

Содержание:

Введение

На сегодняшний день очень важное значение в сфере компьютерных технологий занимает программное обеспечение. Оно, на уровне с аппаратным обеспечением, обеспечивает удобное использование пользователем вычислительных ресурсов, доступных современному обществу. Создание программного обеспечения является очень сложным процессом, состоящим из многих стадий и этапов.

На сегодняшний день существует большое количество разновидностей программного обеспечения, которые появлялись постепенно, в соответствии с растущими требованиями пользователей к использованию вычислительных ресурсов.

Актуальность темы исследования заключается в том, что программное обеспечение является перспективной на сегодняшний день технологией, имеющей много различных аспектов. Обзор ключевых различий, достоинств и недостатков каждой разновидности должен помочь в выборе наиболее подходящей под определенные цели программы. Именно по этой причине является актуальным изучение разновидностей программного обеспечения и рассмотрение каждого вида в отдельности.

Объектом исследования является программное обеспечение и его историческое развитие. Предметом исследования являются разновидности программного обеспечения.

Целью работы является получение, структурирование и сравнение знаний о разновидностях программного обеспечения, посредством изучения каждой из них.

Задачами данной работы являются:

- изучение понятия программного обеспечения;
- обзор истории появления и развития программного обеспечения;
- изучение разновидностей классификаций программного обеспечения;
- краткий обзор каждой разновидности программного обеспечения.

В основу изучения вопроса создания и проектирования программного обеспечения были положены работы Виктора Мелехина и Евгения Павловского, а также некоторых авторов журнала «Программные продукты и системы».

1. Современное программное обеспечение персонального компьютера

1.1. Понятие программного обеспечения

Программное обеспечение, согласно стандарту ISO/IEC 2382-1:1993, представляет собой все или часть правил, процедур, программ и соответствующей документации системы обработки информации.

Также существуют и другие определения из российских и международных стандартов:

- согласно стандарту IEEE Std 829—2008, программное обеспечение является компьютерными программами, процедурами и, возможно, соответствующей документацией и данными, которые относятся к функционированию компьютерной системы;
- согласно тому же стандарту IEEE Std 829—2008, программное обеспечение является программой или множеством программ, которые используются для управления компьютером;
- согласно стандарту ГОСТ 19781-90, программное обеспечение является совокупностью программ системы обработки информации и программных документов, которые необходимы для эксплуатации этих программ[\[1\]](#).

Программное обеспечение представляет один из видов обеспечения вычислительной системы, наряду с математическим, аппаратным, лингвистическим, информационным, методическим и организационным обеспечением.

Программное обеспечение представляет собой то, что делает компьютеры универсальными и позволяет использовать типовую вычислительную машину с целью решения самых разнообразных задач[\[2\]](#).

Академическими областями, которые изучают программное обеспечение, являются программная инженерия и информатика.

В компьютерном сленге часто используется слово «софт», которое произошло от английского слова «software», впервые в этом смысле примененного в 1958 году в статье журнала *American Mathematical Monthly* Джон Тьюки, математиком из Принстонского университета[3] [1, 3].

1.2. История программного обеспечения

Первую программу написала Ада Лавлейс для разностной машины Чарльза Бэббиджа, но так как эта машина не была достроена, разработки леди Лавлейс остались чисто теоретическими.

Первая теория, касающаяся программного обеспечения, была предложена английским математиком Аланом Тьюрингом в 1936 году в эссе «On computable numbers with an application to the Entscheidungsproblem» («О вычислимых числах с приложением к проблеме разрешения»). Он создал так называемую машину Тьюринга, математическую модель абстрактной машины, способной выполнять последовательности рудиментарных операций, которые переводят машину из одного фиксированного состояния в другое. Главная идея заключалась в математическом доказательстве факта, что любое наперед заданное состояние системы может быть всегда достигнуто последовательным выполнением конечного набора элементарных команд (программы) из фиксированного набора команд[4].

Первые электронно-вычислительные машины 1940—1950-х годов перепрограммировались путем переключения тумблеров и переподключения кабелей, что требовало глубокого понимания их внутреннего устройства. К таким машинам, в частности, относился ENIAC[5].

