# Предисловие

Вы - талантливый, милый, отзывчивый, но Вам не везет. Вы годами ждете предложения вакансии от работодателя. Обиваете пороги, а если находите работу, то встречаетесь с токсичной атмосферой. Вас выжимают как лимон и увольняют. Значит, Вы используете не тот курс программирования. Поэтому я предлагаю мой курс, который поможет стать программистом. «Курс счастливого программирования».

Одна знакомая говорила, "Русские в Нью-Йорке кончают на компьютер, а потом получают тыщи, тыщи!" Теперь это не катит. Это прошлый век. Также, как и эта ода.

«Ода программисту. Каждый, кто когда-нибудь писал компьютерный код, знает, что человек - не царь природы. На заре компьютерной (и нашей) молодости нас учили "околокомпьютерным аксиомам": 1. самая совершенная программа имеет по крайней мере одну ошибку; 2. самую быструю программу можно ускорить в два раза; 3. если программой долго не пользоваться, то в ней появляются новые ошибки. И когда мы резали бритвой перфокарты, пытаясь исправить неверную набивку, мы по-своему ставили главный вопрос философии - прежде, чем ответить, познаваем ли мир, скажи, познаваема ли операционная система ЕС ЭВМ. Мы доверяли умной электронной машине, но проверяли и полировали, полировали и еще раз полировали код.

Многие сменили профессию или пали в борьбе роковой, угодив в психушку. Но жертвы программистов не были напрасны. Наверняка многие из вас имеют теперь персональные компьютеры, с которыми можно сыграть в шахматы или, заглянув в интернет, узнать прогноз погоды, или пригласить домой веселых девчонок. Я смело говорю, компьютер - мой лучший друг, лучше даже, чем адвокат и психоаналитик. Только нельзя им помыкать, в смысле лить кофе на клавиатуру, или выдыхать табачный дым ему в морду-монитор, он очень обидчив, можно недосчитаться важных файлов. Так выпьем, чтобы наша мечта - "50 лет упорного кодирования - пенсия на Багамах" стала реальностью!»

На самом деле, немного усидчивости и у Вас всё получится.

# 1. В начало начал

## 1.1. История математического моделирования.

Давайте начнем с определений. Система - это совокупность элементов, находящихся во взаимоотношениях и связях друг с другом, которая образует определенную целостность, единство. Как сказал Пифагор, "сведение множества к единому является фундаментальным принципом красоты".

Математическое моделирование - процесс установления соответствия объекту реальности некоторого математического объекта, называемого математической моделью, и исследование этой модели, позволяющее получать характеристики рассматриваемого реального объекта.

Математические модели являются одним из основных инструментов познания человеком явлений окружающего мира. Под математическими моделями понимаются основные закономерности и связи, присущие исследуемому явлению. Это могут быть формулы или уравнения, наборы правил или соглашений, выраженные в математической форме.

С незапамятных времен математика, механика, физика и другие точные естественные науки использовали математические модели для описания изучаемых ими явлений. Так, законы Ньютона полностью определяют закономерности движения планет вокруг Солнца.

Само слово "алгоритм" происходит от имени средневекового узбекского ученого аль-Хорезми. Даже древнегреческие ученые знали алгоритм нахождения числа "π" с высокой точностью. Ньютон предложил эффективный численный метод решения алгебраических уравнений, а Эйлер предложил численный метод решения обыкновенных дифференциальных уравнений. Как вы знаете, модифицированные методы Ньютона и Эйлера до сих пор занимают почетное место в арсенале вычислительной математики.

Используя основные законы механики, относительно легко создать уравнения, описывающие движение космического корабля, например, от Земли до Луны. Однако получить их решение в виде простых формул не представляется возможным. С появлением компьютеров эта задача стала проще. Компьютеры используются для расчета траекторий космических аппаратов.

Методология математического моделирования кратко выражена знаменитой триадой "модель-алгоритм-программа", сформулированной академиком А. А. Самарским из России. Также надо упомянуть 3 источника функционирования компьютера: аппаратное обеспечение, промежуточное программное обеспечение, программное обеспечение.

Компьютерные технологии начали развиваться еще до открытия электричества. В 1642 году французский математик Блез Паскаль изобрел устройство для сложения десятичных чисел. В основе устройства были шестерни на стержнях.

В 1673 году немецкий философ, математик и физик Готфрид Вильгельм Лейбниц создал "ступенчатый вычислитель" - вычислительную машину, которая позволяет складывать, вычитать, умножать, делить и извлекать квадратные корни, используя двоичную систему счисления.

