



Алгоритм

Баганова Елена Николаевна
ГБОУ СОШ №892 г. Москва
Учитель ИиИКТ

АЛГОРИТМ (план) - это строгая
последовательность действий,
приводящая
к желаемому результату.



Исполнитель алгоритма -
это человек или робот (ЭВМ),
умеющий выполнять некоторый,
вполне определённый набор действий,
называемый системой команд.

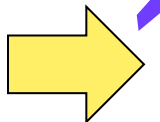


Робот (ЭВМ) выполняет алгоритм
формально, т.е. не вникая в
содержание поставленной задачи, а
только строго выполняя
последовательность действий.



Свойства алгоритма

- Раздельность
 - Массовость
 - Результативность
 - Определённость
 - Понятность



Раздельность (дискретность)

- алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов.



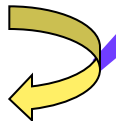
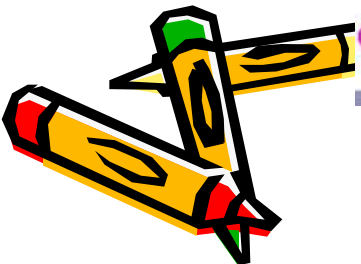
Массовость



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

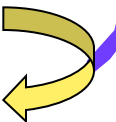


- 1) $30 + x = 67$
- 2) $x + 25 = 60$
- 3) $y + 110 = 200$
- 4) $x - 15 = 21$
- 5) $y - 37 = 123$
- 6) $x - 110 = 68$
- 7) $23 - x = 11$
- 8) $45 - y = 9$
- 9) $10 - y = 0$
- 10) $15 - y = 15$



Результативность (конечность)

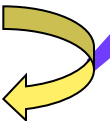
- алгоритм должен приводить к решению задачи за конечное число шагов.



Определённость



- каждое правило алгоритма должно быть чётким, однозначным и не оставлять места для домысливания .



Понятность

- алгоритм может быть выполнен только исполнителем, который *понимает* каждую команду алгоритма и может ее исполнить в строгом соответствии с ее назначением, т.е. каждая команда должна входить в систему команд этого исполнителя.



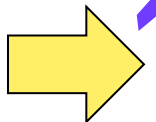
Формы представления алгоритма:



1. Словесная
(изложение на естественном языке)

2. Графическая
(схема, блок-схема)

3. Программная
(запись на каком-либо языке программирования)



Пример словесной формы:



Чтобы приготовить молочно - рисовую кашу в мультиварке:

1. В чашу мультиварки высыпает промытый рис.
2. Заливаем рис молоком и водой.
3. Добавляем сахар и сливочное масло.
4. Все перемешиваем.
5. Закрываем крышку мультиварки, выбираем режим «Молочная Каша» и устанавливаем время 35 минут.
6. После сигнала, кашку перемешаем.



Примеры графической формы :



Примеры программной формы:



```

Program Kornil;
Var
  a,b,c,d,x1,x2:Real;
Begin
  WriteLn('Задайте коэффициенты a, b, c: ');
  ReadLn(a,b,c);
  d:=b*b-4*a*c;
  If d<0 Then WriteLn('Действительных корней нет.')
  Else
    Begin
      x1:=(-b-Sqrt(d))/(2*a);
      x2:=(-b+Sqrt(d))/(2*a);
      WriteLn('Корни уравнения: x1=',x1,' x2=',x2);
    End;
End.
    
```

Пример программы на языке C#

```

using System;
namespace Example
{
  class Program
  {
    static void Main()
    {
      Console.WriteLine("Hello,
World!");
      Console.ReadKey(true);
    }
  }
}
    
```