Важным шагом в сторону современных компьютеров был переход к архитектуре Джона фон Неймана, впервые реализованной в Великобритании, в разработанной под руководством Дж. Р. Уомерзли и при участии Алана Тьюринга компьютере, известном как Марк I. Первая программа, хранимая в памяти компьютера, была запущена на нем 21 июня 1941 года. Для облегчения программирования этой машины Тьюринг придумал систему сокращенного кодирования, в которой для представления двоичного машинного кода использовалась последовательность телетайпных символов, выводимых на перфоленту[6].

Один из сотрудников Тьюринга, Джон Мочли, став позднее руководителем и сооснователем компании Eckert-Mauchly Computer Corporation, разработавшей такие ЭВМ, как BINAC и UNIVAC, поручил своим сотрудникам создать транслятор алгебраических формул. Хотя эта амбициозная цель в 1940-х годах и не была достигнута, под руководством Мочли был разработан так называемый «Краткий код», в котором операции и переменные кодировались двухсимвольными сочетаниями. Краткий код был реализован с помощью интерпретатора. Грейс Хоппер, работая с начала 1950-х годов над набором математических подпрограмм для UNIVAC I, изобрела программу-компоновщик «A-0», которая по заданному идентификатору осуществляла выборку нужной подпрограммы из библиотеки, хранящейся на магнитной ленте, и записывала ее в отведенное место оперативной памяти[7].

В 1950-е годы появились первые высокоуровневые языки программирования, Джон Бэкус разработал FORTRAN, а Грейс Хоппер — COBOL. Подобные разработки значительно упростили написание прикладного программного обеспечения, которое писала тогда каждая фирма, приобретающая вычислительную машину[8].

В начале 1950-х годов понятие программного обеспечения еще не сложилось. Так не говорилось о нем ничего в вышедшей в январе 1952 года в журнале Fortune статье «Office Robots», описывавшем компьютеры Univac. Хотя в статье уже рассказывается о компьютере, как о универсальном устройстве, процесс программирования в этой статье был анахронически описан как «переключение тумблеров». Однако к середине 50-х годов уже вполне сложилась разработка программного обеспечения на заказ, хотя сам термин «программное обеспечение» еще не использовался, тогда говорили просто о «программировании на заказ» или «программистском обслуживании». Первой программной фирмой стала компания System Development Corporation, созданная в 1956 году на базе принадлежащей правительству США фирме RAND Corporation. На этом этапе заказчиками уникального и нетиражируемого программного обеспечения были крупные корпорации и государственные структуры, и стоимость в один миллион долларов за программу не была чем-то необычным[9].

Собственно, сам термин программного обеспечения вошел в широкий обиход с начала 1960-х годов, когда стало актуальным разграничение команд, управляющих компьютером, и его физических компонентов — аппаратного обеспечения. Тогда же и началось становление индустрии программного обеспечения, как самостоятельной отрасли. Первой компанией по разработке ПО стала основанная в 1959 году Роем Наттом и Флетчером Джоунсом Computer Sciences Corporation с

начальным капиталом в 100 долларов. Первыми клиентами CSC и появившихся вслед за нею софтверных компаний были сверхкрупные корпорации и государственные организации, вроде NASA, и фирма продолжала работать на рынке заказного ПО, как и другие первые программистские частные стартапы, такие как Computer Usage Company[10].

Первыми самостоятельно выпущенными программными продуктами, не поставляемыми в комплекте с компьютерным оборудованием, были выпущенный фирмой Applied Data Research в 1965 году генератор компьютерной документации AUTOFLOW, автоматически рисующий блок-схемы, и транслятор языка программирования MARK-IV, разработанный в 1960—1967 годах в Informatics, Inc. Становление рынка корпоративного программного обеспечения тесно связано с появлением семейства компьютеров IBM System/360. Достаточно массовые, относительно недорогие вычислительные машины, совместимые друг с другом на уровне программного кода, открыли дорогу тиражируемому программному обеспечению[11].

Постепенно круг заказчиков программного обеспечения расширялся, что стимулировало разработку новых видов программного обеспечения. Так появились первые фирмы, специализирующиеся на разработке систем автоматизированного проектирования.

