

***Алгоритм. Свойства  
алгоритма. Способы записи  
алгоритмов.***

Борисов В.А.

КАСК – филиал ФГБОУ ВПО РАНХ и ГС

Красноармейск 2011 г.



# Цель

---

- Научиться решать задачи на ЭВМ, т.е. использовать для решения задач ЭВМ, которая эти задачи решать не умеет.



# Этапы решения задачи

---

- выбирают способ решения задачи и изучают его во всех подробностях;
- сообщают исполнителю выбранный метод в абсолютно понятном для него виде;
- исполнитель решает задачу строго в соответствии с методом.



# Первый этап

---

- Не вызывает затруднений, так как для большинства встречающихся задач метод решения либо известен из практики, либо подсказывается здравым смыслом, либо описан в литературе.



## Второй этап

---

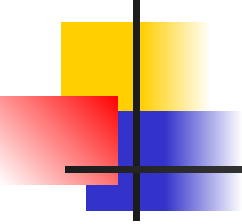
- Значительно сложнее, так как если способ решения задачи описан произвольно, нет гарантии, что он будет верно понят исполнителем.

# Правила описания метода



---

- выделить величины, являющиеся исходными для задачи;
- разбить процесс решения задачи на такие этапы, которые известны исполнителю и которые он может выполнить однозначно без всяких пояснений;
- указать порядок выполнения этапов;
- указать признак окончания процесса решения задачи;
- указать во всех случаях, что является результатом решения задачи.

- 
- 
- Описание метода, выполненное в соответствии с этими правилами, называется алгоритмом решения задачи.



# Алгоритм

---

- Метод решения задачи, записанный по определенным правилам, обеспечивающим однозначность его понимания и механического исполнения при всех значениях исходных данных.



# Толковый словарь по информатике



---

- «*Алгоритм* — точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату».



# Алгоритм заварки чая

---

- Подготовить исходные величины — чай, воду, чайник, стакан, ложку.
- Налить в чайник воду.
- Довести воду до кипения и снять с огня.
- Всыпать в чайник чай.
- Довести воду до кипения (но не кипятить), снять с огня.
- Чай готов. Процесс прекратить.

# Основные свойства алгоритма



---

- дискретность;
- определенность;
- результативность;
- массовость.

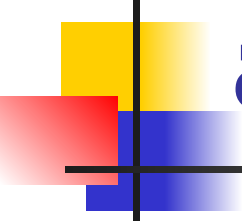


# Дискретность алгоритма

---

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

# Определенность алгоритма



---

- Каждая команда алгоритма должна быть понятна исполнителю, не оставлять места для ее неоднозначного толкования и неопределенного исполнения.

# Результативность алгоритма



---

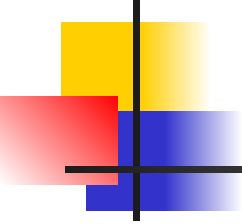
- Свойство алгоритма, состоящее в том, что он всегда приводит к результату через конечное, возможно, очень большое число шагов.



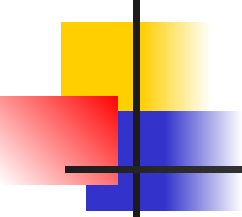
# Массовость алгоритма

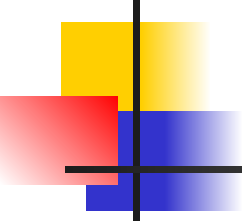
---

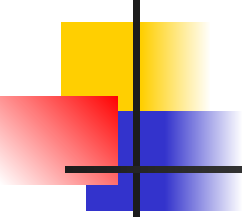
- Заключается в том, что каждый алгоритм, разработанный для решения некоторой задачи, должен быть применим для решения задач этого типа при всех допустимых значениях исходных данных.

- 
- 
- Возможность использования ЭВМ вместо человека объясняется соответствием свойств алгоритма и ЭВМ: алгоритм допускает механическое выполнение его для решения задачи, а ЭВМ может механически, не вникая, выполнять операции в заданном порядке.



