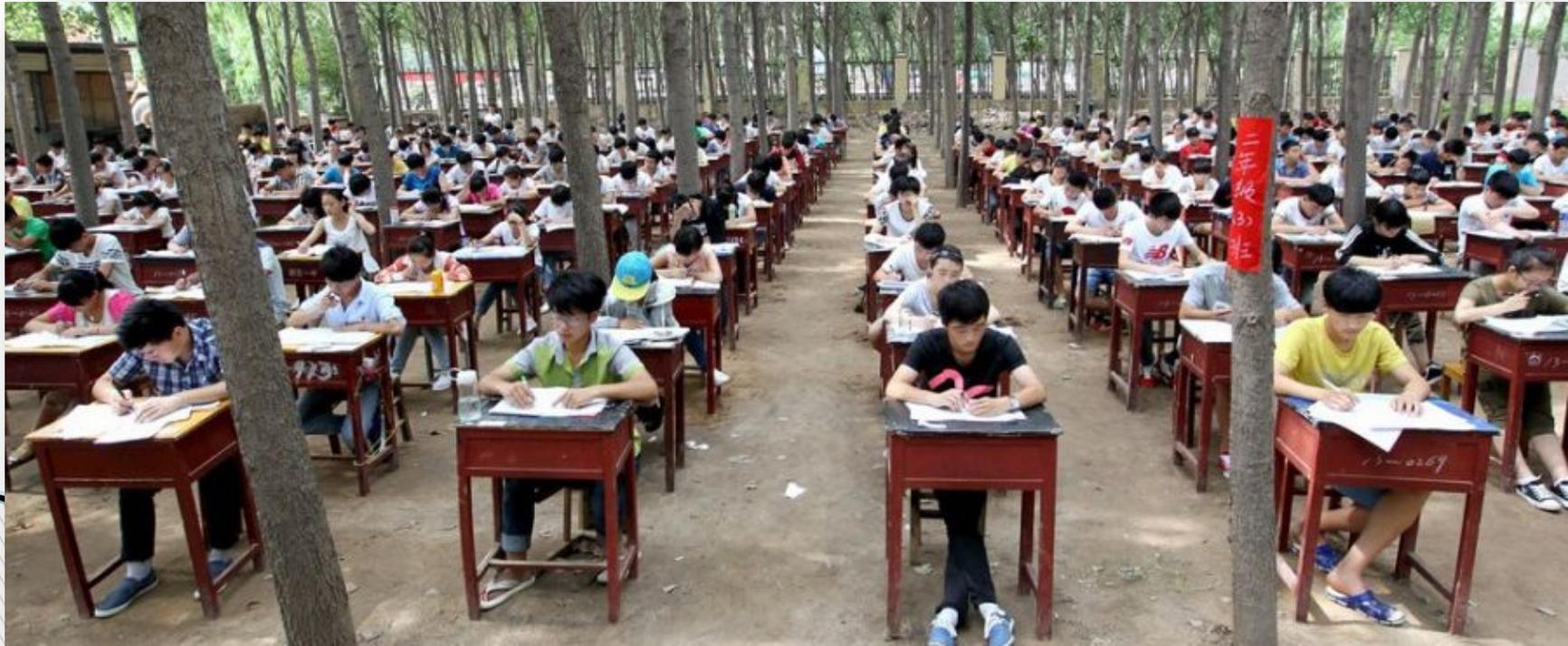


**Приведите пример
бесконечной команды**

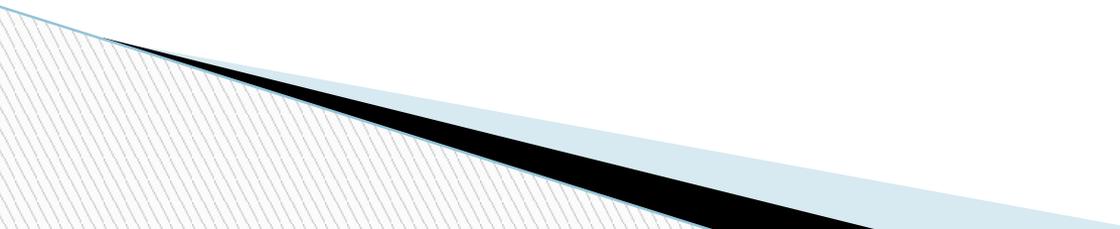


Массовость

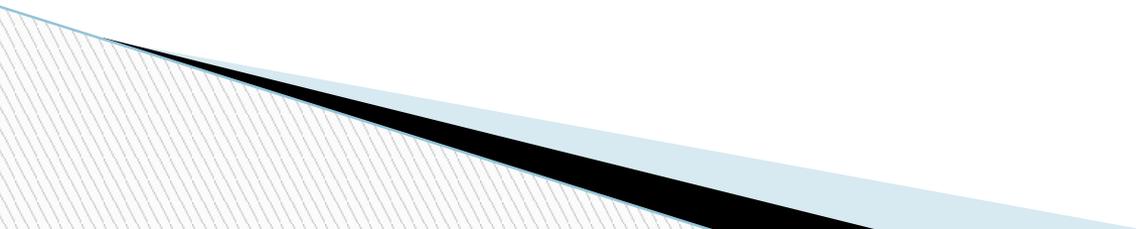
Алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными.



Порядок выполнения алгоритма

- действия выполняются в порядке их записи;
 - нельзя менять местами никакие два действия алгоритма;
 - нельзя не закончив одного действия переходить к следующему.
- 

4. Формы записи алгоритма



Словесная

Запись на естественном языке.

Рецепт «Омлет на сковородке»

1. Разбить яйца в миску, взболтать их вилкой, влить молоко, перемешать, посолить.
2. Подогреть сковороду с маслом.
3. Влить яйца и жарить минут 5 на медленном огне. Перемешать омлет, чтобы он поджарился равномерно.
4. Накрыть его крышкой и подержать минуты 2-3 - так он станет пышным.

Псевдокод

Полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.

```
Set Sum(X)=0
for каждого клиента x
    if клиент оплатил услуги вперед
    and ((Текущий месяц)>=(2 мес. после даты приобретения))
    and ((Текущий месяц)<=(14 мес. после даты приобретения))
    then Sum(X) = Sum(x) + (сумма, заплаченная
клиентом)/12
```

Программная

Тексты программ (листинги) на языках программирования.

```
Program Check_Group
use crystallographic_symmetry, only: Space_Group_Type, set_spacegroup