В ноябре 1966 года журнал Business Week впервые обратился к теме индустрии программного обеспечения. Статья называлась «Software Gap — A Growing Crisis for Computers» и рассказывала как о перспективности этого бизнеса, так и о кризисе, связанном с нехваткой программистов. Типичные программные продукты того времени служили для автоматизации общих для бизнеса задач, таких, как начисление заработной платы или автоматизации бизнес-процессов таких предприятий среднего бизнеса, как производственное предприятие или коммерческий банк. Стоимость такого ПО, как правило, была между пятью и ста тысячами долларов[12].

Появление в 1970-х годах первых персональных компьютеров создало предпосылки и для зарождения массового рынка программного обеспечения. Изначально программы для персональных компьютеров распространялись в «коробочной» форме через торговые центры или по почте и имели цену 100—500 долларов США [13].

Знаковыми для зарождающегося массового рынка программного обеспечения стали такие продукты, как электронная таблица VisiCalc, идея которой пришла Дэниелу Бриклину, когда тот, будучи выпускником MIT и инженером-программистом в DEC, посещал курсы в Гарвардской школе бизнеса и хотел облегчить себе утомительные финансовые расчеты, и текстовый процессор WordStar, разработку которого начал Сеймур Рубинштейн, тщательно изучив потребности рынка. О VisiCalc впервые заговорили, как о killer application, то есть компьютерном приложении, которое самим фактом своего существования доказывает нужность и необходимость покупки платформы, для которой реализована такая программа. Для VisiCalc и WordStar такой платформой стали персональные компьютеры, которые благодаря им из богатой игрушки для гиков стали рабочим инструментом. С них началась микрокомпьютерная революция, а у этих программ появились конкуренты: электронные таблицы SuperCalc, Lotus 1-2-3, система управления базами данных dBase II, текстовый процессор WordPerfect и другие подобные. Текстовые процессоры, электронные таблицы, системы управления базами данных а так же графические редакторы вскоре стали основными продуктами рынка программного обеспечения для персональных компьютеров.

Массовое тиражирование позволило снизить к середине 1990 годов стоимость программного обеспечения для персональных компьютеров до ста — пятисот долларов, при этом бизнес производителей ПО приобрел определенное сходство с бизнесом звукозаписывающих компаний[14] [4, 7, 11].

2. Классификация программного обеспечения

2.1. Системное программное обеспечение

По назначению программы делят прикладные и системные.

В общем виде место системного программного обеспечения отображена на рисунке 1.

