

Зрюмова А. Г., Зрюмов Е. А., Пронин С. П.

НАПИСАНИЕ на ЗАКАЗ:

1. Дипломы, курсовые, чертежи...
2. Диссертации и научные работы.
3. Школьные задания.

Онлайн-консультации.

ЛЮБАЯ тематика,
в том числе ТЕХНИКА.

Приглашаем авторов.

УЧЕБНИКИ, ДИПЛОМЫ, ДИССЕРТАЦИИ:

полные тексты в электронной библиотеке

www.учебники.информ2000.рф.

ИНФОРМАТИКА

Барнаул 2011

Министерство образования и науки Российской Федерации

ГОУ ВПО Алтайский государственный технический университет
им. И. И. Ползунова

А. Г. Зрюмова, Е. А. Зрюмов, С. П. Пронин

ИНФОРМАТИКА

Учебное пособие

Рекомендовано Алтайским государственным техническим университетом им. И. И. Ползунова в качестве учебного пособия для студентов АлтГТУ, обучающихся по направлению подготовки 200100 «Приборостроение»

Рекомендовано Учебно-методическим объединением вузов Российской Федерации по образованию в области приборостроения и оптоэлектроники для студентов высших учебных заведений, обучающихся по направлению подготовки 200100 – Приборостроение

Изд-во АлтГТУ
Барнаул 2011

УДК 681.3(075)

Зрюмова, А. Г. Информатика : учебное пособие / А. Г. Зрюмова, Е. А. Зрюмов, С. П. Пронин; Алт. гос. техн. ун-т им. И. И. Ползунова. – Барнаул : Изд-во АлтГТУ, 2011. – 177 с.

ISBN 978-5-7568-0843-8

Учебное пособие содержит необходимые теоретические сведения для изучения всех основных разделов информатики и практические примеры использования современных информационных технологий. По своему содержанию учебное пособие полностью соответствует требованиям государственных образовательных стандартов для технических направлений высшего профессионального образования и направлено на формирование профессиональных компетенций бакалавров.

Рекомендовано Алтайским государственным техническим университетом им. И. И. Ползунова в качестве учебного пособия для студентов АлтГТУ, обучающихся по направлению подготовки 200100 «Приборостроение». Протокол № 5 от 19 января 2011 г.

Рекомендовано Учебно-методическим объединением вузов Российской Федерации по образованию в области приборостроения и оптоэлектроники для студентов высших учебных заведений, обучающихся по направлению подготовки 200100 – Приборостроение. Протокол №1 от 08.02.2011 г.

Рецензенты: А. Г. Якунин, д. т. н., профессор, заведующий кафедрой «Вычислительные системы и информационная безопасность» Алтайского государственного технического университета им. И. И. Ползунова;

А. А. Веряев, д. п. н., профессор кафедры «Информационные технологии» Института физического и математического образования Алтайской государственной педагогической академии

ISBN 978-5-7568-0843-8

© Алтайский государственный технический университет
им. И. И. Ползунова, 2011

© Зрюмова А. Г., Зрюмов Е. А., Пронин С. П., 2011

Содержание

Предисловие	7
Введение в информатику	8
ГЛАВА 1	
Основные понятия информатики	9
1.1 Сигналы, данные, информация	9
1.2 Позиционные системы счисления	12
1.3 Единицы измерения количества информации	17
Тезаурус	18
Контрольные вопросы	19
Список рекомендуемой литературы	19
ГЛАВА 2	
Технические средства реализации информационных процессов	20
2.1 История развития информационных технологий	20
2.1.1 Первое поколение ЭВМ	21
2.1.2 Второе поколение ЭВМ	22
2.1.3 Третье поколение ЭВМ	22
2.1.4 Четвертое поколение ЭВМ	23
2.2 Архитектура компьютера	23
2.2.1 Архитектура с фиксированным набором устройств	23
2.2.2 Вычислительные системы с открытой архитектурой	24
2.2.3 Архитектура суперкомпьютеров	24
2.3 Классификация компьютеров	25
2.4 Состав персонального компьютера	27
2.4.1 Материнская плата	27
2.4.2 Микропроцессор	29
2.4.3 Оперативная память	35
2.4.4 Видеокарта	37
2.4.5 Устройства хранения данных	38
2.4.5.1 Жесткий диск	39
2.4.5.2 Оптические накопители данных	40
2.4.5.3 Голографическая память	43
2.4.5.4 Флэш-память	44
2.4.6 Устройства ввода данных	44
2.4.6.1 Клавиатура	44
2.4.6.2 Мышь	45
2.4.6.3 Сканер	46
2.4.6.4 Веб-камера	49
2.4.7 Устройства вывода данных	50
2.4.7.1 Монитор	50
2.4.7.1.1 Монитор на электронно-лучевой трубке	51