use reflections_utilities, only: Hkl_Absent
use Symmetry_Tables, only: spgr_info, Set_Spgr_Info

..... ! Read reflections, apply criterion of "goodness" for checking,
      ! set indices i1,i2 for search in space group tables ...
..... ! omitted for simplicity
call Set_Spgr_Info()
m=0
do_group: do i=i1,i2
  hms=adjust1(spgr_info(i)%HM)
  hall=spgr_info(i)%hall
  if( hms(1:1) /= "P" .and. .not. check_cent ) cycle do_group ! Skip centred groups
  call set_spacegroup(hall,Spacegroup,Force_Hall="y")
  do j=1,nhkl
    if(good(j) == 0) cycle !Skip reflections that are not good (overlap) for checking
    absent=Hkl_Absent(hkl(:,j), Spacegroup)
    if(absent .and. intensity(j) > threshold) cycle do_group !Group not allowed
  end do
  ! Passing here means that all reflections are allowed in the group -> Possible group!
  m=m+1
  num_group(m)=i
end do do_group
write(unit=*,fmt=*) " => LIST OF POSSIBLE SPACE GROUPS, a total of ",m," groups are possible"
write(unit=*,fmt=*) " -----"
write(unit=*,fmt=*) "      Number (IT)      Hermann-Mauguin Symbol      Hall Symbol"
write(unit=*,fmt=*) " -----"
do i=1,m
  j=num_group(i)
  hms=adjust1(spgr_info(j)%HM)
  hall=spgr_info(j)%hall
  numg=spgr_info(j)%N
  write(unit=*,fmt="(i10,4a)") numg,"          ",hms,"          ",hall
end do
.....
```