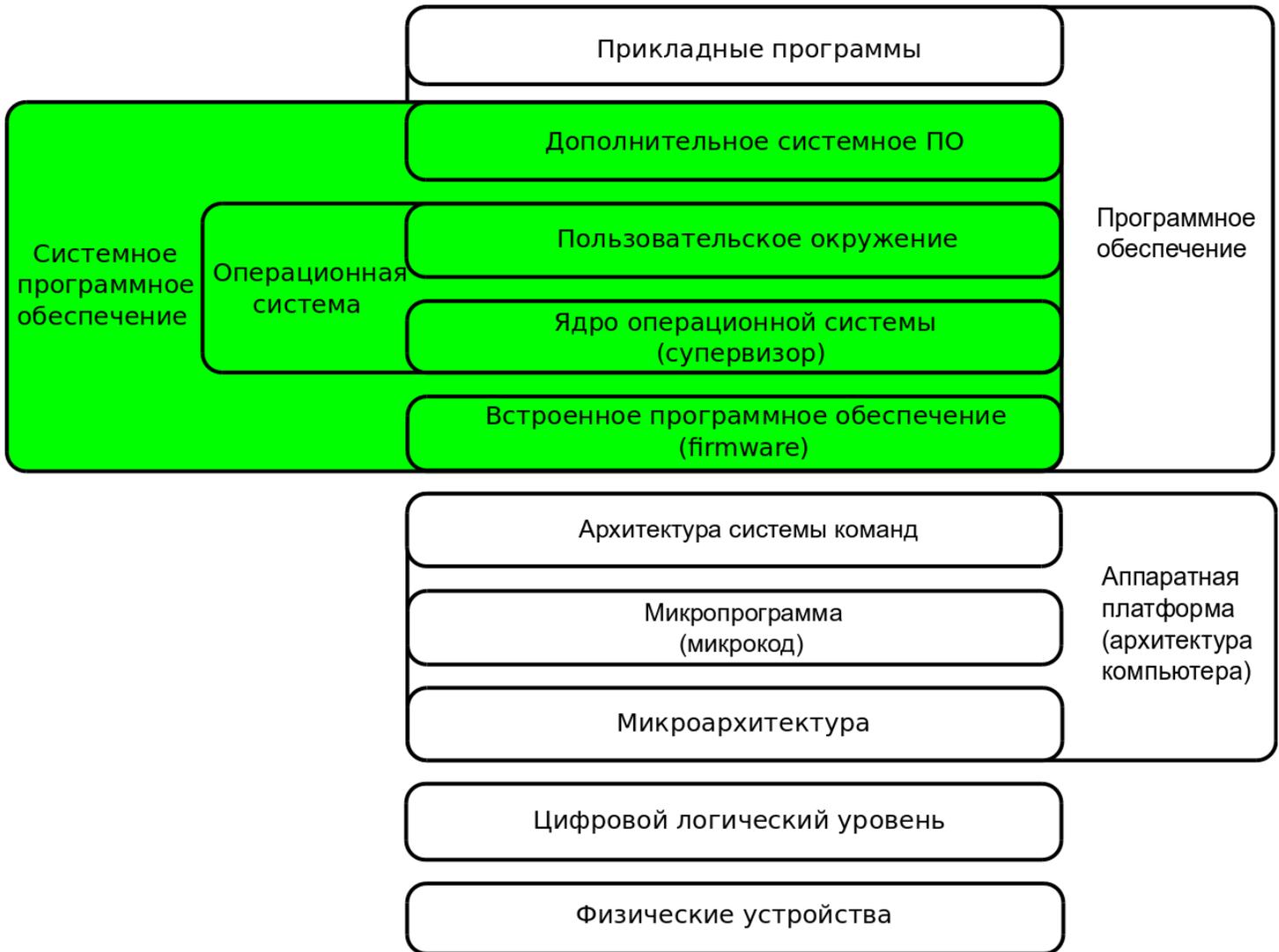


Рис. 1. Место системного программного обеспечения в многоуровневой схеме вычислительной системы

Системное программное обеспечение является комплексом программ, обеспечивающих управление компонентами компьютерной системы, такими как оперативная память, процессор, сетевое оборудование, устройства ввода-вывода, выступая в виде межслойного интерфейса, с одной стороны которого приложения пользователя, а с другой — аппаратное обеспечение. В отличие от прикладного программного обеспечения, системное не решает конкретные практические задачи, а только обеспечивает работу других программ, за счет предоставления им сервисных функций, которые абстрагируют детали микропрограммной и аппаратной реализации вычислительной системы, и управляет аппаратными ресурсами вычислительной системы[15].

Создание системного программного обеспечения называется системным программированием. Программист, который специализируется на системном программировании называется системным программистом.

Отнесение того или иного программного обеспечения к системному условно, и зависит от соглашений, которые используются в конкретном контексте. Обычно к системному программному обеспечению относятся утилиты, операционные системы управления базами данных, системы, системы программирования, широкий класс связующего программного обеспечения[\[16\]](#) [2, 5].

2.1.1. Операционная система

В основе системного программного обеспечения лежит операционная система, представляющая собой комплекс системных программ, который расширяет возможности вычислительной системы, а также обеспечивает управление ее ресурсами, выполнение и загрузку прикладных программ, а также взаимодействие с пользователями. В большинстве вычислительных систем операционные системы являются основной, наиболее важной частью системного программного обеспечения[\[17\]](#).

Основными функциями операционной системы как системного программного обеспечения являются:

- стандартизованный доступ к периферийным устройствам ввода-вывода;
- загрузка приложений в оперативную память и их выполнение;
- управление доступом к данным на энергонезависимых носителях, которые организованы в той или иной файловой системе;
- управление оперативной памятью, то есть распределение между процессами и виртуальная память;
- сетевые операции, поддержка стека протоколов;
- пользовательский интерфейс[\[18\]](#).

