

Тема урока: **Программный принцип работы компьютера.**

Изучить лекцию и составить конспект

Алгоритм – последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью понятных исполнителю команд.

Исполнителем алгоритма может быть человек или автоматическое устройство – компьютеры, роботы, станки, спутники, сложная бытовая техника и даже детские игрушки. Каждый алгоритм создается в расчете на вполне конкретного исполнителя.

Компьютер, как исполнитель, любую работу выполняет по программе. Программы пишут люди, а компьютер формально их выполняет.



Разработчики систем искусственного интеллекта пытаются научить машину, подобно человеку, самостоятельно строить программу своих действий, исходя из условия задачи.



Ставится цель превращения компьютера из формального исполнителя в интеллектуального исполнителя.

Работа обоих исполнителей состоит из четырёх блоков, но формальный исполнитель работает по уже готовой программе, а интеллектуальный – сам составляет программу и получает результат.

Информация для компьютера – **данные**, представленные в форме, приемлемой для её передачи и обработки на компьютере.

Для работы с данными компьютеру необходимы инструкции (**команды, правила действия**). Команды формируются в перечень команд.

Алгоритм – это последовательность действий (команд) для достижения цели.

В XIX веке английским математиком и инженером Чарльзом Бэббиджем был разработан проект вычислительной машины, которая предназначалась

для автоматического проведения длинных цепочек вычислений. Главной особенностью конструкции этой машины является программный принцип работы.

Чарльза Беббиджа считают изобретателем компьютера – он впервые соединил механический арифмометр с идеей программного управления.

По своему назначению компьютер – это универсальный прибор для работы с информацией.

В основу работы компьютеров положен программный принцип управления. Любой компьютер представляет собой автоматическое устройство, работающее по заложенным в него программам.

Первая вычислительная машина, способная хранить программу в своей памяти, разрабатывалась в 1943—1948 гг. в США под руководством Джона Мочли и ПреснераЭкерта.

В 1945 г. к работе был привлечен знаменитый математик Джон фон Нейман, который сформулировал общие принципы функционирования универсальных вычислительных устройств.

Первый компьютер, в котором были полностью реализованы эти принципы, был построен в 1949 г. английским исследователем Морисом Уилксом. Изменяется элементная база, компьютеры становятся все более и более мощными, но до сих пор большинство из них соответствуют тем принципам, которые изложил в своем докладе в 1945 г. Джон фон Нейман.

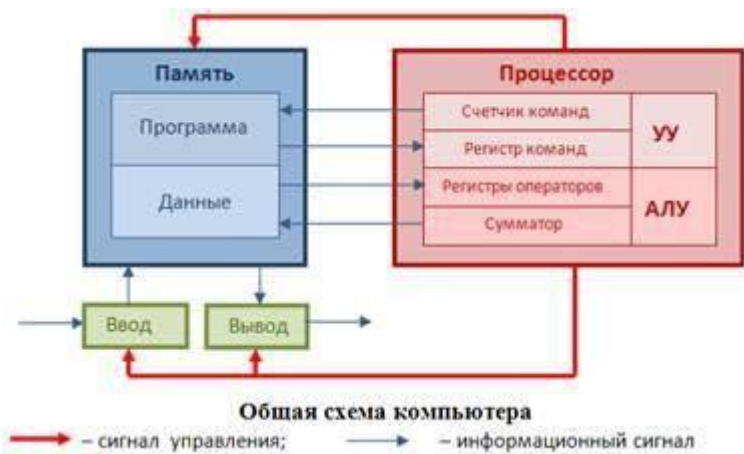
Согласно фон Нейману, ЭВМ состоит из следующих основных блоков:

- арифметико-логическое устройство, выполняющее арифметические и логические операции;
- устройство управления, которое организует процесс выполнения программ;
- запоминающее устройство, или память, для хранения программ и данных;
- внешние устройства для ввода-вывода информации.

В современных компьютерах это:

- память (запоминающее устройство — ЗУ), состоящая из перенумерованных ячеек;
- процессор, включающий в себя устройство управления (УУ) и арифметико-логическое устройство (АЛУ);
- устройство ввода;
- устройство вывода.

Эти устройства соединены между собой каналами связи, по которым передается информация.



Функции памяти: - прием информации из других устройств; - запоминание информации; - выдача информации по запросу в другие устройства машины.