2.4.7.1.2 Жидкокристаллический монитор.....	52
2.4.7.1.3 Плазменный монитор.....	53
2.4.7.1.4 Органический светодиодный монитор.....	53
2.4.7.1.5 Стереоскопический монитор.....	55
2.4.7.2 Принтер.....	55
2.4.7.3 Плоттер.....	57
Тезаурус.....	58
Контрольные вопросы.....	58
Список рекомендуемой литературы.....	58
ГЛАВА 3	
Программные средства реализации информационных процессов.....	59
3.1 Понятие программного обеспечения.....	59
3.1.1 Базовый уровень программного обеспечения.....	59
3.1.2 Системный уровень программного обеспечения.....	60
3.1.3 Служебный уровень программного обеспечения.....	60
3.1.4 Прикладной уровень программного обеспечения.....	62
3.2 Операционные системы.....	63
3.2.1 Основные понятия операционных систем.....	63
3.2.2 Файловая система.....	65
3.2.2.1 Файловая система FAT.....	66
3.2.2.2 Файловая система NTFS.....	69
3.2.3 Классификация операционных систем.....	71
3.2.4 Обзор современных операционных систем.....	72
3.2.5 Принципы работы операционной системы Microsoft Windows XP.....	74
3.3 Обработка текста с помощью Microsoft Office Word 2007.....	77
3.4 Создание электронных таблиц с помощью Microsoft Office Excel 2007.....	82
3.5 Создание презентаций с помощью Microsoft Office PowerPoint 2007.....	86
3.6 Системы управления базами данных.....	90
3.6.1 Реляционная модель данных.....	91
3.6.1.1 Отношения.....	91
3.6.1.2 Ключи.....	92
3.6.1.3 Связывание таблиц.....	92
3.6.2 Нормализация реляционных баз данных.....	93
3.6.3 Денормализация реляционных баз данных.....	94
3.6.4 Разработка баз данных в Microsoft Office Access 2007.....	95
3.7 Технологии обработки графической информации.....	100
3.7.1 Растровая графика.....	101
3.7.2 Векторная графика.....	102
3.7.3 Цветовые модели.....	103
3.7.4 Обработка графической информации с помощью Adobe CS4.....	104

3.7.4.1 Программа обработки растровой графики Adobe Photoshop CS4.....	105
3.7.4.2 Программа обработки векторной графики Adobe Illustrator CS4.....	106
3.7.4.3 Программа разработки анимации Adobe Flash Professional CS4.....	106
Тезаурус.....	107
Контрольные вопросы.....	108
Список рекомендуемой литературы.....	108
ГЛАВА 4	
Моделирование функциональных и вычислительных задач.....	109
4.1 Моделирование как метод познания.....	109
4.2 Классификация и формы представления моделей.....	111
4.3 Методы и технологии моделирования.....	113
4.4 Информационная модель объекта.....	114
4.5 Математическое моделирование с помощью Mathcad.....	115
4.5.1 Выполнение арифметических операций.....	116
4.5.2 Решение систем линейных уравнений.....	117
4.5.3 Построение графиков функций.....	118
4.5.4 Работа с матрицами.....	120
4.5.5 Обработка изображений.....	121
4.5.6 Интегрирование и дифференцирование функций.....	123
Тезаурус.....	125
Контрольные вопросы.....	125
Список рекомендуемой литературы.....	125
ГЛАВА 5	
Компьютерные сети.....	126
5.1 Сетевые технологии обработки данных.....	126
5.2 Топологии компьютерных сетей.....	127
5.2.1 Топология «шина».....	128
5.2.2 Топология «звезда».....	128
5.2.3 Топология «кольцо».....	129
5.2.4 Ячеистая топология.....	130
5.3 Линии передачи данных в компьютерных сетях.....	131
5.3.1 Проводные линии передачи данных.....	131
5.3.1.1 Витая пара.....	132
5.3.1.2 Коаксиальный кабель.....	132
5.3.1.3 Оптоволоконный кабель.....	133
5.3.2 Беспроводные линии передачи данных.....	135
5.3.2.1 Инфракрасные линии передачи данных.....	136
5.3.2.2 Радиопередача данных.....	136
5.4 Сетевое оборудование.....	138