Существуют две группы определений операционных систем: «совокупность программ, управляющих оборудованием» и «совокупность программ, управляющих другими программами». Обе они имеют свой точный технический смысл, который, однако, становится ясен только при более детальном рассмотрении вопроса о том, зачем вообще нужны операционные системы[\[19\]](#).

Есть приложения вычислительной техники, для которых операционные системы излишни. Например, встроенные микрокомпьютеры содержатся сегодня во многих бытовых приборах, автомобилях, сотовых телефонах и т. п. Зачастую такой компьютер постоянно исполняет лишь одну программу, запускающуюся по включении. И простые игровые приставки — также представляющие собой специализированные микрокомпьютеры — могут обходиться без операционной системы, запуская при включении программу, записанную на вставленном в устройство «картридже» или компакт-диске. Тем не менее некоторые микрокомпьютеры и игровые приставки все же работают под управлением особых собственных операционных систем. В большинстве случаев, это UNIX-подобные системы.

Помимо операционных систем к системному программному обеспечению относят встроенное программное обеспечение и утилиты [\[20\]](#) [6, 8, 10].

2.1.2. Встроенное программное обеспечение

Встроенные программы называют также *firmware*. Они представляют собой программы, которые зашиты в цифровые электронные устройства. В ряде случаев они являются по сути частью операционной системы, которая хранится в постоянной памяти. В достаточно простых устройствах вся операционная система может быть встроенной. Многие устройства современных компьютеров имеют собственные прошивки, которые осуществляют управление этими устройствами и упрощают взаимодействие с ними [\[21\]](#).

Термином прошивка иногда называют образ посеянного запоминающего устройства, который предназначен для записи в память соответствующего устройства с целью обновления его микропрограммы, а также собственно процесс записи этого образа в энергонезависимую память устройства.

Прошивка памяти осуществляется при изготовлении устройства различными способами — например, установкой микросхемы памяти с записанным содержимым. Большинство устройств допускает замену содержимого памяти [\[22\]](#) [9, 12].

2.1.3. Утилиты

Утилитами называют программы, которые предназначены для решения узкого круга вспомогательных задач. Иногда утилиты относят к классу сервисного программного обеспечения[23].

Утилиты используются для мониторинга производительности оборудования и показателей датчиков, расширения возможностей, контроля показателей и управления параметрами оборудования.

Утилиты могут входить в состав операционных систем, идти в комплекте со специализированным оборудованием или распространяться отдельно[24].

В качестве типов утилит можно назвать утилиты работы с реестром, тесты оборудования, утилиты мониторинга оборудования и дисковые утилиты.

В качестве примеров дисковых утилит можно назвать:

- утилиты проверки диска, которые осуществляют поиск неправильно записанных либо поврежденных различным путем файлов и участков диска и их последующее удаление для эффективного использования дискового пространства;
- дефрагментаторы;
- утилиты разметки диска, которые осуществляют деление диска на логические диски, которые могут иметь различные файловые системы и восприниматься операционной системой как несколько различных дисков;
- утилиты очистки диска, которые осуществляют удаление временных файлов, ненужных файлов и чистку корзины;
- утилиты сжатия дисков, которые осуществляют сжатие информации на дисках для увеличения вместимости жестких дисков;
- утилиты резервного копирования, которые осуществляют создание резервных копий целых дисков и отдельных файлов, а также восстановление из этих копий[25] [13, 14].

2.2. Прикладное программное обеспечение

Прикладное программное обеспечение является комплексом программ, которые предназначены для выполнения определенных задач и рассчитаны на непосредственное взаимодействие с пользователем. В большинстве операционных систем прикладные программы не могут обращаться к ресурсам компьютера

напрямую, а взаимодействуют с оборудованием и другими программами посредством системного программного обеспечения.