Функции процессора: - обработка данных по заданной программе путем выполнения арифметических и логических операций; - программное управление работой устройств компьютера.

Одна часть процессора, которая выполняет команды, называется *арифметико-логическим устройством*, а другая его часть, выполняющая функции управления устройствами, — *устройством управления*. Обычно эти устройства выделяются чисто условно, конструктивно они не разделены.

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых *регистрами*. Регистр выполняет функцию кратковременного хранения числа или команды. Основным элементом регистра является электронная схема, называемая *триггером*.

Регистр представляет собой совокупность триггеров, связанных друг с другом определенным образом общей системой управления.

Существует несколько типов регистров, отличающихся видом выполняемых операций. Некоторые важные регистры имеют свои названия, например:

- *сумматор* — регистр АЛУ, участвующий в выполнении каждой операции;

- *счетчик команд* — регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды. Он служит для автоматической выборки программы из последовательных ячеек памяти;

- *регистр команд* — регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения кода операции, остальные — для хранения кодов адресов операндов.

Компьютер является универсальным исполнителем по обработке информации. Значит, для него, как для любого исполнителя, существует определенная система команд (СКИ). Такая система команд для компьютера называется **языком машинных команд (ЯМК)**

Программа для компьютера – это алгоритм, разработанный на ЯМК. Или, **Программа управления компьютером** – это последовательность

команд ЯМК, где каждая команда – директива для процессора на выполнение определённого действия.

Рассмотрим этапы выполнения программы.



Согласно принципам Джона фон Неймана, программа во время её исполнения и данные, которые она обрабатывает, находятся в оперативной памяти (принцип хранимой в памяти программы). Процессор исполняет программу начиная с первой команды и заканчивая последней.

- Какое основное свойство оперативной памяти? (*энергозависимость, работает с данными, активными в текущий момент времени*)

Какие есть особенности в восприятии информации человеком и компьютером? (*человек воспринимает информацию с помощью органов чувств, в виде знаков и сигналов, а компьютер воспринимает информацию в виде цифр (0 и 1).*)

- Как сделать так, чтобы программа, написанная человеком была понятна компьютеру? (*нужен способ перевода*)

Для компьютера вся информация должна быть представлена в двоичных кодах, т.е. необходим способ перевода. Такой способ перевода называется **трансляцией**, а выполняет это транслятор.

Вывод: Устройством, которое обрабатывает информацию в компьютере, является процессор, следовательно, алгоритм должен использовать систему команд процессора, или другими словами записан на машинном языке, представляющем собой последовательности нулей и единиц

Сначала программисты, работавшие на компьютерах первого поколения (50-е – 60-е г.г.), составляли программы на ЯМК (в двоичных кодах), но это довольно сложная работа, поэтому для облегчения программирования были созданы языки программирования высокого уровня (ЯПВУ) - это искусственно созданные языки с несколькими десятками слов (операторов) и строгими правилами синтаксиса. Составление программ на ЯПВУ намного проще. Примеры ЯПВУ: Фортран, Паскаль, Бейсик, Си и др.

Для того чтобы процессор мог выполнить программу, написанную на языке программирования, она и данные с которыми она работает должны быть загружены в оперативную память. Программа написана и загружена в оперативную память и для того чтобы процессор ее выполнил в оперативной памяти, должна быть еще и программа переводчик (транслятор), который переводит программу с языка высокого уровня на язык машинных команд

Таким образом, цепочка событий от составления программы на ЯПВУ до получения результатов решения задачи выглядит так



Человек всегда должен понимать ограниченность возможности компьютера как исполнителя, необходимость предусмотреть все тонкости команд, поручаемых компьютеру. Человек разрабатывает алгоритм, записывает его на ЯПВУ и анализирует результаты выполнения программы.

Компьютер является формальным исполнителем программ.

Итак, компьютер не может обойтись без программы и исходных данных, подготовить их может только человек.

Поэтому можно говорить, что **решение задач компьютером** - это формальное исполнение алгоритма (программы), а компьютер является формальным исполнителем.

Компьютер может быть использован для решения самых разнообразных задач, поэтому, исходя из условия задачи, человек решает, каким программным средством пользоваться. Если в состав ПО входят программы, подходящие для решения задач человека, то удобнее ими воспользоваться (текстовый редактор, электронные таблицы, базы данных, презентации).