$$\begin{aligned}
 a &:= \begin{pmatrix} 145 \\ 210 \\ 160 \end{pmatrix} & b &:= \begin{pmatrix} 237 \\ 278 \end{pmatrix} \\
 \sum a &= 515 & \sum b &= 515 \\
 M &:= \text{rows}(a) & N &:= \text{rows}(b) \\
 M &= 3 & N &= 2 \\
 c &:= \begin{pmatrix} 11.5 & 7 & 12 \\ 6.2 & 10 & 9.0 \end{pmatrix} \\
 f(x) &:= \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} c_{i,j} \cdot x_{i,j} \\
 x_{N-1,M-1} &:= 0 \\
 \text{Given} \\
 x_{0,0} + x_{0,1} + x_{0,2} &= b_0 & x_{0,0} + x_{1,0} &= a_0 & x_{0,0} \geq 0 & x_{1,0} \geq 0 \\
 x_{1,0} + x_{1,1} + x_{1,2} &= b_1 & x_{0,1} + x_{1,1} &= a_1 & x_{0,1} \geq 0 & x_{1,1} \geq 0 \\
 & & x_{0,2} + x_{1,2} &= a_2 & x_{0,2} \geq 0 & x_{1,2} \geq 0 \\
 \text{sol} &:= \text{Minimize}(f, x) \\
 \text{sol} &= \begin{pmatrix} 0 & 210 & 27 \\ 145 & 0 & 133 \end{pmatrix} & f(\text{sol}) &= 3.89 \times 10^3
 \end{aligned}$$



Закрепление знаний



1. Свойство алгоритма, которое определяет возможность завершения процесса. (результативность)
2. Объект, умеющий выполнять определенный набор действий. (исполнитель)
3. У исполнителя Вычислитель две команды, которым присвоены номера:
 1. **умножь на 3**
 2. **вычти 2**Первая из них увеличивает число на экране в 3 раза, вторая уменьшает его на 2.
Получите из числа 2 число 30, в ответе запишите **только** номера команд.

1 2 1 2 1



БЛОК - СХЕМА



Это графический способ описания алгоритма (плана) с помощью геометрических фигур, называемых **БЛОКАМИ**, каждая из которых отвечает за определённое действие.

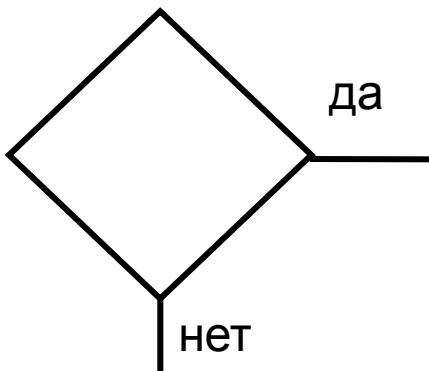
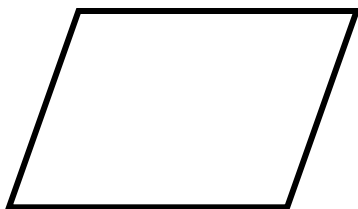
Блоки соединяются между собой линиями связи в виде —, если они рисуются сверху - вниз и слева - направо. В противном случае —▶.

Алгоритм записанный в виде последовательности блоков, называется **БЛОК - СХЕМОЙ**.



НАЧАЛО

КОНЕЦ



Блок
начала и конца
алгоритма

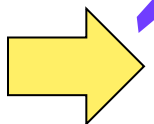
Вспомогательный блок
(блок ввода/ вывода)

Логический блок
(блок проверки условия)

Вычислительный блок
(процесс)

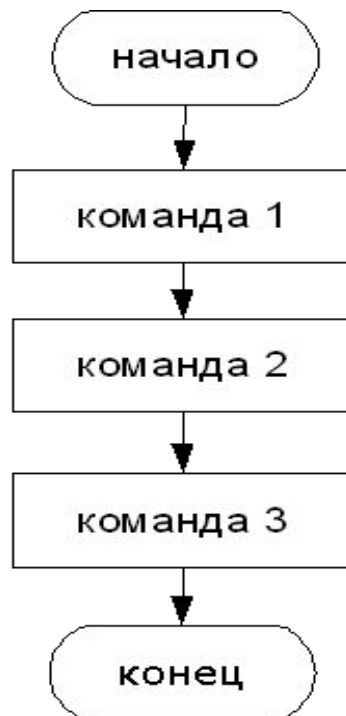
Виды алгоритмов

- Линейный алгоритм
(следование)
- Разветвляющийся алгоритм
(ветвление)
- Циклический алгоритм
(повторение)