К прикладному программному обеспечению относятся компьютерные программы, написанные для пользователей или самими пользователями для задания компьютеру конкретной работы. В качестве примера прикладного программного обеспечения можно привести программы обработки заказов или создания списков рассылки. Программистов, пишущих прикладное программное обеспечение, называют прикладными программистами[26].

По типу программного обеспечения различают программное обеспечение общего назначения, программное обеспечение развлекательного назначения, программное обеспечение специального назначения и профессиональное программное обеспечение[27].

По сфере применения выделяют:

- прикладное программное обеспечение предприятий и организаций. Например, финансовое управление, система отношений с потребителями, сеть поставок. К этому типу относится также ведомственное программное обеспечение предприятий малого бизнеса, а также программное обеспечение отдельных подразделений внутри большого предприятия;
- программное обеспечение, обеспечивающее доступ пользователя к устройствам компьютера;
- программное обеспечение инфраструктуры предприятия, обеспечивающее общие возможности для поддержки программного обеспечения предприятий. Это системы управления базами данных, серверы электронной почты, управление сетью и безопасностью;
- программное обеспечение информационного работника, обслуживающее потребности индивидуальных пользователей в создании и управлении информацией. Это, как правило, управление временем, ресурсами, документацией, например, текстовые редакторы, электронные таблицы, программы-клиенты для электронной почты и блогов, персональные информационные системы и медиаредакторы;
- программное обеспечение для доступа к контенту, используемое для доступа к тем или иным программам или ресурсам без их редактирования. Предназначено для групп или индивидуальных пользователей цифрового контента. Это, например, медиаплееры, веб-браузеры, вспомогательные браузеры и другие подобные[28];

- образовательное программное обеспечение по содержанию близко к программному обеспечению для медиа и развлечений, но, в отличие от него, имеет четкие требования по тестированию знаний пользователя и отслеживанию прогресса в изучении того или иного материала. Многие образовательные программы включают функции совместного пользования и многостороннего сотрудничества;
- имитационное программное обеспечение, используемое для симуляции физических или абстрактных систем в целях научных исследований, обучения или развлечения;
- инструментальные программные средства в области медиа, обеспечивающие потребности пользователей, которые производят печатные или электронные медиаресурсы для других потребителей, на коммерческой или образовательной основе. Это программы полиграфической обработки, верстки, обработки мультимедиа, редакторы HTML, редакторы цифровой анимации, цифрового звука и другие подобные;
- прикладные программы для проектирования и конструирования, используемые при разработке аппаратного и программного обеспечения. Охватывают автоматизированное проектирование, автоматизированную инженерию, редактирование и компилирование языков программирования, программы интегрированной среды разработки[29] [8, 14].

2.3. Другие классификации программного обеспечения

По степени тиражируемости все программное обеспечение делится на три такие категории, как:

- программное обеспечение, разрабатываемое на заказ;
- программное обеспечение для организаций и крупных корпораций;
- программное обеспечение для массового потребителя.

По степени переносимости программы делят на такие, как:

- кроссплатформенные;
- платформозависимые[30].

По способу распространения и использования программы делят на такие, как:

- свободные;
- открытые;
- несвободные, то есть закрытые[\[31\]](#).

По видам программы делят на такие, как:

- комплекс, когда программа состоит из двух или более комплексов и компонентов, которые выполняют взаимосвязанные функции, и применяется в составе другого комплекса или самостоятельно;
- компонент, когда программа рассматривается как единое целое, выполняет законченную функцию и применяется в составе комплекса или самостоятельно [\[32\]](#) [2, 4].

Заключение

В данной работе было рассмотрено понятие программного обеспечения с точки зрения многих стандартов. Согласно стандарту ISO/IEC 2382-1:1993, программное обеспечение представляет собой все или часть правил, процедур, программ и соответствующей документации системы обработки информации. Также была рассмотрена история развития программного обеспечения начиная с первых алгоритмов Ады Лавлейс и заканчивая массовым распространением программного обеспечения в 90-х годах 20 века.

Во второй части работы была рассмотрена классификация программного обеспечения, были выделены классификация по назначению, по степени тиражируемости, по степени переносимости, по способу распространения и использования и по видам.

Отдельное внимание было уделено классификация по назначению с выделением системного и прикладного программного обеспечения.