5.5 Эталонная модель OSI.....	140
5.6 Сетевые архитектуры.....	141
5.6.1 Сетевая архитектура Ethernet.....	141
5.6.2 Сетевая архитектура Token Ring.....	142
5.7 Принцип работы глобальной сети Интернет.....	142
5.7.1 Передача данных в глобальной сети Интернет.....	143
5.7.2 Службы глобальной сети Интернет.....	143
5.8 Информационная безопасность.....	145
5.8.1 Классификация вредоносных программ.....	146
5.8.2 Методы обеспечения информационной безопасности.....	149
5.8.3 Антивирусная программа Kaspersky Internet Security 2011.....	151
Тезаурус.....	151
Контрольные вопросы.....	152
Список рекомендуемой литературы.....	152
ГЛАВА 6	
Алгоритмизация и программирование.....	153
6.1 Понятие программы и алгоритма.....	153
6.2 Блок-схема алгоритма.....	154
6.3 Основные алгоритмические конструкции.....	155
6.4 Классификация языков программирования.....	158
6.4.1 Алгоритмическое программирование.....	160
6.4.2 Процедурное программирование.....	160
6.4.3 Объектно-ориентированное программирование.....	160
6.5 Обзор современных языков программирования.....	161
6.6 Введение в язык программирования Python.....	164
6.6.1 Типы данных Python.....	165
6.6.2 Ссылки на объекты Python.....	165
6.6.3 Коллекции данных Python.....	166
6.6.4 Инструкции управления Python.....	166
6.6.5 Использование функций Python.....	168
6.6.6 Объектно-ориентированная модель Python.....	168
6.6.7 Работа с графическими файлами.....	172
Тезаурус.....	174
Контрольные вопросы.....	175
Список рекомендуемой литературы.....	175
Алфавитный указатель.....	176

Предисловие

Учебное пособие посвящено изучению дисциплины «Информатика», которая прочно вошла в бытовую жизнь и профессиональную деятельность всех людей. Эта наука развивается наиболее динамично по сравнению с любой другой областью знаний, поэтому требует постоянного совершенствования знаний, умений и навыков от специалистов, работающих в данной отрасли. Учебное пособие предназначено для студентов направления «Приборостроение», а также для бакалавров и магистров, чья учебная деятельность непосредственно связана с информационными технологиями. Особенно интересен представленный материал будет студентам, которые занимаются проектированием и обслуживанием измерительных оптико-электронных систем, так как в пособии приведено множество примеров использования современных информационных технологий в оплотехнике.

Учебное пособие состоит из введения, шести глав и алфавитного указателя. После каждой главы читателю предлагается ответить на ряд вопросов, связанных с изученным материалом, а также рекомендуется список дополнительной литературы.

Глава 1 «Основные понятия информатики» посвящена рассмотрению общих вопросов, связанных с терминологией теории информации и кодирования, характеристикой процессов сбора, обработки, накопления и передачи информации.

Глава 2 «Технические средства реализации информационных процессов» направлена на изучение истории развития информационных технологий, предпосылок создания современной архитектуры компьютера, состава и назначения основных элементов персонального компьютера.