- 
- 
- Отличие указанного процесса решения задачи при использовании ЭВМ в том, что, составляя алгоритм, мы разбиваем процесс решения задачи на такие операции, которые в состоянии выполнить ЭВМ.

- 
- 
- Другое отличие в том, что составленный алгоритм решения задачи следует перевести на язык, понятный ЭВМ.

- 
- 
- Запись алгоритма на языке программирования называется *программой*.
  - Процесс перевода алгоритма на указанный язык — *программированием*.



# Возможности ЭВМ

---

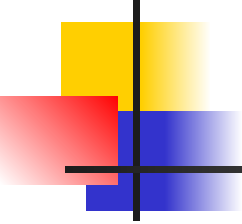
- При составлении алгоритма процесс решения задачи разбивают на этапы, ориентируясь на исполнителя, на операции, известные ему.
- Если нужно составить программу для ЭВМ, то в алгоритме должны предусматриваться те операции, которые ЭВМ способна выполнять.

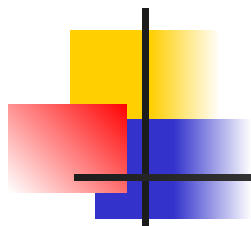


# Величины

---

- Подразделяются на числовые и текстовые, с одной стороны, и на постоянные и переменные — с другой.

- 
- 
- В ЭВМ для любой величины выделяется ячейка памяти.
  - Если это числовая величина, то в ней хранится число, изображающее значение этой величины.
  - Текстовые величины могут быть переменными и постоянными.



# ***Операции над величинами***



## «Ввод»

---

- ЭВМ может считывать конкретные значения исходных величин с различных устройств ввода, например, с клавиатуры, и помещать каждое из них в ячейку, выделенную для соответствующей переменной.





# Операция присваивания

---

- ЭВМ может вычислять значения величины по заданной формуле, содержащей знаки любых арифметических операций, ряда элементарных функций, разные для разных языков программирования.
- В общем виде записывается так:  $x := a$ , где  $x$  — переменная;  $a$  — арифметическое выражение или текст.



# «Вывод» или «Печать»

---

- ЭВМ может печатать на бумаге или выводить на экран монитора значения величин или любой текст.



# «Переход»

---

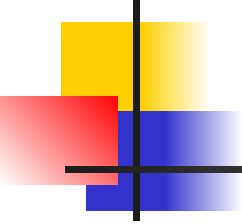
- ЭВМ может переходить от одного этапа решения задачи к любому другому.

# Условный переход (Ветвление)



---

- ЭВМ может сравнивать значения двух арифметических выражений на предмет проверки условий:  $<$ ,  $>$ ,  $=$  и т.д. и в зависимости от результатов проверки выбирать один из двух возможных вариантов дальнейших действий.



---

# ***Способы описания алгоритмов***



# Словесно-формульное описание алгоритма

---

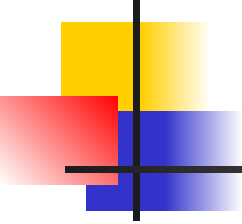
- Описание алгоритма с помощью слов и формул. Это наиболее простой способ.