Список используемой литературы

1. Бутенко Д. В. Алгоритм проведения предпроектных исследований и моделирования информационных систем/ Д. В. Бутенко // Программные продукты и системы – М., 2013. – №4. – С. 53-56.

2. Затонский А. В. Информационные технологии: разработка информационных моделей и систем: Учебное пособие / А. В. Затонский – ИНФРА-М, 2014. – 344 с.
 3. Исаев Г. Проектирование информационных систем / Г. Исаев. — М.: Омега-Л, 2012. — 432 с.
 4. Бодров О. А. Предметно-ориентированные экономические информационные системы / О. А. Бодров, Р. Е. Медведев. — М.: Горячая линия - Телеком, 2013. — 244 с.
 5. Келим Ю. Вычислительная техника / Ю. Келим. — М.: Academia, 2013. — 368 с.
 6. Майерс Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер. — М.: «Диалектика», 2012. — 272 с.
 7. Ефимов И. Н. Качественные и количественные характеристики открытых информационных систем / И. Н. Ефимов // Программные продукты и системы – М., 2012. – №4. – С. 80-83.
 8. Мелехин В. Вычислительные машины / В. Мелехин, Е. Павловский. — М.: ДРОФА, 2013. — 384 с.
 9. Левин В. Информационные технологии в машиностроении / В. Левин. — М.: Academia, 2013. — 272 с.
 10. Бутенко Д. В. Методика концептуального проектирования программных информационных систем / Д. В. Бутенко // Программные продукты и системы – М., 2012. – №2. – С. 101.
 11. Марков А. С. Методы оценки несоответствия средств защиты информации / А. С. Марков, В. Л. Цирлов, А. В. Барабанов. - М.: Радио и связь, 2012. – 192 с.
 12. Мелехин В. Вычислительные системы и сети / В. Мелехин, Е. Павловский. — М.: Academia, 2013. — 208 с.
 13. Бородакий Ю. В. Эволюция информационных систем (современное состояние и перспективы) / Ю. В. Бородакий, Ю. Г. Лободинский. — М.: Горячая линия - Телеком, 2011. — 368 с.
 14. Емельянова Н. З. Проектирование информационных систем: Учебное пособие / Н. З. Емельянова, Т. Л. Партыка, И. И. Попов. - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с.
-
1. Бутенко Д. В. Алгоритм проведения предпроектных исследований и моделирования информационных систем/ Д. В. Бутенко // Программные продукты и системы – М., 2013. – №4. – С. 53-56. [↑](#)
 2. Исаев Г. Проектирование информационных систем / Г. Исаев. — М.: Омега-Л, 2012. — 432 с. [↑](#)

3. Бутенко Д. В. Алгоритм проведения предпроектных исследований и моделирования информационных систем/ Д. В. Бутенко // Программные продукты и системы – М., 2013. – №4. – С. 53-56. [↑](#)
4. Бодров О. А. Предметно-ориентированные экономические информационные системы / О. А. Бодров, Р. Е. Медведев. — М.: Горячая линия - Телеком, 2013. — 244 с. [↑](#)
5. Ефимов И. Н. Качественные и количественные характеристики открытых информационных систем / И. Н. Ефимов // Программные продукты и системы – М., 2012. – №4. – С. 80-83. [↑](#)
6. Марков А. С. Методы оценки несоответствия средств защиты информации / А. С. Марков, В. Л. Цирлов, А. В. Барабанов. - М.: Радио и связь, 2012. – 192 с. [↑](#)
7. Бодров О. А. Предметно-ориентированные экономические информационные системы / О. А. Бодров, Р. Е. Медведев. — М.: Горячая линия - Телеком, 2013. — 244 с. [↑](#)
8. Ефимов И. Н. Качественные и количественные характеристики открытых информационных систем / И. Н. Ефимов // Программные продукты и системы – М., 2012. – №4. – С. 80-83. [↑](#)
9. Марков А. С. Методы оценки несоответствия средств защиты информации / А. С. Марков, В. Л. Цирлов, А. В. Барабанов. - М.: Радио и связь, 2012. – 192 с. [↑](#)
10. Бодров О. А. Предметно-ориентированные экономические информационные системы / О. А. Бодров, Р. Е. Медведев. — М.: Горячая линия - Телеком, 2013. — 244 с. [↑](#)
11. Ефимов И. Н. Качественные и количественные характеристики открытых информационных систем / И. Н. Ефимов // Программные продукты и системы – М., 2012. – №4. – С. 80-83. [↑](#)