В случае, если нельзя воспользоваться готовым программным обеспечением, приходится прибегать к программированию (операционные системы, доработка ОС, трансляторы, драйверы, архиваторы, антивирусы).

Теперь рассмотрим следующий вопрос: тестирование программного обеспечения.

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Первые программные системы разрабатывались в рамках программ научных исследований или программ для нужд министерств обороны. Тестирование таких продуктов проводилось строго формализованно с записью всех тестовых процедур, тестовых данных, полученных результатов. Тестирование выделялось в отдельный процесс, который начинался после завершения кодирования, но при этом, как правило, выполнялось тем же персоналом.

В 1960-х много внимания уделялось «исчерпывающему» тестированию, которое должно проводиться с использованием всех путей в коде или всех возможных входных данных. Было отмечено, что в этих условиях полное тестирование программного обеспечения невозможно, потому что, во-первых, количество возможных входных данных очень велико, во-вторых, существует

множество путей, в-третьих, сложно найти проблемы в архитектуре и спецификациях. По этим причинам «исчерпывающее» тестирование было отклонено и признано теоретически невозможным.

В начале 1970-х годов тестирование программного обеспечения обозначалось как «процесс, направленный на демонстрацию корректности продукта» или как «деятельность по подтверждению правильности работы программного обеспечения». В зарождавшейся программной инженерии верификация ПО значилась как «доказательство правильности». Хотя концепция была теоретически перспективной, на практике она требовала много времени и была недостаточно всеобъемлющей. Было решено, что доказательство правильности — неэффективный метод тестирования программного обеспечения. Однако, в некоторых случаях демонстрация правильной работы используется и в наши дни, например, приёмо-сдаточные испытания. Во второй половине 1970-х тестирование представлялось как выполнение программы с намерением найти ошибки, а не доказать, что она работает. Успешный тест — это тест, который обнаруживает ранее неизвестные проблемы. Данный подход прямо противоположен предыдущему. Указанные два определения представляют собой «парадокс тестирования», в основе которого лежат два противоположных утверждения: с одной стороны, тестирование позволяет убедиться, что продукт работает хорошо, а с другой — выявляет ошибки в программах, показывая, что продукт не работает. Вторая цель тестирования является более продуктивной с точки зрения улучшения качества, так как не позволяет игнорировать недостатки программного обеспечения.

В 1980-е годы тестирование расширилось таким понятием, как предупреждение дефектов. Проектирование тестов — наиболее эффективный из известных методов предупреждения ошибок. В это же время стали высказываться мысли, что необходима методология тестирования, в частности, что тестирование должно включать проверки на всем протяжении цикла разработки, и это должен быть управляемый процесс. В ходе тестирования надо проверить не только собранную программу, но и требования, код, архитектуру, сами тесты. «Традиционное» тестирование, существовавшее до начала 1980-х, относилось только к скомпилированной, готовой системе (сейчас это обычно называется системное тестирование), но в дальнейшем тестировщики стали вовлекаться во все аспекты жизненного цикла разработки. Это позволяло раньше находить проблемы в требованиях и архитектуре и тем самым сокращать сроки и бюджет разработки. В середине 1980-х появились первые инструменты для автоматизированного тестирования. Предполагалось, что компьютер сможет выполнить больше тестов, чем человек, и сделает это более надёжно. Поначалу эти инструменты были крайне простыми и не имели возможности написания сценариев на скриптовых языках.

В начале 1990-х годов в понятие «тестирование» стали включать планирование, проектирование, создание, поддержку и выполнение тестов и тестовых окружений, и это означало переход от тестирования к обеспечению

качества, охватывающего весь цикл разработки программного обеспечения. В это время начинают появляться различные программные инструменты для поддержки процесса тестирования: более продвинутые среды для автоматизации с возможностью создания скриптов и генерации отчетов, системы управления тестами, ПО для проведения нагрузочного тестирования. В середине 1990-х годов с развитием Интернета и разработкой большого количества веб-приложений особую популярность стало получать «гибкое тестирование» (по аналогии с гибкими методологиями программирования).

Домашнее задание:

1. Составить конспект темы.

Выслать работу на электронную почту. Lelya.Stepanova.66@inbox.ru