# Графическое описание алгоритма



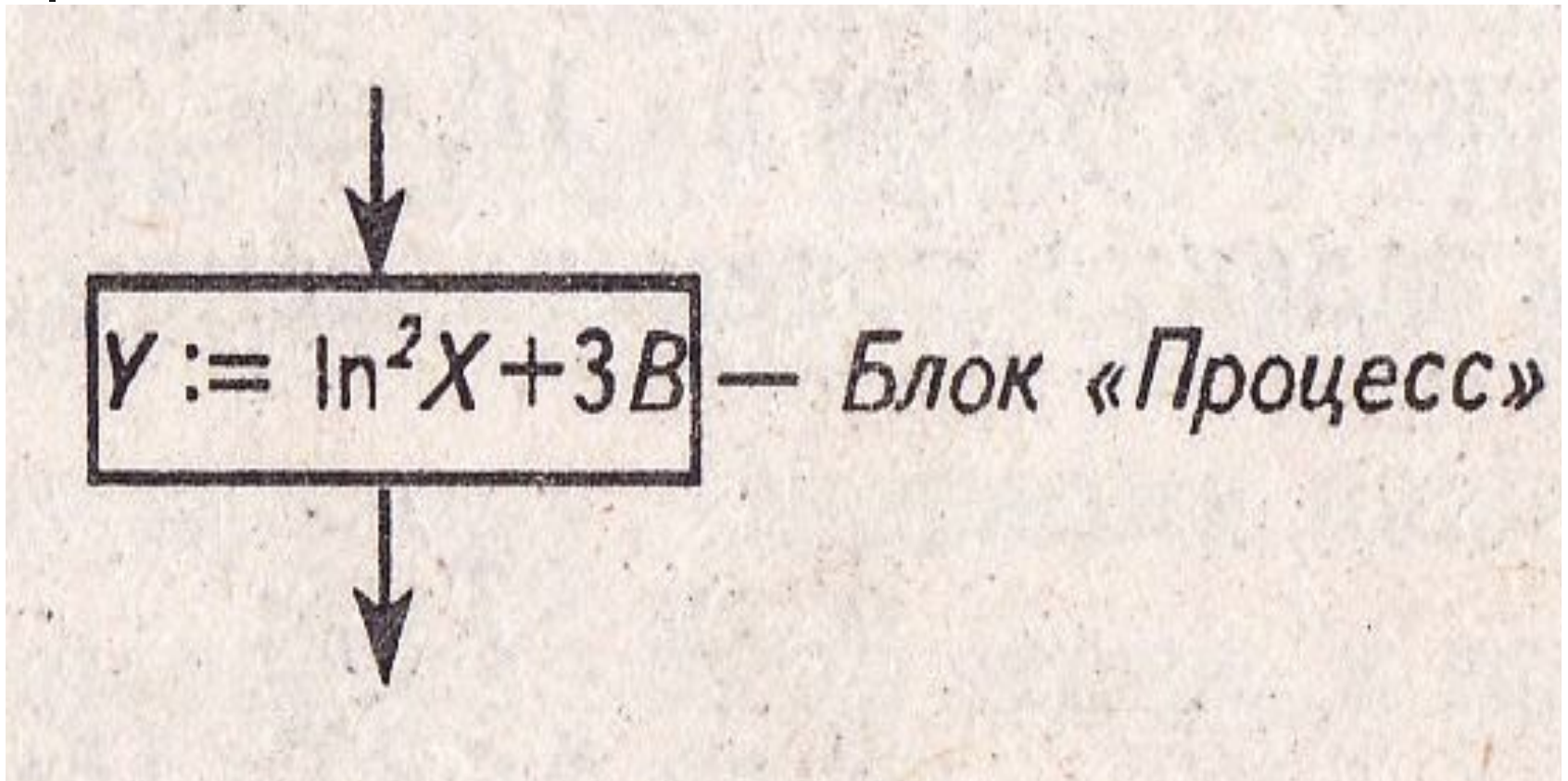
---

- Описание с помощью схем алгоритмов.
- Схема алгоритма представляет собой систему связанных геометрических фигур.
- Каждая фигура обозначает один этап процесса решения задачи и называется блоком.

