

# Костанайский Государственный Университет им. Ахмета Байтурсынова

УНИВЕРСИТЕТ



**Автор презентации:** ст. преподаватель кафедры ИиМ  
Ермагамбетова Гульмира Нурпановна

**Тема:**  
○○○○○○○○○○

# Концепции и свойства алгоритмов, реализация алгоритмов

**Цель:**  
○○○○○○●●■

**Познакомить с понятием алгоритм**  
**и его свойствами, изучить**  
**основные способы записи**  
**алгоритмов**



# План Лекции:

1. Алгоритмы и свойства алгоритмов

2. Блок-схемы как графическая реализация алгоритмов

# Задачи Лекции:

1. Рассмотреть основные понятия алгоритма

2. Показать основные виды и свойства алгоритма.

3. Дать классификацию формам представления алгоритма

4. Рассмотреть блок-схему алгоритма



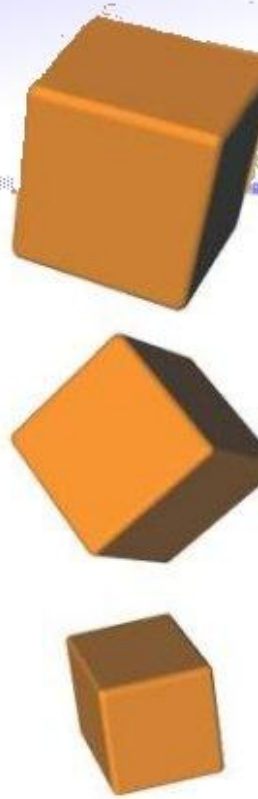
# 1. Алгоритмы и свойства алгоритмов

Слово «Алгоритм» происходит от algorithmi - латинского написания имени аль-Хорезми, под которым в средневековой Европе знали величайшего математика из Хорезма Мухаммеда бен Мусу.



**783-850 гг.**

В своей книге «**Об индийском счете**» он сформулировал правила записи натуральных чисел с помощью арабских цифр и правила действий над ними столбиком.





**Алгоритм** — это метод (способ) решения задачи, записанный по определенным правилам, обеспечивающим однозначность его понимания и механического исполнения при всех значениях исходных данных.

## Алгоритм

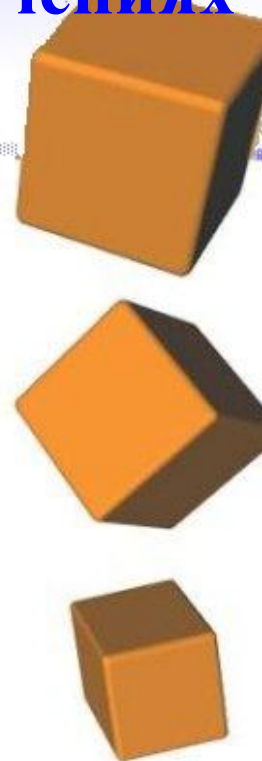


точное предписание, определяющее последовательность действий, обеспечивающую получение требуемого результата из исходных данных.