В 1822 году английский математик Чарльз Бэббидж выдвинул идею создания программно-управляемой вычислительной машины, имеющей арифметическое устройство, устройство управления, ввода и печати. "Разностная машина", так называлось детище Чарльза, работала на паровом двигателе и считывала данные с перфокарт.

В 1943 году в Великобритании был построен компьютер Colossus Mark 1 с использованием 1500 электронных ламп для расшифровки перехваченных немецких шифров. Алан Тьюринг участвовал в разработке математической модели компьютера. Компьютер был собран в декабре 1943 года, а первые реальные расшифровки были сделаны в январе 1944 года после того, как он был установлен в Блетчли-парке.

1945 год. Джон фон Нейман, американский ученый, выдвинул идею использования внешних запоминающих устройств для хранения программ и данных. Фон Нейман разработал структурную принципиальную схему компьютера. Схеме фон Неймана соответствуют большинство современных компьютеров.

ENIAC – первое электронное устройство, которое можно было программировать для решения математических задач. На создание ENIAC ушло 200 000 человеко-часов и 486 804 доллара США. В общей сложности комплекс включал 17 468 ламп 16 различных типов, 7 200 кремниевых диодов, 1 500 реле, 70 000 резисторов и 10 000 конденсаторов общим весом 30 тонн.

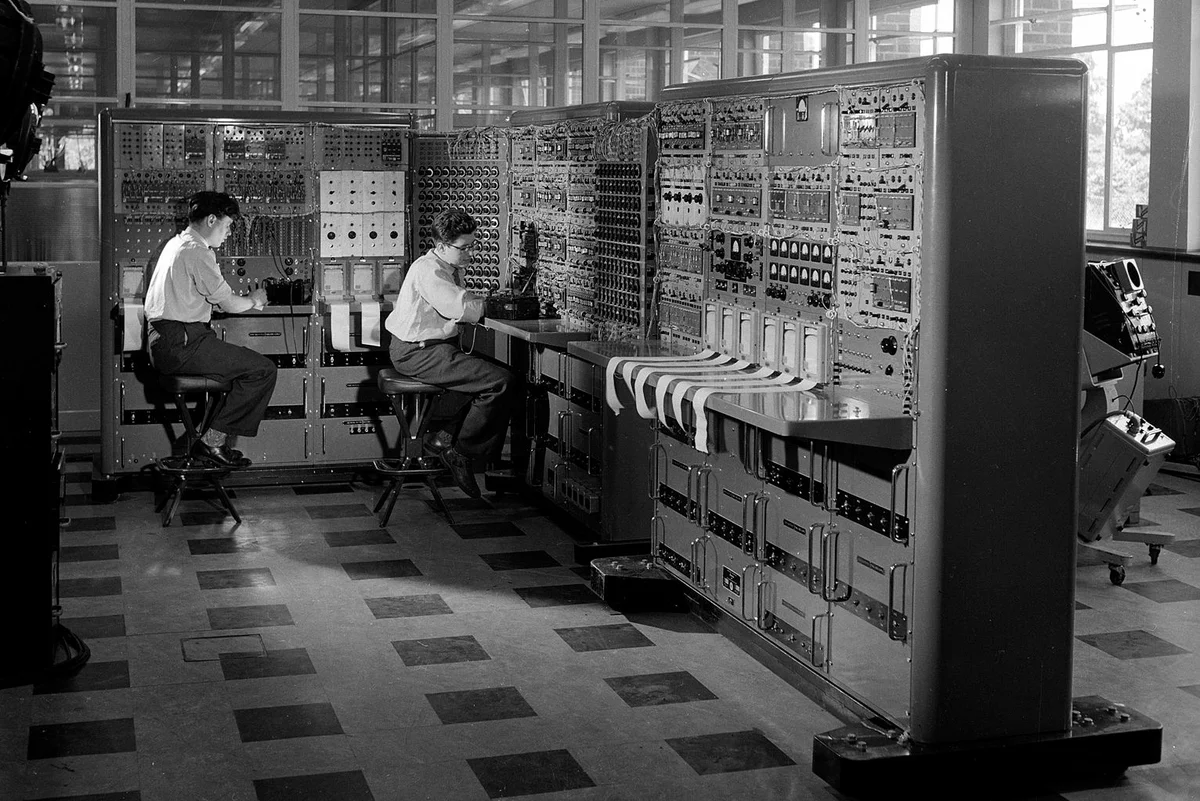


Рис. 1.1. Пользователи ENIAC

В 1955 году изготовлен первый в мире компьютер на основе полупроводниковых транзисторов и диодов без использования электронных ламп - TRADIC. Этот компьютер был разработан Bell Labs для ВВС США и насчитывал 700 транзисторов и 10 000 германиевых диодов. А в 1968 году был создан первый компьютер на основе интегральных схем.