Программа -
это алгоритм
для
компьютера



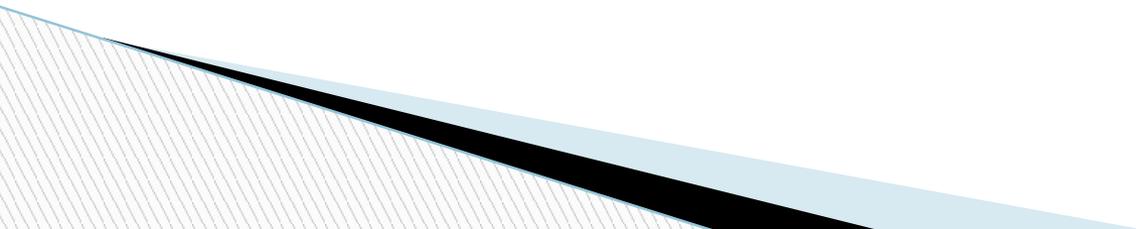
Графическая

Изображения из графических символов.

Алгоритм изображается *в виде* последовательности связанных между собой функциональных блоков.

Такое графическое представление называется *схемой алгоритма* или **блок-схемой**.

5. Блок-схемы, правила их построения



Символы блок-схем

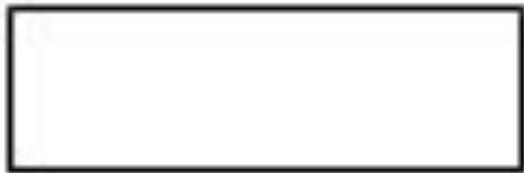


Начало / конец



Ввод/вывод

од



Действи

е



Логический блок
(условие)

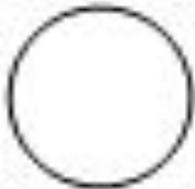
Символы блок-схем



Предопределенный процесс



(подпрограмма)
Блок



модификации
Соедините



ль
Межстраничны
й



соединитель
Линии
и

Правила построения блок-схем

- блок-схема выстраивается в одном направлении: сверху вниз и слева направо;
- все повороты соединительных линий выполняются под углом 90° .

6. Примеры алгоритмов

Алгоритм посадки дерева

Словесный алгоритм

1. Выкопать в земле ямку
2. Опустить в ямку саженец
3. Засыпать ямку с саженцем землёй
4. Полить саженец водой