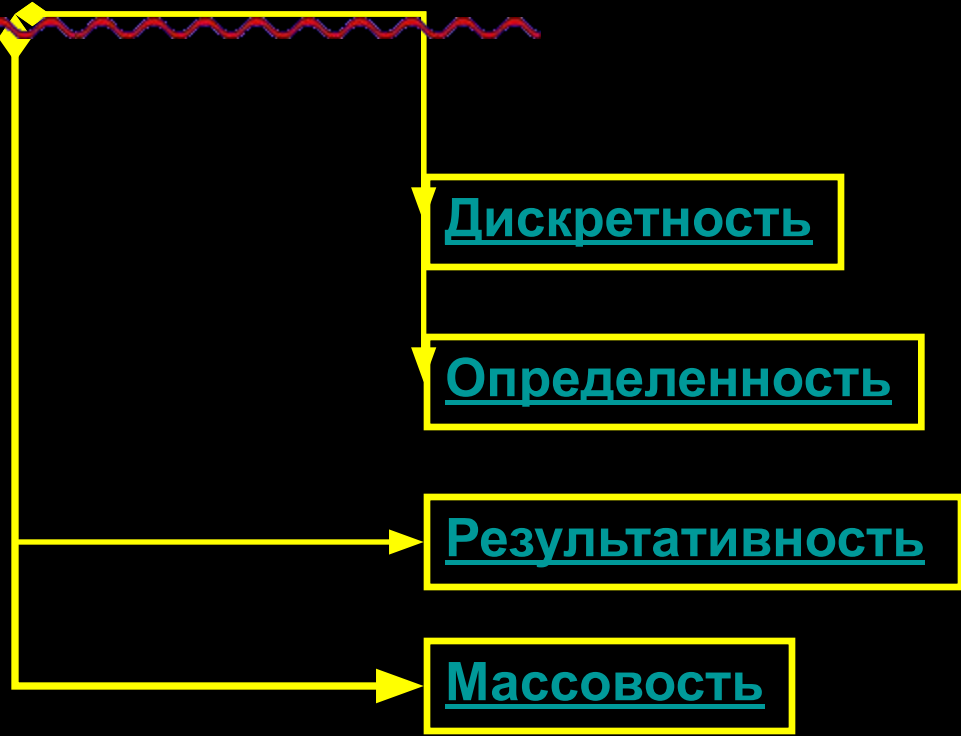
## Исполнитель алгоритма

человек или устройство (в частности, процессор компьютера), умеющий выполнять определённый набор действий. Исполнитель является средством реализации алгоритма.





# Свойства алгоритма





# Дискретность

Свойство алгоритма, означающее, что процесс решения задачи, определяемый алгоритмом, расчленен на отдельные элементарные действия и соответственно алгоритм представляет последовательность указаний, команд, определяющих порядок выполнения шагов процесса.





# Определенность



Это свойство означает, что каждая команда алгоритма (предписание, выдаваемое на каждом шаге) должна быть понятна исполнителю, не оставлять места для ее неоднозначного толкования и неопределенного исполнения. Описание алгоритма должно быть таким, чтобы его мог выполнить любой грамотный пользователь.



# Результативность

Свойство алгоритма, состоящее в том, что он всегда приводит к результату через конечное, возможно, очень большое число шагов.







# Массовость

Это свойство заключается в том, что каждый алгоритм, разработанный для решения некоторой задачи, должен быть применим для решения задач этого типа при всех допустимых значениях исходных данных.



# Правила построения алгоритма

**Первое правило**

**Второе правило**

**Третье правило**

**Четвертое правило**

**Пятое правило**



## Первое правило

При построении алгоритма прежде всего необходимо задать множество объектов, с которыми будет работать алгоритм. Формализованное (закодированное) представление этих объектов носит название данных.

Алгоритм приступает к работе с некоторым набором данных, которые называются **входными**, и в результате своей работы выдает данные, которые называются **выходными**.

**Алгоритм преобразует входные данные в выходные.**



## Второе правило

Для работы алгоритма требуется память.

В памяти размещаются входные данные, с которыми алгоритм начинает работать, промежуточные данные и выходные данные, которые являются результатом работы алгоритма.

Память является дискретной, т.е. состоящей из отдельных ячеек. Поименованная ячейка памяти носит название переменной.

В теории алгоритмов размеры памяти не ограничиваются, т. е. считается, что можно предоставить алгоритму любой необходимый для работы объем памяти.





## Третье правило

Дискретность.

**Алгоритм строится из отдельных шагов  
(действий, операций, команд).**

**Множество шагов, из которых составлен алгоритм, конечно**



## Четвертое правило

Детерменированность.