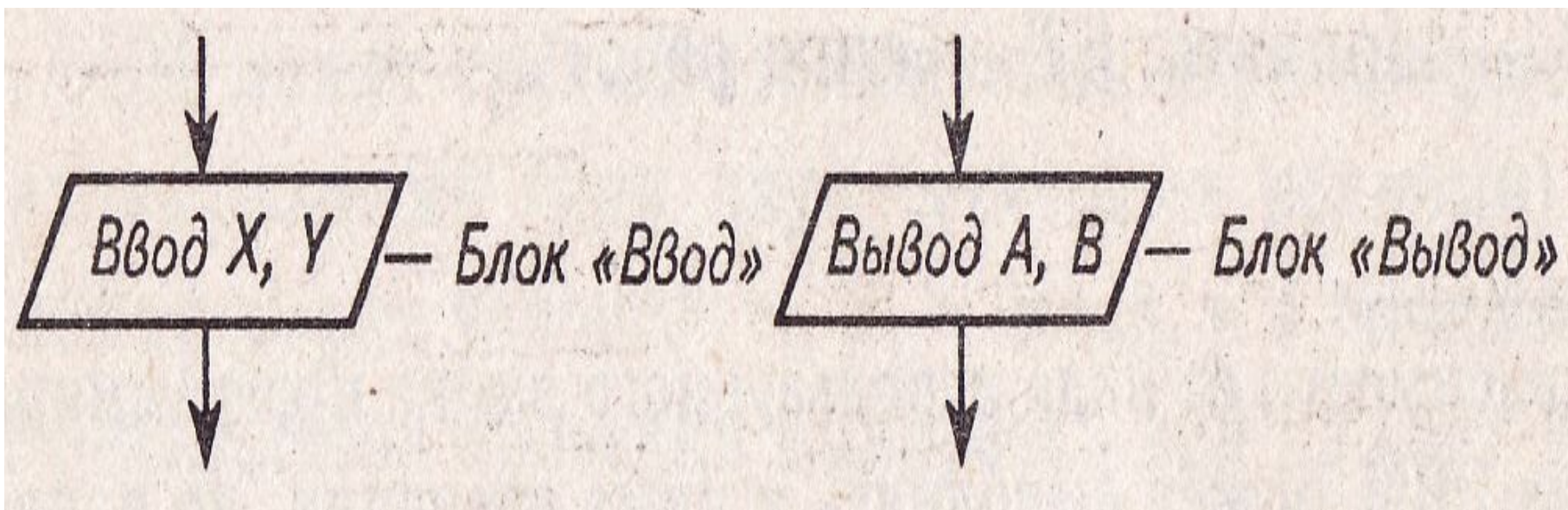
- 
- 
- Порядок выполнения этапов указывается стрелками, соединяющими блоки.
  - В схеме блоки стараются размещать сверху вниз, в порядке их выполнения.
  - Для наглядности операции разного вида изображаются в схеме различными геометрическими фигурами.