Алгоритм в виде блок-схемы



Алгоритм выполнения домашнего задания

Словесный алгоритм

Если уроки выучены, то иди гулять, иначе учи уроки

Начало

Алгоритм в виде блок-схемы

Уроки выучены

Да

Нет

Иди гулять

Учи уроки

Конец



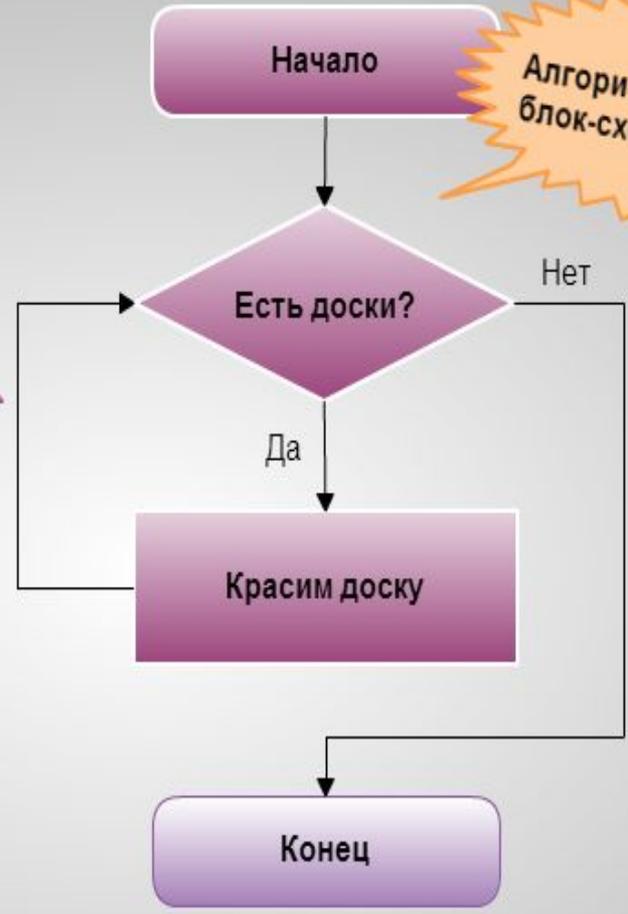
Алгоритм покраски досок

Словесный алгоритм

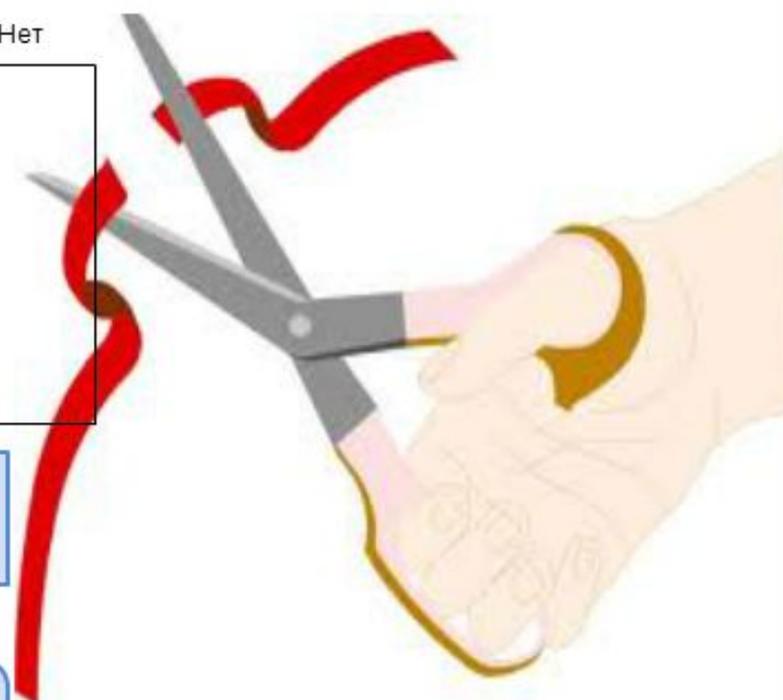
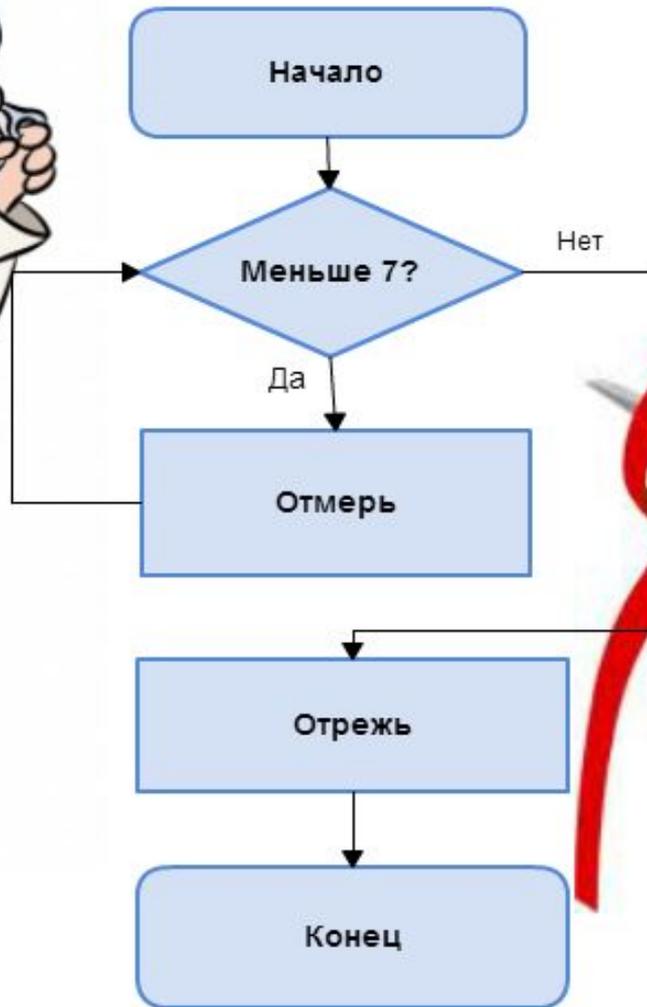
1. покрасим доску
2. если есть еще доска, то перейти к ней и выполнить действие 1
3. если доски закончились, то завершаем работу