**После каждого шага необходимо указывать, какой шаг выполняется следующим, либо давать команду остановки.**



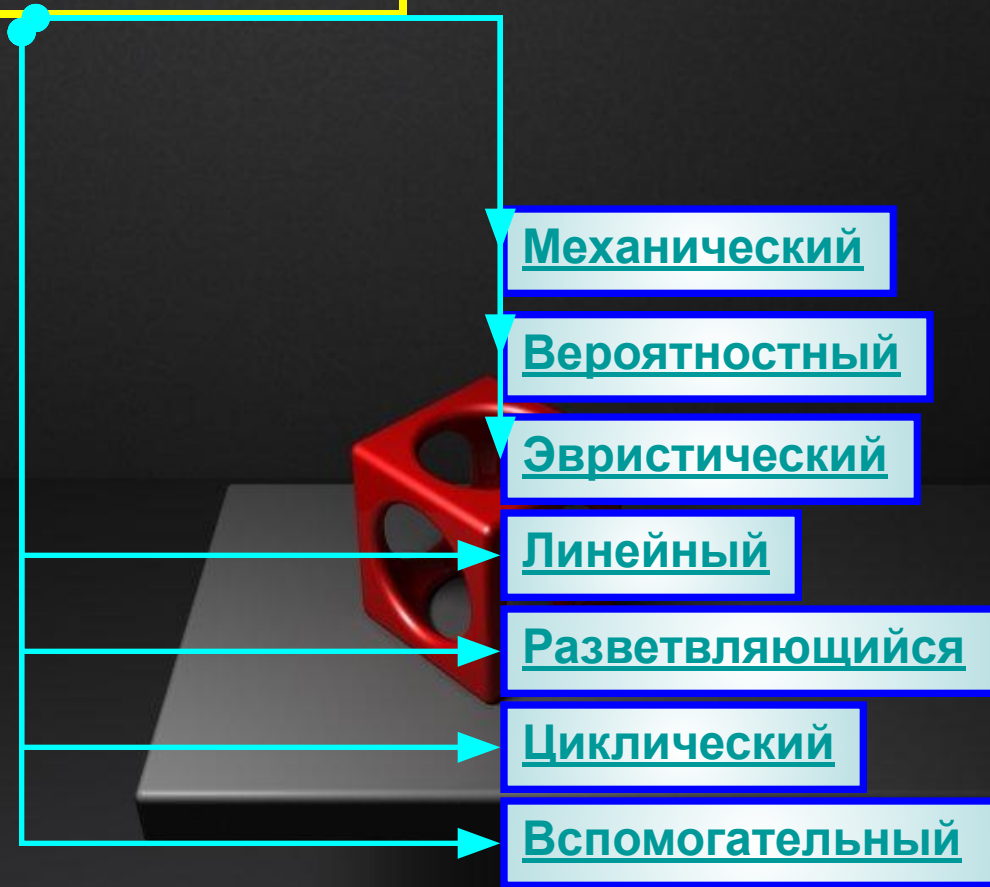
## Пятое правило

Сходимость (результативность).

**Алгоритм должен завершать работу после конечного числа шагов. При этом необходимо указать, что считать результатом работы алгоритма.**



# Виды алгоритмов



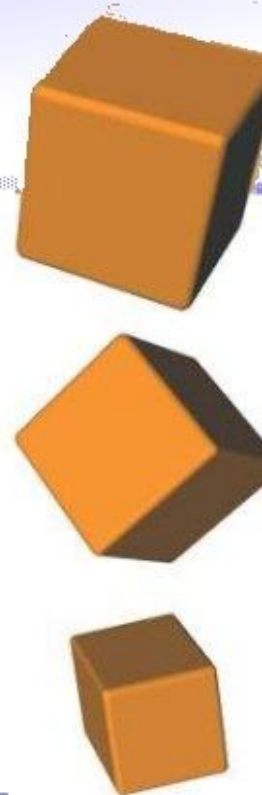
Виды алгоритмов как логико-математических средств отражают указанные компоненты человеческой деятельности и тенденции, а сами алгоритмы в зависимости от цели, начальных условий задачи, путей ее решения, определения действий исполнителя подразделяются на виды.





## Механический

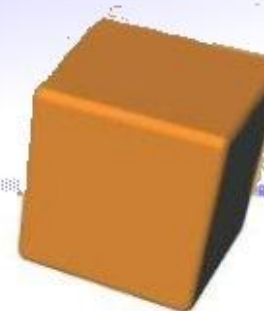
Механический алгоритм задает определенные действия, обозначая их в единственной и достоверной последовательности, обеспечивая тем самым однозначный требуемый или искомый результат, если выполняются те условия процесса, задачи, для которых разработан алгоритм.