Глава 3 «Программные средства реализации информационных процессов» нацелена на изучение системного и служебного программного обеспечения, классификации операционных систем, методов обработки текстовой и графической информации, принципов работы с электронными таблицами, презентациями и базами данных.

Глава 4 «Моделирование функциональных и вычислительных задач» посвящена изучению вопросов, связанных с классификацией и формами представления моделей, методами и технологией моделирования, математическим моделированием оптических задач с помощью программы Mathcad.

Глава 5 «Компьютерные сети» направлена на изучение технических средств компьютерной коммуникации и методов защиты информации в локальных и глобальных компьютерных сетях, принципов функционирования основных сетевых технологий обработки данных, сервисов и стандартов.

Глава 6 «Алгоритмизация и программирование» нацелена на изучение базовых вопросов алгоритмизации и программирования, технологий структурного и объектно-ориентированного программирования, основных алгоритмических конструкций и операторов на примере языка программирования Python.

Введение в информатику

С давних времен люди работали с данными. Древнейшим инструментом для их обработки, который сама природа предоставила в распоряжение человека, была его собственная рука. Однако со временем увеличивалась потребность в создании более эффективных и производительных средств работы с информацией, стали появляться узелки и зарубки, абак и счеты, арифмометры и комптометры и, конечно же, компьютеры. Одновременно с этим разрабатывались теоретические основы новой науки, связанной с обработкой, хранением и передачей информации.

Слово «Информатика» происходит от французского слова «Informatique», образованного в результате объединения терминов «Informacion» (информация) и «Automatique» (автоматика), что выражает ее суть как науки об автоматической обработке информации. Кроме Франции термин «информатика» используется в ряде стран Восточной Европы. В то же время, в большинстве стран Западной Европы и США используется другой термин – «Computer Science» (наука о средствах вычислительной техники).

Сегодня информатика проникла практически во все сферы человеческой деятельности, трудно представить производство, строительство, энергетику, промышленность без компьютеризированных систем. Быт человека также неотрывно связан с персональным компьютером, мобильным телефоном, глобальной сетью Интернет, которые все большее влияние оказывают на жизнь и благосостояние человечества.

Информатика в собственной предметной области испытывает «информационный бум», какого не знает ни одна другая отрасль знаний. В среднем один раз в полтора года удваиваются основные технические параметры аппаратных средств, один раз в два-три года меняются поколения программного обеспечения, и один раз в пять-семь лет меняется база стандартов, интерфейсов и протоколов. В подтверждение этих слов один из ведущих ученых в области информатики Гордон Мур заметил, что если бы автомобилестроение развивалось такими же темпами, как компьютерная техника, то наши машины сейчас двигались бы со скоростью в полмиллиона миль в час и на одном галлоне бензина проезжали бы полмиллиона миль. Цена такой машины была бы такой низкой, что ее проще было бы выбросить на свалку, чем заплатить за один день парковки в Сан-Франциско.

Особенно хотелось бы отметить связь информатики с оптотехникой. Широкое внедрение систем технического зрения требует привлечения средств компьютерной обработки изображений, передачи видеоданных, восстановления и хранения визуальной информации, поэтому современному инженеру в области оптотехники необходимо знать основы информатики, уметь применять эти знания для построения высокотехнологичных оптико-электронных систем.

ГЛАВА 1 ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАТИКИ

В настоящей главе рассматриваются общие вопросы, связанные с терминологией теории информации и кодирования, характеристикой процессов сбора, обработки, накопления и передачи информации.

1.1 Сигналы, данные, информация

Ключевым понятием в информатике является термин «информация», который очень часто связан с такими понятиями, как сигналы и данные. Рассмотрим их более подробно.

Человек и все, что его окружает, благодаря различным видам энергообмена находится в постоянном взаимодействии с окружающим миром. Такое взаимодействие сопровождается появлением **сигналов**.

Сигнал имеет в своей основе материальную природу. Когда сигнал взаимодействует с физическим телом, то оно способно изменять какие-либо свои свойства. Эти изменения можно наблюдать, измерять, фиксировать, контролировать, то есть регистрировать сигналы. Зарегистрированные сигналы представляют собой **данные**. Данные несут в себе информацию о событиях, но не тождественны самой информации.

