**Технология решения задач с помощью компьютера. Алгоритмы. Свойства алгоритмов. Блок-схемы алгоритмов.**

Основные цели:

1. Освоить принципы решения задач с использованием компьютера.
2. Освоить понятие алгоритма, свойства алгоритмов и способы представления алгоритмов.
3. Знать основные блок-схемы алгоритмов.
4. Уметь составлять блок-схемы.
5. Освоить основные конструкции алгоритмического языка.

Решение задач с использование компьютера можно разделить на несколько этапов:

* + Постановка задачи;
  + Определение условия;
  + Построение модели задачи;
  + Описание алгоритма решения задачи;
  + Выбор оптимальных компьютерных средств решения;
  + Описание алгоритма с помощью выбранных программных средств;
  + Тестирование решения задачи;
  + При необходимости, коррекция решения задачи.

Решение задачи начинается с ее постановки**,** которая раскрывает содержание задачи и определяет конечную цель. Задача формулируется на уровне профессиональных понятий и выбирается общий подход к ее решению. Исходные положения и описания задачи должны быть корректны и понятны, так как ошибка в постановке задачи, обнаруженная на последующих этапах, приведет к необходимости начать решение задачи с самого начала.

На этом этапе осуществляется формализация задачи путем описания ее с помощью математических и логических выражений, определяется перечень исходных данных и получаемых результатов, начальные условия, точность и прочее.

Как правило, даже имея математическую модель задачи, не удается получить решение в явном виде, например, в виде формулы, связывающей исходные данные и результаты. В этом случае устанавливается необходимая логическая последовательность математических операций и других действий, с помощью которых будут получены результаты, то есть создается алгоритм решения задачи.

Разработка алгоритма для решения любой задачи является наиболее ответственным и важным моментом, так как именно алгоритм определяет последовательность действий, выполняемых компьютером. Поэтому основной целью при решении задачи является построение хорошего алгоритма.

Необходимо разобрать поподробнее понятие алгоритма. Для решения одной и той же задачи программы могут быть написаны на разных языках программирования и выполняться на разных машинах. Одну и ту же программу разные трансляторы преобразуют в разные последовательности инструкций процессора. Тем не менее, в ходе выполнения программы все различия исчезают, и результат получается одинаковый. Таким образом, программа, не зависимо, на каком языке программирования она написана, содержит нечто постоянное, что и определяет способ решения этой задачи.

Фактически программа содержит порядок действий, благодаря которому возможно решить задачу. При этом совершенно неважно, каким образом эти действия будут выполняться: с помощью компьютера, путем вычислений на бумаге или другим способом. Такой порядок действий называют алгоритмом решения задачи.

|  |
| --- |
| Алгоритм – это строго детерминированная последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью понятных исполнителю команд. (Организованная последовательность действий, допустимых для некоторого исполнителя.) |

Термин алгоритм (algorithmi) происходит от латинского написания имени великого математика 9 века аль-Хорезми, который впервые сформулировал правила выполнения 4-х основных арифметических действий. Эти правила поначалу и называли алгоритмами.

Каждый из нас ежедневно использует различные алгоритмы: инструкции, правила, рецепты и т. п. Обычно мы это делаем не задумываясь. Например, открывая дверь ключом, никто не раздумывает над тем, в какой последовательности выполнять действия. Однако чтобы кого-нибудь (скажем, младшего брата) научить открывать дверь, придется четко указать и сами действия выполнения. Для решения одной и той же задачи программы могут быть написаны на разных языках программирования и выполняться на разных машинах. Одну и ту же программу разные трансляторы преобразуют в разные последовательности инструкций процессора. Тем не менее, в ходе выполнения программы все различия исчезают, и результат получается одинаковый. Таким образом, программа, не зависимо, на каком языке программирования она написана, содержит нечто постоянное, что и определяет способ решения этой задачи.

Фактически программа содержит порядок действий, благодаря которому возможно решить задачу. При этом совершенно неважно, каким образом эти действия будут выполняться: с помощью компьютера, путем вычислений на бумаге или другим способом. Такой порядок действий называют алгоритмом решения задачи.

Алгоритм позволяет формализировать выполнение информационного процесса. Если исполнителем является человек, то он может выполнять алгоритм формально, не вникая в содержание поставленной задачи, а только строго выполняя последовательность действий, предусмотренную алгоритмом.

Рассмотрим работу пользователя. Текстовый редактор представляет пользователю возможность работы в мире своих объектов, которыми являются документы, фрагмент документа, символ и т. д.