## Вероятностный

**Вероятностный (стохастический) алгоритм** дает программе решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.



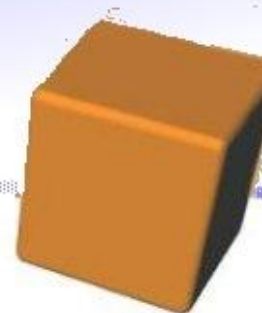


## Эвристический

**Эвристический алгоритм** (от греческого слова “эврика”) – это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя.



К эвристическим алгоритмам относят, например, инструкции и предписания. В этих алгоритмах используются универсальные логические процедуры и способы принятия решений, основанные на аналогиях, ассоциациях и прошлом опыте решения схожих задач.тол

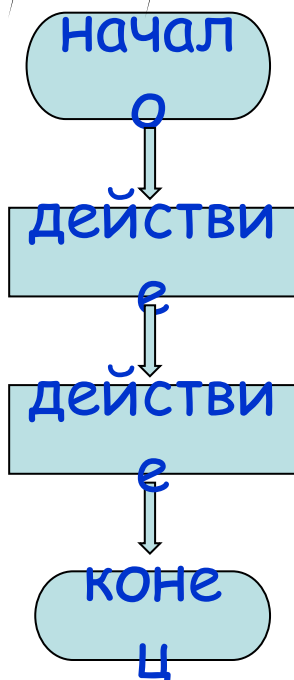




## Линейный



### Линейный



**Линейный алгоритм** – набор команд (указаний), выполняемых последовательно во времени друг за другом.

*Линейные алгоритмы* состоят из команд, которые выполняются последовательно.

Например, при “ решении задачи” сварить борщ - все действия выполняются одно за другим.

Они как бы выстраиваются в одну линию.

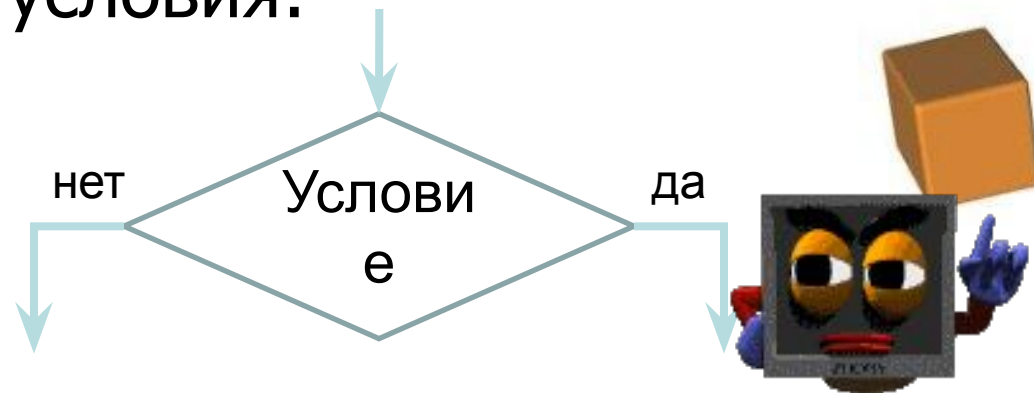


# Разветвляющийся

**Разветвляющимся** называется такой алгоритм, в котором выбирается один из нескольких возможных путей (вариантов) вычислительного процесса. Каждый подобный путь называется **ветвью алгоритма**.

Признаком разветвляющегося алгоритма является наличие **блока (операций) проверки условия**.

Блок проверки условия:

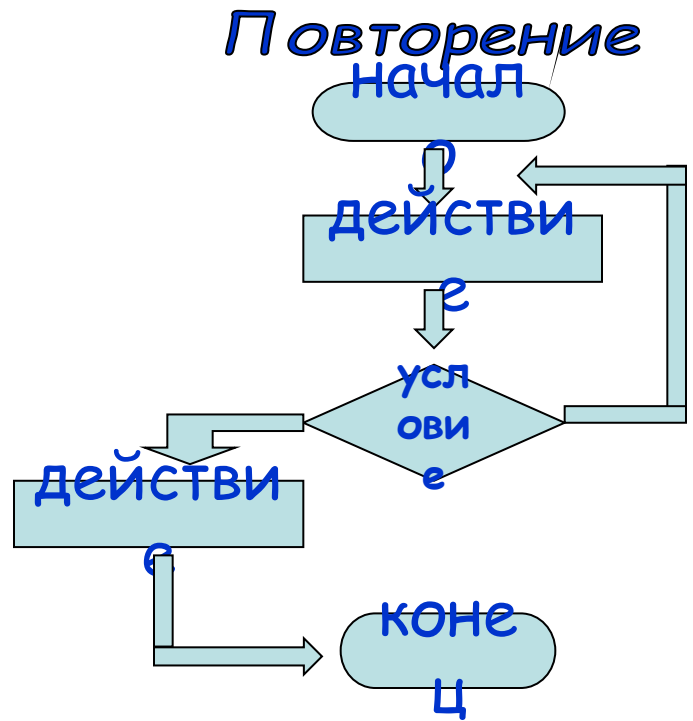


# Циклический

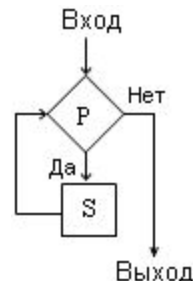
Циклический алгоритм предполагает наличие действий, выполняющихся многократно.