12. Марков А. С. Методы оценки несоответствия средств защиты информации / А. С. Марков, В. Л. Цирлов, А. В. Барабанов. - М.: Радио и связь, 2012. - 192 с. [↑](#)
13. Бодров О. А. Предметно-ориентированные экономические информационные системы / О. А. Бодров, Р. Е. Медведев. — М.: Горячая линия - Телеком, 2013. — 244 с. [↑](#)
14. Ефимов И. Н. Качественные и количественные характеристики открытых информационных систем / И. Н. Ефимов // Программные продукты и системы - М., 2012. - №4. - С. 80-83. [↑](#)
15. Затонский А. В. Информационные технологии: разработка информационных моделей и систем: Учебное пособие / А. В. Затонский - ИНФРА-М, 2014. - 344 с. [↑](#)
16. Келим Ю. Вычислительная техника / Ю. Келим. — М.: Academia, 2013. — 368 с. [↑](#)
17. Бутенко Д. В. Методика концептуального проектирования программных информационных систем / Д. В. Бутенко // Программные продукты и системы - М., 2012. - №2. - С. 101. [↑](#)
18. Майерс Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер. — М.:«Диалектика», 2012. — 272 с. [↑](#)
19. Бутенко Д. В. Методика концептуального проектирования программных информационных систем / Д. В. Бутенко // Программные продукты и системы - М., 2012. - №2. - С. 101. [↑](#)
20. Мелехин В. Вычислительные машины / В. Мелехин, Е. Павловский. — М.: ДРОФА, 2013. — 384 с. [↑](#)
21. Левин В. Информационные технологии в машиностроении / В. Левин. — М.: Academia, 2013. — 272 с. [↑](#)

22. Мелехин В. Вычислительные системы и сети / В. Мелехин, Е. Павловский. — М.: Academia, 2013. — 208 с. [↑](#)
23. Бородакий Ю. В. Эволюция информационных систем (современное состояние и перспективы) / Ю. В. Бородакий, Ю. Г. Лободинский. — М.: Горячая линия - Телеком, 2011. — 368 с. [↑](#)
24. Емельянова Н. З. Проектирование информационных систем: Учебное пособие / Н. З. Емельянова, Т. Л. Партыка, И. И. Попов. - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с. [↑](#)
25. Бородакий Ю. В. Эволюция информационных систем (современное состояние и перспективы) / Ю. В. Бородакий, Ю. Г. Лободинский. — М.: Горячая линия - Телеком, 2011. — 368 с. [↑](#)
26. Мелехин В. Вычислительные машины / В. Мелехин, Е. Павловский. — М.: ДРОФА, 2013. — 384 с. [↑](#)
27. Емельянова Н. З. Проектирование информационных систем: Учебное пособие / Н. З. Емельянова, Т. Л. Партыка, И. И. Попов. - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с. [↑](#)
28. Мелехин В. Вычислительные машины / В. Мелехин, Е. Павловский. — М.: ДРОФА, 2013. — 384 с. [↑](#)
29. Емельянова Н. З. Проектирование информационных систем: Учебное пособие / Н. З. Емельянова, Т. Л. Партыка, И. И. Попов. - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с. [↑](#)
30. Мелехин В. Вычислительные машины / В. Мелехин, Е. Павловский. — М.: ДРОФА, 2013. — 384 с. [↑](#)
31. Бодров О. А. Предметно-ориентированные экономические информационные системы / О. А. Бодров, Р. Е. Медведев. — М.: Горячая линия - Телеком, 2013. —

244 с. [↑](#)

32. Затонский А. В. Информационные технологии: разработка информационных моделей и систем: Учебное пособие / А. В. Затонский - ИНФРА-М, 2014. - 344 с.

[↑](#)