Предположим, что пользователю необходимо провести редактирование текста. Пусть у нас есть объект – фрагмент. Надо перевести его из исходного состояния в конечное.

ИНФОРМАЦИОННАЯ МОДЕЛЬ

Модель информационная

Необходимую для реализации такого преобразования последовательность действий, то есть алгоритм, запишем на естественном языке, который понятен пользователь компьютера:

* + Выделить слово "информационная".
  + Вырезать этот фрагмент.
  + Установить курсор на позицию после слова "модель".
  + Вставить вырезанный фрагмент текста.
  + Установить курсор на позицию перед словом "информационная" в первом случае.
  + Вырезать.

Каждая команда алгоритма должна однозначно определять действия исполнителя, для этого необходимо формализировать запись алгоритма и заменить содержательную модель текста на его формальную модель. Формальная модель представляет текст делящимся на страницы, состоящие из определенного набора строк, которые в свою очередь, включают определенное количество знакомест (символов).

Наш текст состоит из одной страницы, которая содержит одну строку. Команде "Выделить слово информационная" на формальном языке соответствует команда "Выделить символы с 1 по 15", а команде "Установить курсор после слова модель" соответствует команда "Установить курсор после 7 символа".

1. Выделить символы с 1 по 15.
2. Вырезать этот фрагмент и поместить его в буфер обмена.
3. Установить курсор на позицию после 7 символа.
4. Вставить вырезанный фрагмент текста.

Теперь этот алгоритм "редактирование" пользователь может выполнять формально. В процессе выполнения алгоритма на компьютере пользователь будет выполнять команды алгоритма с помощью клавиатуры и мыши. Фактически же пользователь будет давать команды объектам программной среды Windows, которые будут действительными исполнителями алгоритма.

Рассмотрим основные свойства алгоритмов.

Алгоритмы состоят из отдельных команд, которые исполнитель выполняет одну за другой в определенной последовательности. Разделение информационного процесса в алгоритме на отдельные команды называется дискретностью.

Запись алгоритма должна быть такова, чтобы, выполнив очередную команду, исполнитель знал, какую команду следует выполнять следующей. Это свойство называется детерминированностью. Должны быть определены начальное состояние объекта и его конечное состояние. Алгоритм должен обеспечивать преобразование объекта из начального состояния в конечное за конечное число шагов. Такое свойство называют результативностью.

Алгоритм может быть представлен:

* + в графическом виде,
  + в словесном описании,
  + в виде таблицы,
  + последовательность формул,
  + записанным на языке программирования.

Словесный способ представления алгоритма имеет ряд недостатков:

* + описание строго не формализуемы;
  + они страдают многословностью записей;
  + допускают неоднозначность толкования отдельных предписаний.

Графический способ представления алгоритмов является наиболее компактным и наглядным по сравнению с словесным. При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Такое графическое представление называется блок-схемой алгоритма. В блок-схеме каждому типу действий соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий.

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных базовых элементов. Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: следование, ветвление, цикл. Характерной особенностью базовых структур является наличие в них одного входа и одного выхода.

* Следование. Образуется из последовательности действий, следующих одно за другим.
* Ветвление. Обеспечивает в зависимости от результата проверки условия выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь выбран.
* Цикл. Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла.

Для единой записи алгоритмов применяют систему обозначений и правил, называемую псевдокод. Он занимает промежуточное место между естественным и формальным языками. В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, но в то же время есть некоторые конструкции, присущие формальным языкам, например, служебные слова. Они выделяются в печатном тексте жирным шрифтом, а на письме подчеркиваются. Примером псевдокода служит школьный алгоритмический язык в русской нотации. Этот язык называют просто «алгоритмический язык».

Основные служебные слова:

**алг** (алгоритм),

**арг** (аргумент),

**рез** (результат).

**нач** (начало),

**кон** (конец),

**цел** (целый)

**сим** (символьный),

**лог** (логический),

**таб** (таблица),

**нц** (начало цикла),

**кц** (конец цикла),

**вещ** (вещественный),

**длин** (длина),

**дано, надо, если, то, иначе, все, для, от, до, знач, и, или, да, нет, при, выбор, ввод, вывод, пока, не, утв.**

Общий вид алгоритма:

**алг** название алгоритма (аргументы и результаты)

**дано** условия применимости алгоритма

**надо** цель выполнения алгоритма

**нач** описание промежуточных величин

последовательность команд (тело алгоритма)

**кон**

Часть алгоритма от слова **алг** до слова **нач** называется заголовком, в часть между словами **нач** и **кон** – телом алгоритма.