Например, алгоритм рыбной ловли – отдельные действия в алгоритме будут повторяться.

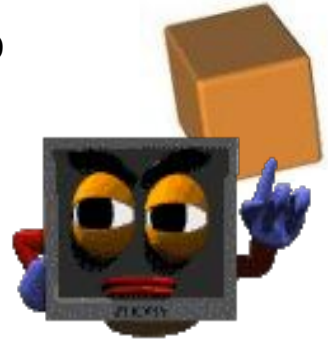
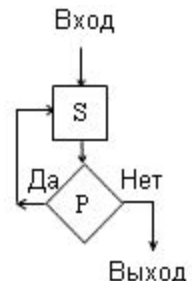
Циклическим алгоритмом называется алгоритм, предусматривающий многократное повторение одного и того же действия над новыми данными.



Цикл - пока



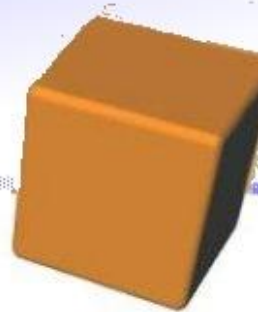
Цикл - до



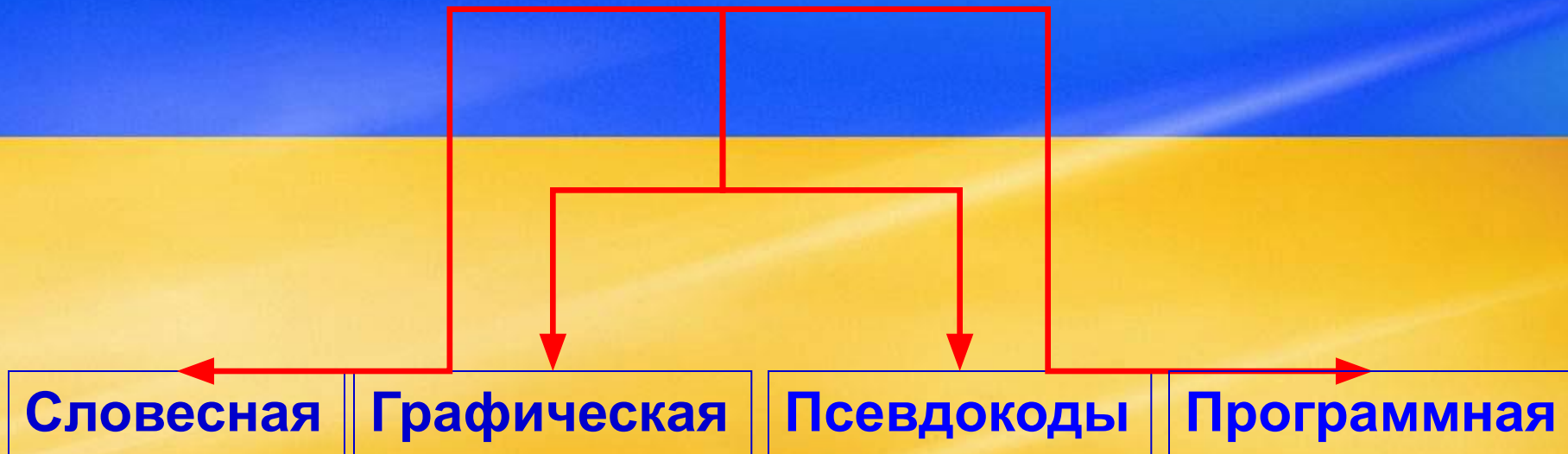


**Вспомогательный (подчиненный) алгоритм (процедура) – алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи.**

**В некоторых случаях при наличии одинаковых последовательностей указаний (команд) для различных данных с целью сокращения записи также выделяют вспомогательный алгоритм.**



# Формы представления алгоритмов



1. Налить в чайник воду.