Алгоритм в виде блок-схемы

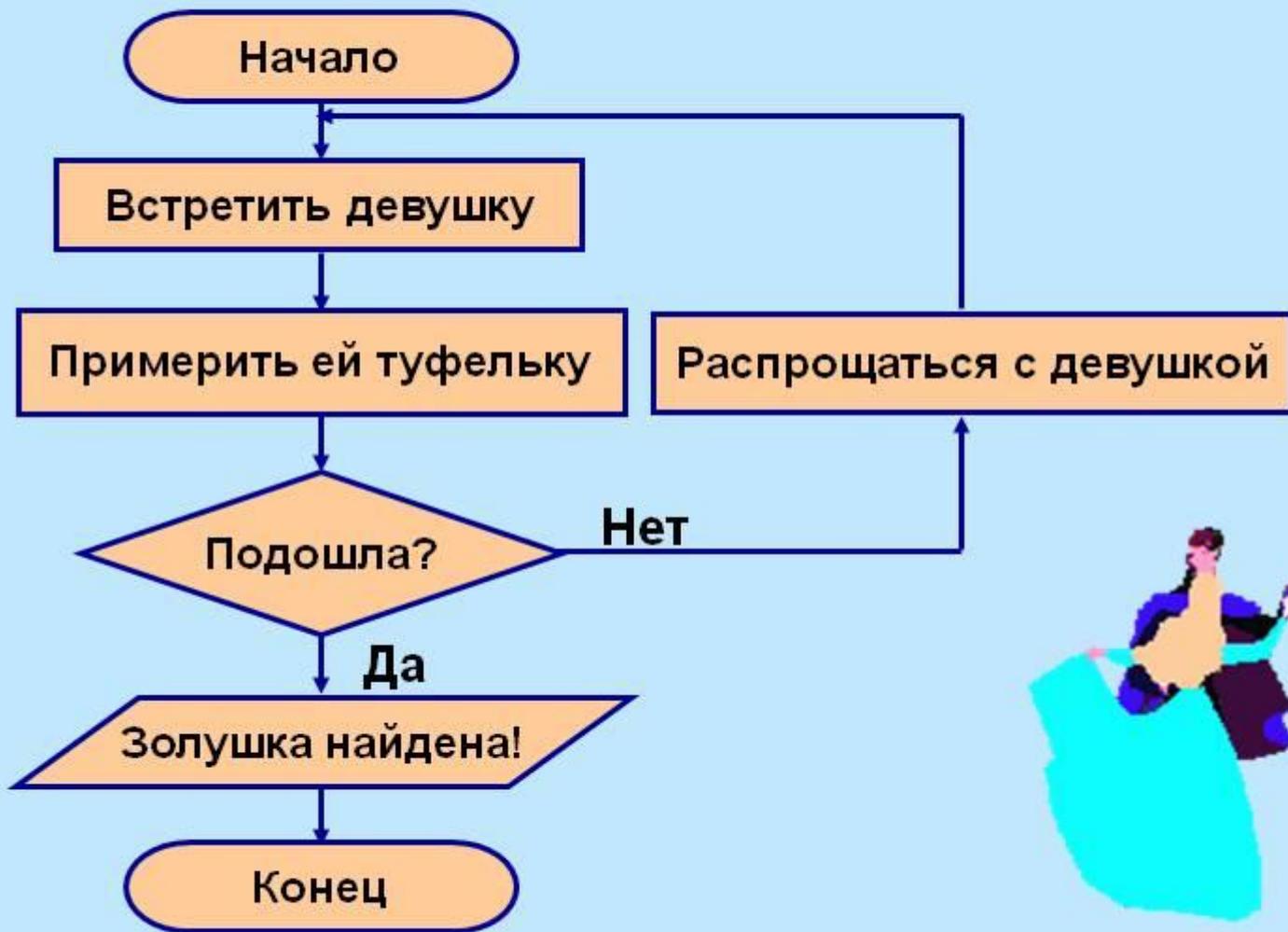


Семь раз отмерь, один раз отрежь



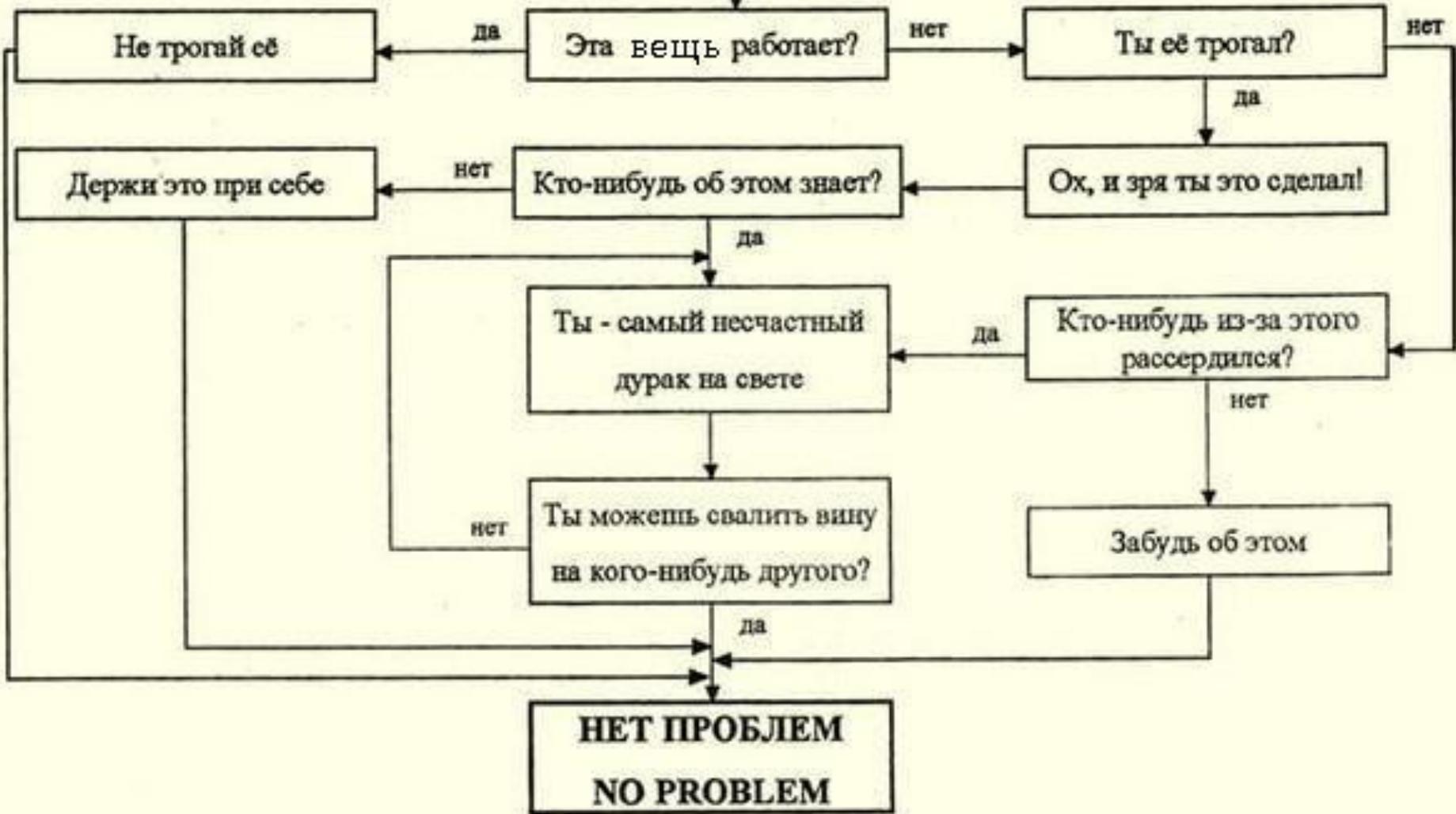