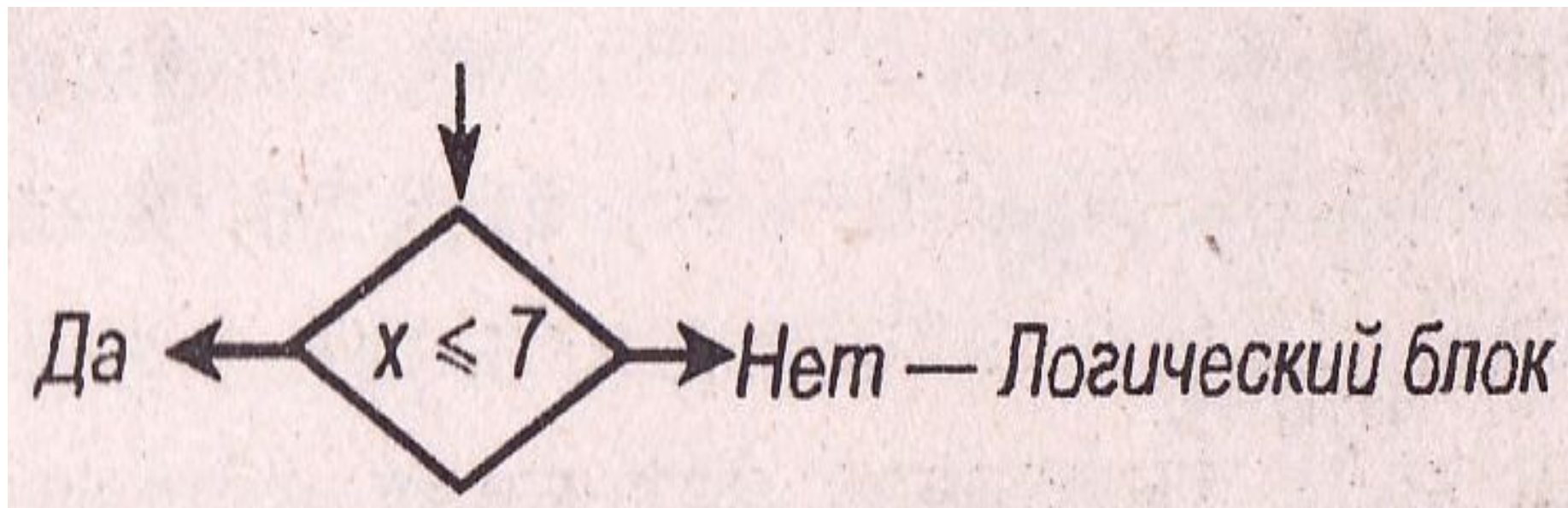
# Операция присваивания



# Операции Ввод и Вывод



# Операция Условный переход





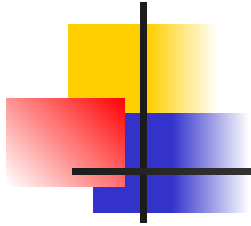
# Блок Начало

---



# Блок Останов





---

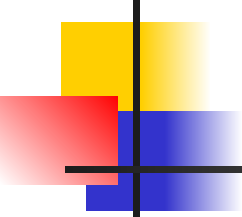
***Описание алгоритма на  
алгоритмическом  
языке (алгоязыке)***



# Алгоритмический язык

---

- Средство для записи алгоритмов в аналитическом виде, промежуточном между записью алгоритма на естественном языке и записью на языке ЭВМ.

- 
- 
- Запись алгоритма на алгоязыке весьма близка к его словесно-формульному описанию.
  - Разница состоит в том, что в алгоязыке используется ограниченный набор терминов, более строгие правила записи операции и т.д. с целью обеспечения однозначности понимания алгоритма.






# Виды алгоритмов

---

- линейные,
- разветвляющиеся,
- циклические.

# Общие правила построения схемы алгоритма задачи



---

- Выявить исходные данные, результаты, назначить им имена.
- Выбрать метод решения задачи.
- Разбить метод решения задачи на этапы.
- Изобразить каждый этап в виде соответствующего блока-схемы алгоритма и указать стрелками порядок их выполнения.

# Основные принципы алгоритмизации



---

- общие правила построения схемы алгоритма задачи;
- предусмотреть выдачу результатов или сообщений об их отсутствии;
- обеспечить возможность после выполнения любой операции так или иначе перейти к выходу схемы.



# Исполнение алгоритмов

---

- Должно привести к решению соответствующей задачи, получению результата.

# Суть процесса исполнения алгоритма



---

- Скрупулезное и педантичное выполнение операций алгоритма в требуемом порядке.

# Исполнение алгоритма, заданного схемой



---

- блоки выполняются строго в порядке, определяемом стрелками, соединяющими их, т.е. после выполнения операций некоторого блока переходим к выполнению операций следующего блока, связанного с первым стрелкой.

# Цели исполнения алгоритма



---

- научить читать и механически выполнять алгоритмы;
- проверить, правильно ли понимаются операции, включаемые в алгоритм;
- проверить правильность составленного алгоритма.



# Отладка алгоритмов

---

- Процесс выявления и исправления ошибок в нем.
- Суть отладки алгоритма в том, что выбирается некоторый набор исходных данных называемый тестовым набором, и задача с этим набором решается дважды: один раз — исполнением алгоритма, второй раз — каким-либо иным способом, исходя из условия задачи.





# Данные тестового набора

---

- обеспечить проверку выполнения всех операций алгоритма;
- свести количество вычислений к минимуму.