2. Зажечь спичку.

3. Открыть кран газовой горелки.

4. Поднести спичку к горелке.

5. Поставить чайник на плиту.

6. Ждать, пока вода закипит.

7. Выключить газ.

## Блок-схема

начало

конец

Выполнение действия

```
unit UPersonal;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes,  
Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, ExtCtrls, StdCtrls;
```

```
type
```

```
TPersonal = class(TForm)
```

```
pers_edit.Edit2.Text="";
```

```
pers_edit.Edit3.Text="";
```

```
pers_edit.Edit4.Text="";
```

```
end;
```

```
procedure TPersonal.Button2Click(Sender: TObject);
```

```
begin
```

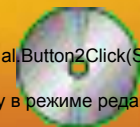
```
{открываем форму в режиме редактирования
```

```
записи }
```

```
pr:='Edit';
```

```
pers_edit.show;
```

```
end;
```

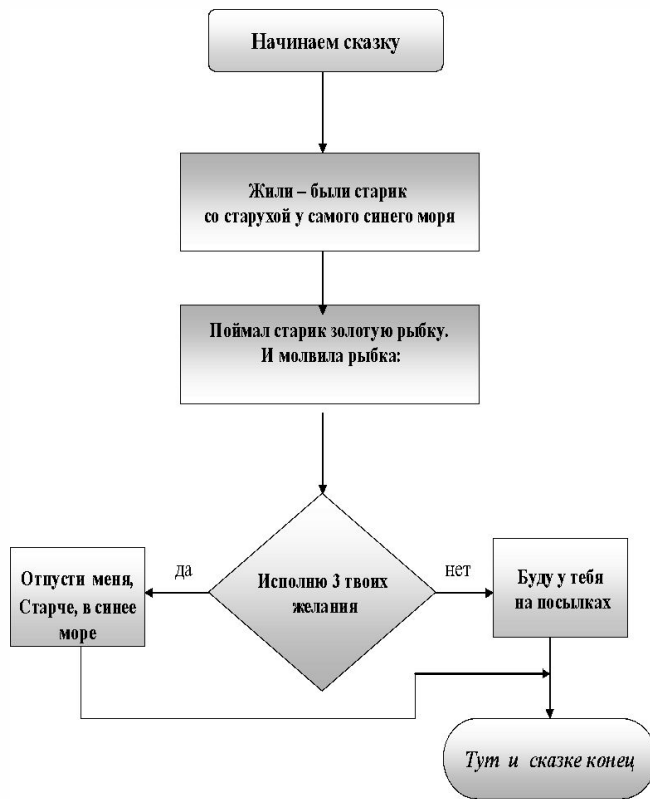






**2. Блок-схемы как графическая  
реализация алгоритмов**

**Блок-схемой** называется графическое изображение логической структуры алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций.



## Графическое изображение и название символов

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		<u>Вычислительное действие или последовательность действий</u>
Решение		<u>Проверка условий</u>
Модификация		<u>Начало цикла</u>
Предопределенный процесс		<u>Вычисления по подпрограмме, стандартной подпрограмме</u>
Ввод-вывод		<u>Ввод-вывод в общем виде</u>
Пуск-останов		<u>Начало, конец алгоритма, вход и выход в подпрограмму</u>
Документ		<u>Вывод результатов на печать</u>