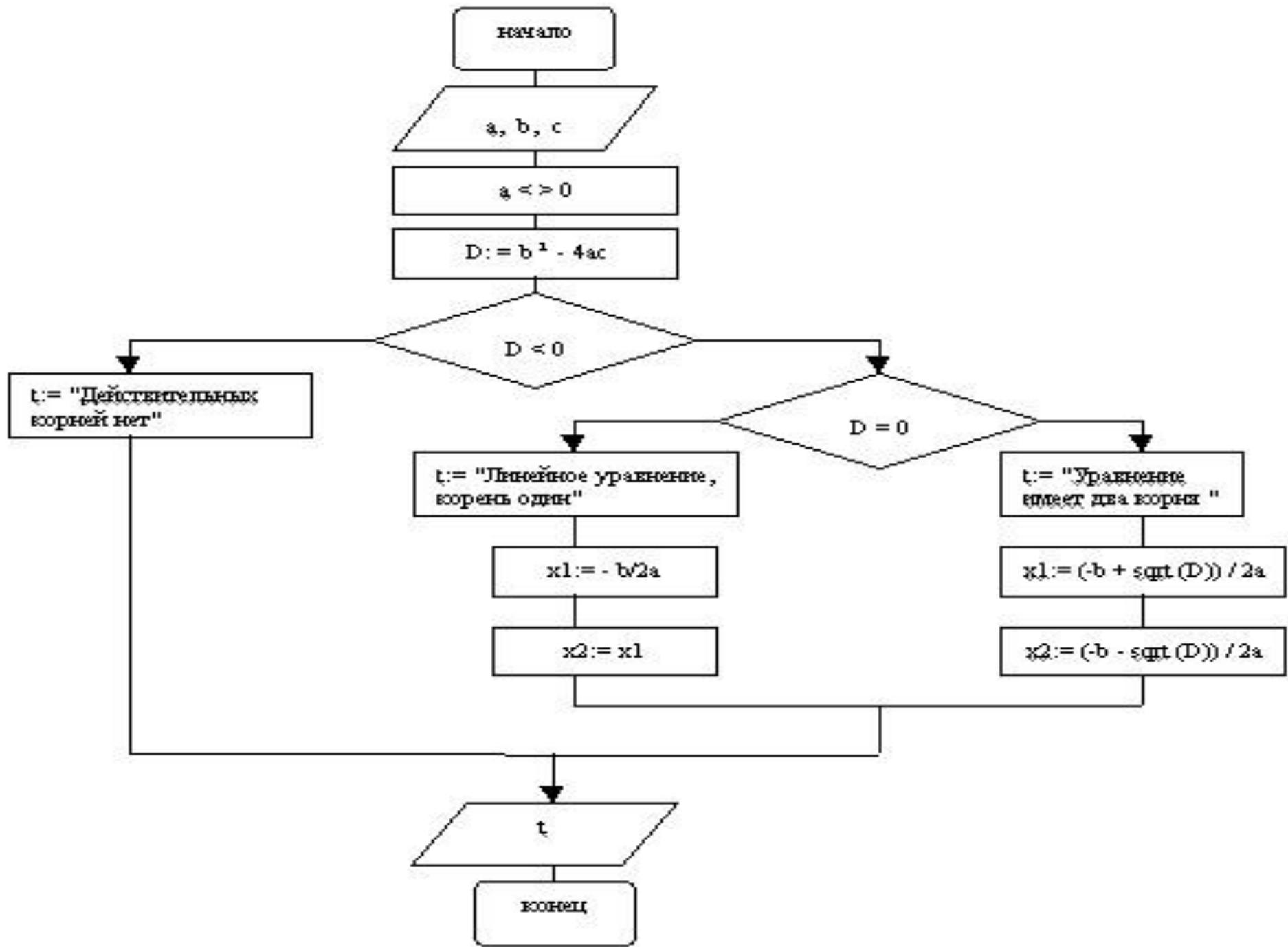
Алгоритм поиска Золушки



Решение проблем

ВХОД





Домашнее задание

- Происхождение слова «*алгоритм*»
 - Примеры алгоритмов из окружающей жизни
 - Преимущества и недостатки различных способов записи алгоритмов
- 