1971. Именно с этого года начинается разработка микропроцессоров, и первый микропроцессор был создан американской компанией Intel. Он назывался Intel 4004. В том же году были внедрены интегральные схемы памяти.

В 1973 году появился процессор Intel 8080, который надолго стал стандартом для процессоров.

Операционная система (ОС) - системное программное обеспечение, благодаря которому активируются технические средства компьютера. Первой операционной системой для компьютеров была GM-NAA. Он был создан в 1955 году Робертом Патриком из General Motors и Оуэном Моком из North American Aviation. Она был основан на системных мониторах и работала исключительно на больших машинах. Основная функция GM-NAA заключается в автоматическом выполнении новой программы после завершения старой программы.

Затем была разработана ОС для мейнфрейма. Мэйнфрейм - это высокопроизводительный компьютер, предназначенный для обработки больших объемов данных и хранения больших объемов данных, а также обладающий большим объемом памяти. Первый мэйнфрейм был создан IBM в 1964 году и назывался IBM System/360.

В настоящее время наиболее популярными операционными системами являются Windows, UNIX, Linux и Mac OS.

## 1.2. Эволюция языков программирования.

Стремительное развитие новых языков программирования началось сразу же после появления первых компьютеров с электронными лампами. В то время компьютеры стоили очень дорого. Покупка одного компьютера стоила в десятки или даже сотни раз дороже, чем стоимость разработки любой программы. Для такой машины требовался высокопроизводительный код, который вручную составлялся на Ассемблере.

Только в 50-х годах был разработан первый алгоритмический язык для личных нужд IBM. Он назывался FORTRAN. Конечно, к тому времени уже существовало несколько разработок в языках, преобразующих арифметические выражения в машинный код, но именно создание FORTRAN считается началом новой эры - переходом к парадигме алгоритмических языков.

Уже в конце 50-х годов появилась хорошая альтернатива FORTRAN. Питер Наур разработал алгоритмический язык, независимый от архитектуры. Он назывался ALGOL. В этом случае разработчики также попытались расширить возможности языка, максимально приблизив систему обозначений к математической.

В начале 60-х годов производители IBM представили не столь популярный язык PL/I. Он работал с архитектурой IBM 360 и служил своего рода расширением FORTRAN, используя некоторые инструменты языка COBOL. Первым языком, в котором введено понятие класса, является Simula-67. Он был создан Дал и Найард в конце 60-х годов.

70-е годы произвели на свет Pascal. Он быстро приобрел массовую популярность. Чуть позже Министерство обороны США приступило к разработке языка высокого уровня. Проект был разработан 4 независимыми группами, а разработка требований и спецификаций была основана на языке Pascal. Полученный продукт был выпущен в начале 80-х годов, он получил название Ada.

70-е годы также дали нам универсальный язык Си. Он был написан Кеном Томпсоном и Деннисом Ричи. Язык был очень популярен среди системных программистов, и на нем было разработано первое ядро UNIX. В 1982 году стандарт C начал разрабатываться в ANSI, и полученная версия была принята в 1990 году. На основе этого языка были разработаны современные языки Java и C++.

В дополнение к языкам, которые работают с алгоритмами, также были разработаны другие языки. COBOL был разработан для обработки бизнес-информации, а Prolog и LISP считаются языками искусственного интеллекта. Кроме того, последний широко используется в теории игр.

С переходом на персональные компьютеры языки программирования стали отдельной частью сред разработки. Сейчас есть даже языки, используемые в офисных программах - например, VBA.

В настоящее время активно развиваются многопроцессорные системы, сервисы и микросервисы. Если вы посмотрите на сайты поиска работы для программистов, то самыми популярными языками являются Java, C++, C#, JavaScript и Python. В этом курсе мы будем использовать C#.