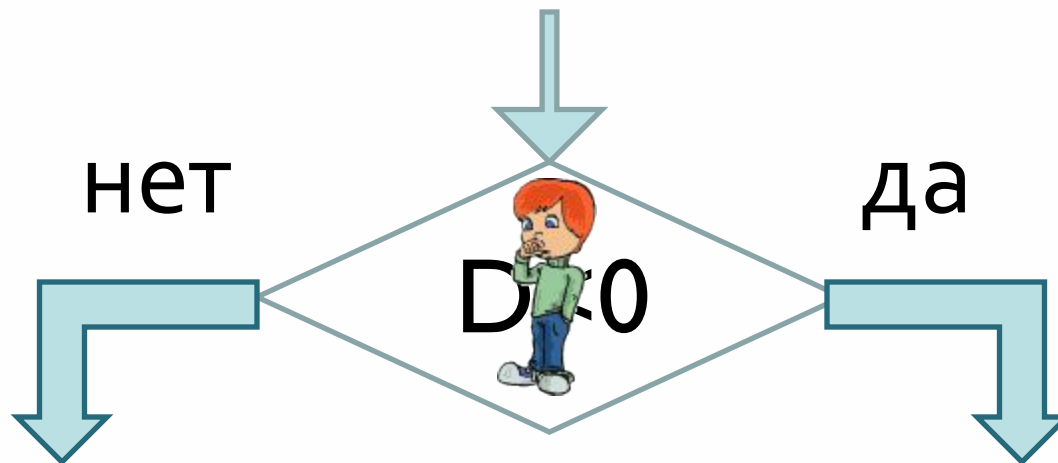






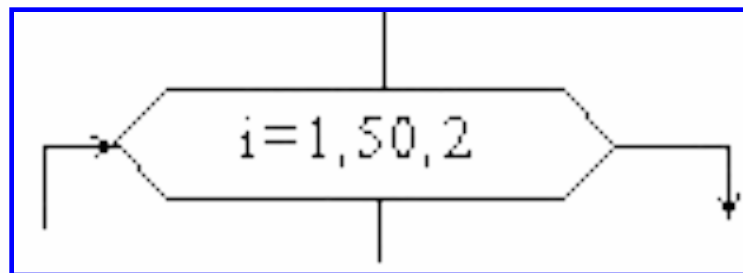
Блок "решение" используется для обозначения переходов управления по условию.

В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

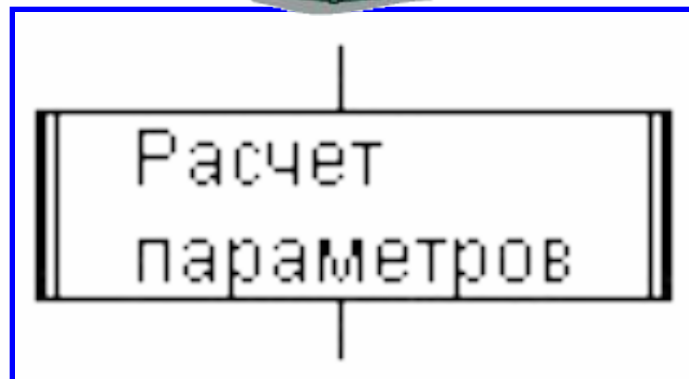


Блок "модификация" используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование).

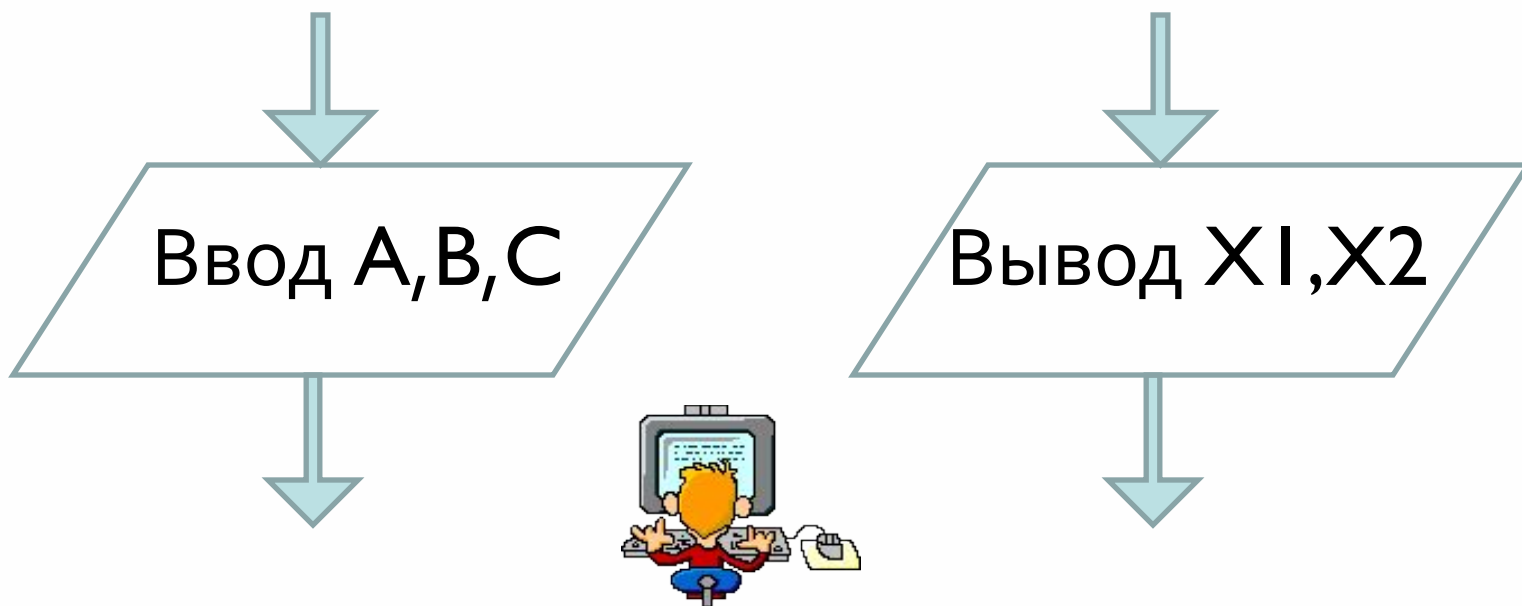
Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.



