ВВЕДЕНИЕ

Курсовая работа - это самостоятельный исследовательский труд студента, выполненный на основе использования научной, учебной, методической литературы и прочих учебно-методических материалов. При подготовке и написании курсовой работы перед студентами ставятся следующие задачи:

* Приобрести навыки самостоятельной работы с литературой по
* выбранной тематике;
* Систематизировать и закрепить теоретические знания по дисциплине;
* Развить навыки решения практических задач;
* Уметь связно излагать материал, касающийся выбранной проблематики;
* Самостоятельно производить необходимые виды анализа для разработки программ.

Курсовая работа выполняется каждым студентом в соответствии с полученным индивидуальным заданием и представляется на рецензирование с последующей ее устной защитой и оценкой.

Целью курсовой работы является закрепление знаний и навыков, полученных в процессе изучения дисциплины. Получение практических навыков по реализации программных средств на языке программирования Си.

Программирование можно рассматривать как искусство, науку, ремесло. Программирование — это искусство получения ответов от машины. Для этого в узком смысле нужно составить специальный код для технического устройства, а в широком — разработать программы на языках программирования, т. е. не просто составить код, а выполнить интеллектуальную работу по составлению высокоразумных программ для решения различных задач во всех сферах человеческой деятельности.

Программирование — процесс описания последовательности действий решения задачи средствами конкретного языка программирования и оформление результатов описания в виде программы. Эта работа требует точности, аккуратности и терпения. Команды машине должны формулироваться абсолютно четко и полно, не должны содержать никакой двусмысленности.

Уровень программирования определяется четырьмя взаимосвя­занными факторами развития: возможностями компьютеров, теори­ей и языками, искусством и технологией программирования.

1. История и этапы развития программирования

С глубокой древности известны попытки создать устройства, ус­коряющие и облегчающие процесс вычислений. Еще древние греки и римляне применяли приспособление, подобное счетам, — абак. Такие устройства были известны и в странах Древнего Востока. В XVM в. немецкие ученые В. Шиккард (1623), Г.Лейбниц (1673) и французский ученый Б. Паскаль (1642) создали механические вы­числительные устройства — предшественники всем известного арифмометра. Вычислительные машины совершенствовались в те­чение нескольких веков. Но при этом не применялось понятие «программа и программирование».

В 1854 г. английский математик Джордж Буль опубликовал кни­гу «Законы мышления», в которой развил алгебру высказываний —Булеву алгебру. На ее основе в начале 80-х гг. XIX в. построена тео­рия релейно-контактных схем и конструирования сложных дискрет­ных автоматов. Алгебра логики оказала многогранное влияние на развитие вычислительной техники, являясь инструментом разработ­ки и анализа сложных схем, инструментом оптимизации большого числа логических элементов, из многих тысяч которых состоит со­временная ЭВМ.

Идеи Ч. Бэббиджа реализовал американский ученый Г. Холлерит, который с помощью построенной счетно-аналитической маши­ны и перфокарт за три года обработал результаты переписи населе­ния в США по состоянию на 1890 г. В машине впервые было ис­пользовано электричество. В 1896 г. Холлеритом была основана фирма по выпуску вычислительных перфорационных машин и пер­фокарт.

В 1936 г. английский математик А. Тьюринг ввел понятие ма­шины Тьюринга, как формального уточнения интуитивного понятия алгоритма. Ученый показал, что любой алгоритм в некотором смысле может быть реализован на машине Тьюринга, а следователь­но, доказывал возможность построения универсальной ЭВМ. И та, и другая машины аналогично могут быть снабжены исходными данными решаемой задачи и программой ее решения. Машину Тьюринга можно считать как бы идеализированной моделью универ­сальной ЭВМ.

В 40-х гг. XX в. механическая элементная база вычислительных машин стала заменяться электрическими и электронными устройствами. Первые электромеханические машины были созданы в Германии К. Цузе (Ц-3, 1941 г.) и в США под руководством профессора Гарвардского университета Г. Айкена (МАРК-1, 1944 г.). Первая электронная машина создана в США группой инженеров под руководством доктора Пенсильванского университета Дж. Мочли и аспиранта Дж. Экксрта (ЭНИАК — электронный числовой интегратор и калькулятор, 1946 г.). В 1949 г. в Англии была построена EDSAC — первая машина, обладающая автоматическим программным управ­лением, внутренним запоминающим устройством и другими необхо­димыми компонентами современных ЭВМ.

Логические схемы вычислительных машин были разработаны в конце 1940-х гг. Дж. фон Нейманом, Г. Гольдстайном и А. В. Берксом. Особый вклад в эту работу внес американский математик Джон фон Нейман, принимавший участие в создании ЭНИАК. Он предложил идею хранения команд управления и данных в машин­ной памяти и сформулировал основные принципы построения современных ЭВМ. ЭВМ с хранимой программой оказались более быстродействующими и гибкими, чем ранее созданные.

В 1951 г. в США было налажено первое серийное производство электронных машин УНИВАК (универсальная автоматическая вы­числительная машина). В это же время фирма IBM начала серий­ный выпуск машины IBM/701.

В СССР первыми авторами ЭВМ, изобретенной в декабре 1948 г., являются И. С. Брук и Б. И. Рамеев. А первая советская ЭВМ с сохраняющейся программой создана в 1951 г. под руково­дством С. А Лебедева (МЭСМ — малая электронная счетная маши­на). В 1953 г. в Советском Союзе начался серийный выпуск машин, первыми их которых были БЭСМ-1, «Стрела».