Следование

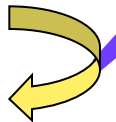
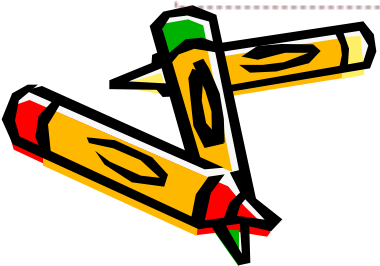
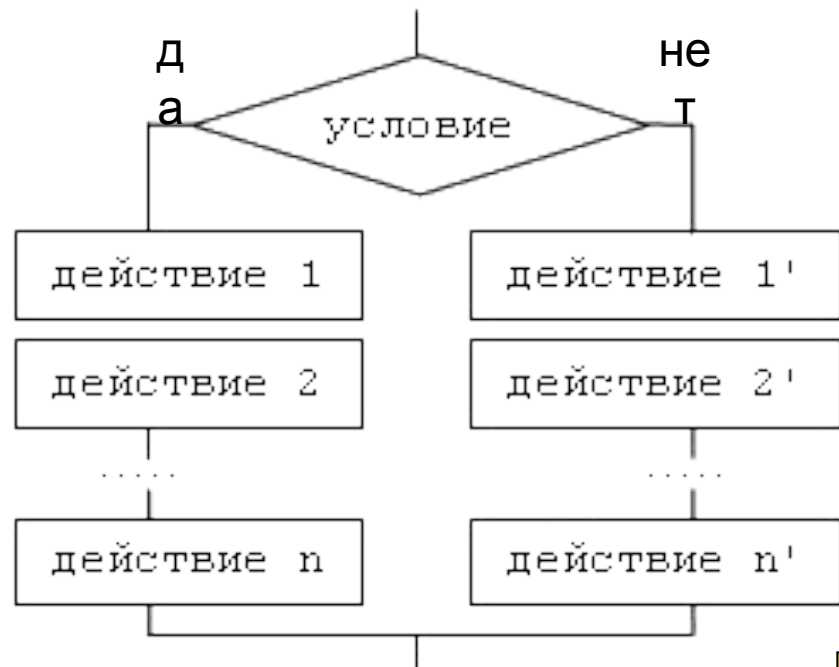
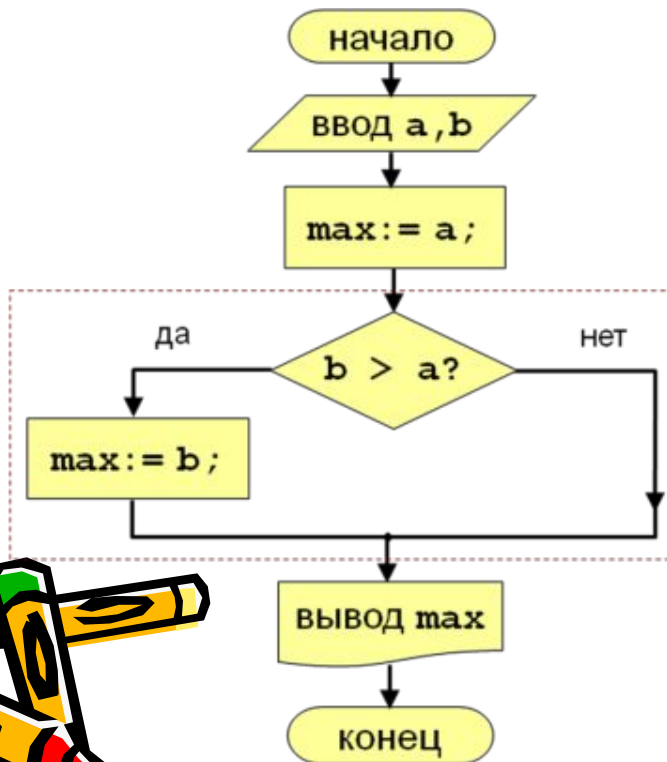
образуется из последовательности действий, следующих одно за другим.



Ветвление



обеспечивает в зависимости от результата проверки условия (ДА или НЕТ) выбор одного из путей работы алгоритма.