## 1.3 Основные структуры данных и алгоритмы.

Объект в программировании - некоторая сущность в цифровом пространстве, обладающая определённым состоянием и поведением, имеющая определённые свойства (атрибуты) и операции над ними (методы). Как правило, при рассмотрении объектов выделяется то, что объекты принадлежат одному или нескольким классам, которые определяют поведение (являются моделью) объекта. Термины «экземпляр класса» и «объект» взаимозаменяемы.

Тип данных определяет, что именно представляют собой данные, как они хранятся в памяти, какие операции с ними можно выполнять. Изначально, типы данных делятся на простые и составные. Простой - это тип данных, объекты (переменные или постоянные) которого не имеют доступной программисту внутренней структуры. Для объектов составного типа данных, в противовес простому, программист может работать с элементами внутренней его структуры.

Класс – это способ описания объекта (сущности), определяющий состояние и поведение, зависящее от этого состояния, а также правила для взаимодействия с данной сущностью (контракт). С точки зрения программирования класс можно рассматривать как набор данных (полей, атрибутов, членов класса) и функций для работы с ними (методов).

Интерфейс (англ. interface) — программная/синтаксическая структура, определяющая отношение между объектами, которые разделяют определённое поведенческое множество и не связаны никак иначе. При проектировании классов, разработка интерфейса тождественна разработке спецификации (множества методов, которые каждый класс, использующий интерфейс, должен реализовывать).

Коллекция - это структура данных (тип, класс или, еще лучше, интерфейс), предназначенная для хранения определенного количества объектов (в зависимости от языка и терминологии они должны быть одного типа или могут быть разных типов).

Различные типы коллекций могут быть статическими или динамическими, они могут изменять свой размер или оставаться постоянными, они могут быть упорядоченными (точнее, они учитывают порядок элементов) и неупорядоченными (соответственно, они не учитываются).

Наиболее часто используемые структуры: массив, список, набор, словарь, кортеж.

Граф - это совокупность точек, соединенных линиями. Точки называются вершинами или узлами, а линии называются ребрами или дугами.

Степень входа вершины - это количество ребер, входящих в нее, а степень выхода-количество исходящих ребер.

Граф, содержащий ребра между всеми парами вершин, является полным.

Существуют графы, ребра которых соответствуют определенному числовому значению. Они называются взвешенными графами, и это значение является весом ребра.

Когда у ребра оба конца совпадают, т. е. оно покидает вершину и входит в нее, такое ребро называется петлей.

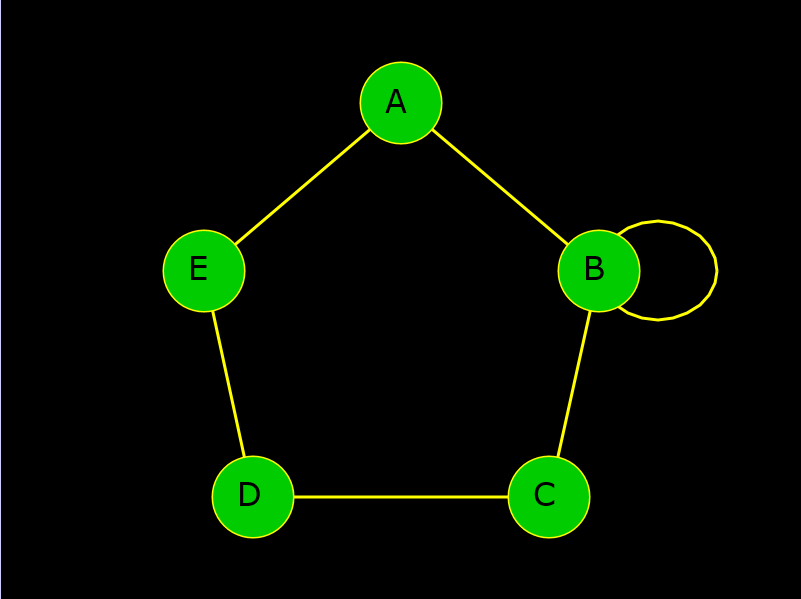


Рис. 1.2. Граф с петлей

Дерево - это связный граф без циклов. Корень - это самая верхняя вершина дерева. Лист-это вершина, у которой нет потомков. Обход дерева - систематический просмотр всех вершин, при котором каждая вершина встречается один раз.

Двоичное дерево – дерево, в котором у всех вершин не более двух ребер.

Двоичное дерево поиска - это двоичное дерево, удовлетворяющее следующим условиям:

дочерние узлы - это бинарные деревья поиска;

для произвольного узла:

все значения левого поддерева меньше значения родительского узла;

все значения правого поддерева больше значения родительского узла.