Блок "**предопределенный процесс**" используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.



Операции **Ввод** и **Вывод** изображаются параллелограммом:



При реализации этого блока в программе необходимо ввести исходные данные или немедленно вывести результат работы.





Блок «**Пуск-остановка**» используется для обозначения переходов управления по условию.



Начало процесса решения задачи обозначается блоком **Начало**.

Завершение процесса решения задачи обозначается блоком **Останов**

# Литература



1. Н.Вирт. Алгоритмы и структуры данных: Пер. с англ. Д.Б.Подшивалова. – М.: Мир, 1989. – 360 с., ил.
2. Гудман С., Хидетниемеи С. Введение в разработку и анализ алгоритмов. Пер. с англ. под ред. В.В.Мартынюка. – М.: Мир, 1981, 368 с.
3. Долинский М.С. Алгоритмизация и программирование на Turbo Pascal: от простых до олимпиадных задач. Учебное пособие. СПб.: Питер, 2005. 237 с.: ил.
4. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы: Построение и анализ / Пер. с англ. под ред. А.Шеня. – М.: МЦНМО: БИНОМ. Лаборатория знаний, 2004. – 2-е изд., стереотип. – 960 с.: 263 ил.
5. Дж.Макконнелл. Основы современных алгоритмов. 2-е дополненное издание. Пер. с англ. под ред. С.К.Ландо, Дополнение М.В.Ульянова. – Москва, Техносфера, 2004 – 368 с.
6. В.С.Новичков, Н.И.Парфилова, А.Н.Пылькин. Алгоритмизация и программирование на Турбо Паскале. Учебное пособие. М.: Горячая линия – Телеком, 2005. 438 с.: ил.
7. Окулов С.М. Основы программирования. М.: БИНОМ. Лаборатория знаний, 2004. 424 с.: ил.
8. Окулов С.М. Программирование в алгоритмах. М.: БИНОМ. Лаборатория знаний, 2004. 341 с.: ил.
9. Ставровский А.Б. Первые шаги в программировании. Самоучитель: – М.: Издательский дом “Вильямс”, 2003. 368 с.: ил.





## Контрольные вопросы:

Что такое алгоритм?

Кто является исполнителем алгоритма?

Перечислите основные свойства алгоритма?

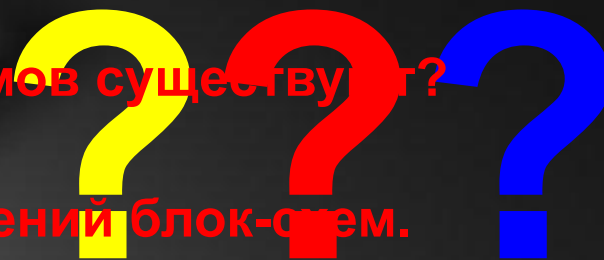
В чем заключается первое правило построения алгоритма?

Какие бывают виды алгоритмов?

Какие формы представления алгоритмов существуют?

Что такое блок-схема?

Классификация графических изображений блок-схем.





**Спасибо за Внимание!**