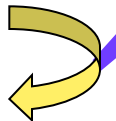
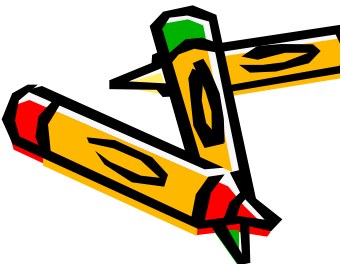
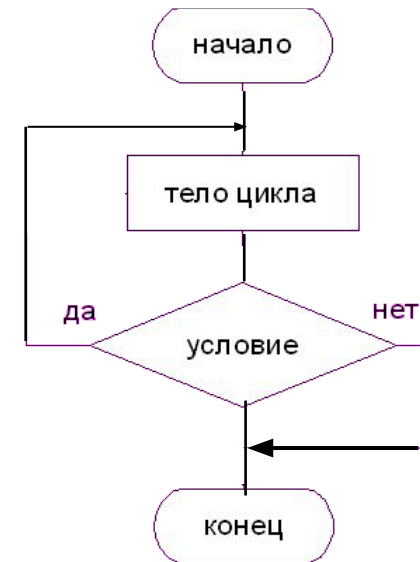
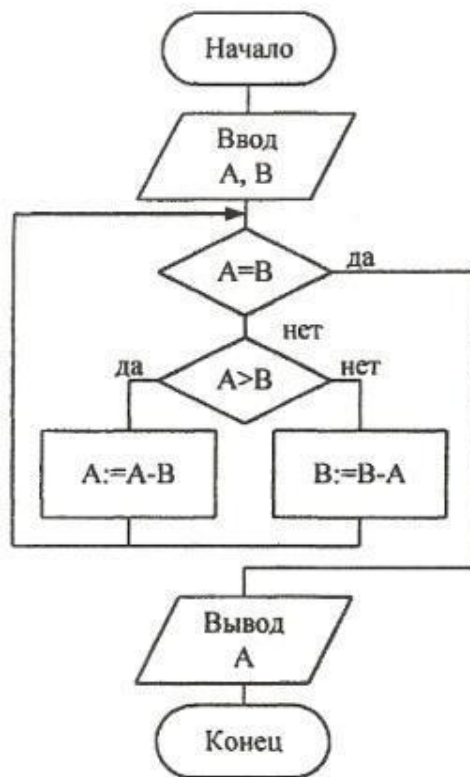
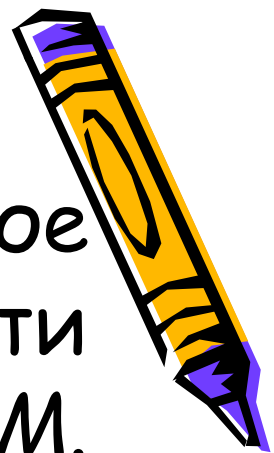


Повторение

обеспечивает

многократное

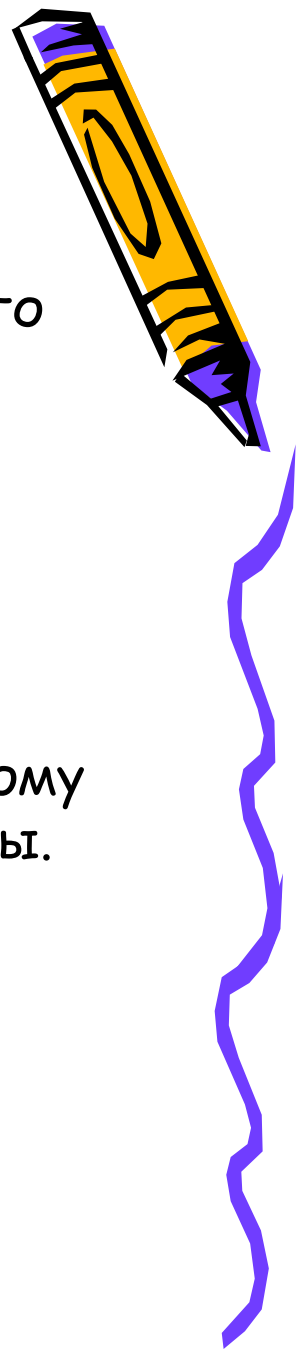
выполнение некоторой совокупности действий, которая называется **ЦИКЛОМ**.



Закрепление знаний

1. Создайте алгоритм нахождения площади прямоугольного треугольника и оформите его в виде блок-схемы.

2. Создайте алгоритм перехода дороги по пешеходному переходу ЗЕБРА и оформите его в виде блок-схемы.



Используемые материалы

1. Семакин И.Г., Хеннер Е.К. Задачник-практикум по информатике. М.: Бином, 2008 г.
2. Шауцукова Л.З. Информатика. Учеб. пособие для 10 - 11 классов: Просвещение, 2000 г.