С появлением цифровых программно-управляемых машин родилась новая область прикладной математики — программирование. Как область науки и профессия она возникла в 1950-х гг. Первона­чально программы составлялись вручную на машинных языках (в машинных кодах). Программы были громоздки, их отладка — очень трудоемка. Для упрощения приемов и методов составления и отладки программ были созданы мнемокоды, по структуре близкие к машинному языку и использующие символьную адресацию. Ассемблеры переводили программу, записанную в мнемокоде, на машинный язык и, расширенные макрокомандами, используются и в настоящее время. Далее были созданы автокоды, которые можно применять на различных машинах, и позволившие обмениваться программами. Автокод — набор псевдокоманд для решения специализиро­ванных задач, например, научных или инженерных. Для таких задач имеется развитая библиотека стандартных программ.

До конца 1950-х гг. ЭВМ основным элементом конструкции были электронные лампы (1-е поколение). В этот период развитие идеологии и техники программирования шло за счет достижений американских ученых Дж. фон Неймана, сформулировавшего ос­новные принципы построения ЭВМ, и Дж. Бэкуса, под руково­дством которого в 1954 г. был создан Fortran (Formula Translation) — первый язык программирования высокого уровня, используемый до настоящего времени в разных модификациях. Так, в 1965 г. в Дартмутском колледже Д. Кэмэни и Т. Куртцем была разработана упро­щенная версия Фортрана — Basic. В 1966 г. комиссия при Амери­канской ассоциации стандартов (ASA) разработала два стандарта языка: Фортран и Базисный Фортран. Используются также даль­нейшие модификации языка (например 1970, 1990 гг.).

В США в 1954 г. стал применяться алгебраический подход, совпадающий, по существу, с операторным методом. В 1956 г. корпора­цией IBM разработана универсальная ПП Фортран для автоматического программирования на ЭВМ IBM/704.

Усложнение структуры ЭВМ привело (в 1953 г. для машин И-го поколения) к созданию операционных систем (ОС) — специальных управляющих программ для организации и решения задач на ЭВМ. Например, мониторная система МТИ, созданная в Массачусетском технологическом институте, обеспечивала пакетную обработку, т. е. непрерывное, последовательное прохождение через ЭВМ многих групп (пакетов) заданий и пользование библиотекой служебных программ, хранимой в машине. Это позволило совместить операции по запуску с выполнением программ.

Для ПЭВМ к настоящему времени разработаны ОС: MS DOS, Windows, ОС/2, МасОС, Unix, Linux и др. Широкое распростране­ние получили ОС MS DOS и Windows, имеющие развитый интер­фейс и широкий набор приложений, позволяющих последователь­ное выполнение заданий из пакета, обработку различной информа­ции во многих сферах человеческой деятельности.

Различают технологии, используемые на конкретных этапах разработки или для решения отдельных задач этих этапов, и технологии, охватывающие несколько этапов или весь процесс разработки. В основе первых, как правило, лежит ограниченно применимый метод, позволяющий решить конкретную задачу. В основе вторых обычно лежит методология, определяющая совокупность методов, используемых на разных этапах разработки.

1 этап - «стихийное» программирование. Первые программы имели простейшую структуру, состояли из программы на машинном языке и обрабатываемых ею данных. Сложность программ в машинных кодах ограничивалась способностью программиста одновременно мысленно отслеживать последовательность выполняемых операций и местонахождение данных при программировании.

Появление ассемблеров. Программы стали более «читаемыми».

Создание языков программирования высокого уровня существенно упростило программирование вычислений, снизив уровень детализации операций. Это, в свою очередь, позволило увеличить сложность программ.

2 этап - структурный подход к программированию - представляет собой совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки ПО. В основе – декомпозиция сложных систем с целью последующей реализации в виде отдельных небольших подпрограмм. С появлением других принципов декомпозиции данный способ получил название процедурной декомпозиции. Структурный подход требовал представления задачи в виде иерархии подзадач простейшей структуры. Проектирование осуществлялось «сверху – вниз» и подразумевало реализацию общей идеи, обеспечивая проработку интерфейсов подпрограмм.

Из-за роста сложности и размеров разрабатываемого ПО потребовалось развитие структурирования данных. Следовательно, в языках появляется возможность определения пользовательских типов данных. - появилась и начала развиваться технология модульного программирования. Модульное программирование -выделение групп подпрограмм, использующих одни и те же глобальные данные в отдельно компилируемые модули (библиотеки подпрограмм), Связи между модулями при использовании данной технологии осуществляются через специальный интерфейс, в то время как доступ к реализации модуля запрещен.

3 этап - объектный подход к программированию (с середины 80-х до конца 90-х годов XX в.). Объектно-ориентированное программирование - технология создания сложного ПО, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств. Взаимодействие программных объектов в такой системе осуществляется путем передачи сообщений.

Были созданы среды, поддерживающие визуальное программирование. При использовании визуальной среды появляется возможность проектировать некоторую часть, например, интерфейсы будущего продукта, с применением визуальных средств добавления и настройки специальных библиотечных компонентов.

4 этап –компонентный подход и CASE-технологии. Компонентный подход - построение ПО из отдельных компонентов – физически отдельно существующих частей программного обеспечения, которые взаимодействуют между собой через стандартизованные двоичные интерфейсы. В отличие от обычных объектов объекты-компоненты можно собрать в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (без исходных текстов) и использовать в любом языке программирования, поддерживающем соответствующую технологию.

Отличительной особенностью современного этапа развития технологии программирования, кроме изменения подхода, является создание и внедрение автоматизированных технологий разработки и сопровождения программного обеспечения, которые были названы CASE-технологиями. На сегодня существуют CASE-технологии, поддерживающие как структурный, так и объектный подходы к программированию.

Появление нового подхода не означает, что отныне все программное обеспечение будет создаваться из программных компонентов, но анализ существующих проблем разработки сложного программного обеспечения показывает, что он будет применяться достаточно широко.

1. Классификация языков программирования и сред разработки