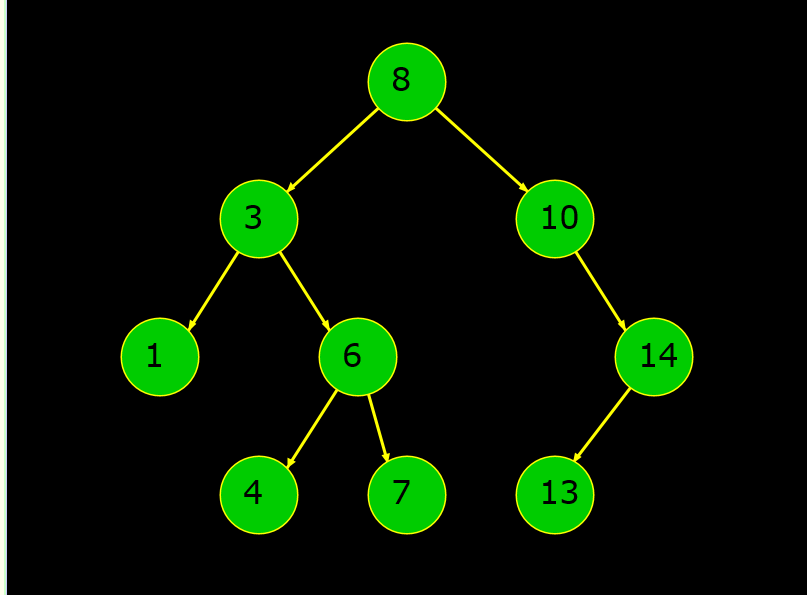


Рис. 1.3. Двоичное дерево поиска

Конечный автомат (или просто FSM - Finite-state machine) - это вычислительная модель, основанная на гипотетической машине состояний. Одновременно может быть активным только одно состояние. Следовательно, для выполнения каких-либо действий машина должна изменить свое состояние.

Конечный автомат можно представить в виде графа, вершины которого являются состояниями, а ребра - переходами между ними. У каждого ребра есть метка, которая информирует вас о том, когда должен произойти переход.

Реализация конечного автомата начинается с определения его состояний и переходов между ними. Представьте себе конечный автомат, описывающий действия муравья, несущего листья в муравейник. Отправной точкой является состояние "искать лист", которое остается активным до тех пор, пока муравей не найдет лист. Когда это произойдет, состояние изменится на "домой с листом". То же самое состояние будет оставаться активным до тех пор, пока наш муравей не доберется до муравейника. После этого состояние снова изменится на " искать лист". При поиске листа возможно появление лисы. Тогда переход в состояние «прятаться». Лиса ушла, возврат в «искать лист».

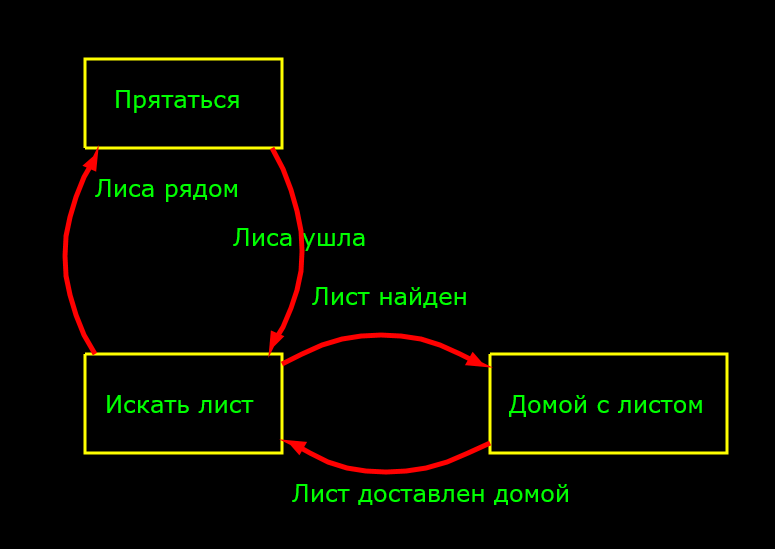


Рис. 1.4. Пример конечного автомата

Значения в моделировании бывают:

дискретные (прерывистые) значения, которые принимают только разделенные значения, которые могут быть пронумерованы;

непрерывный (аналоговый), который может принимать любое значение с определенного интервала.