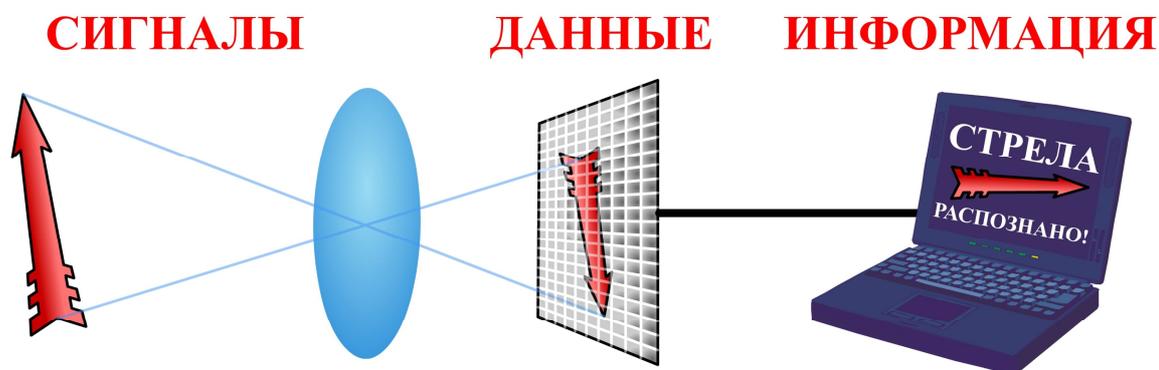


Рисунок 1.1 – Пример регистрации сигнала на фотоприемнике видеокамеры

Данные всегда должны быть отражены на каком-либо материальном носителе. Самым распространенным носителем данных в мире является бумага. В процессе развития информационных технологий появились и успешно развиваются такие носители данных, как лазерные диски, магнитные накопители. Сегодня активно применяются для регистрации сигналов видеокамеры (рисунок 1.1).

С точки зрения оптики отраженная от объекта световая волна видимого диапазона представляет собой сигнал, посланный объектом. Изображение объекта можно зарегистрировать на материальном носителе, например, фотопри-

емнике видеокамеры, при этом сигнал становится данными. Зарегистрированное изображение с помощью специальных методов на персональном компьютере можно распознать и сопоставить с определенным классом изображений, в этом случае мы имеем дело уже с информацией.

Из представленного примера видно, что к имеющимся данным необходимо применить методы, которые преобразуют данные в информацию. Методы, в свою очередь, тоже могут быть различными. Так, в рассмотренном примере, в качестве метода для распознавания изображений, могут использоваться системы искусственного интеллекта – нейронные сети.

Таким образом, **информация** представляет собой продукт взаимодействия данных и адекватных им методов. Термин «информация» происходит от латинского слова «informatio», что означает «сведения, разъяснения, изложение», то есть первоначальный смысл термина «информация» заключался в передаче каких-либо сведений. Рассматривая происхождение латинской основы термина in→forma-tio, становится ясным, что прежде чем передавать сведения, нужно, чтобы фрагмент окружающего мира нашел отражение в воспринимающем субъекте (in→), а результат этого отражения был зафиксирован в материальной форме (forma). Таким образом, существование информации без отражения в реальном мире невозможно.

При этом нельзя забывать о динамическом характере информации, который проявляется в том, что информация постоянно меняется и существует только в момент взаимодействия данных и методов. Также важным является применение к данным адекватных методов, так как одни и те же данные могут поставлять разную информацию в зависимости от степени адекватности взаимодействующих с ними методов.



Рисунок 1.2 – Свойства информации

В рамках информатики выделяют следующие свойства информации: дуализм, полнота, достоверность, адекватность, доступность, актуальность (рисунок 1.2). Рассмотрим эти свойства более подробно.

Дуализм информации проявляется в ее взаимодействии с методами. Так, субъективность информации является понятием относительным, потому что методы, применяемые для получения этой информации, являются субъективными. Отсюда следует, что более объективной будет та информация, на которую методы оказывают наименьшее значение. Например, если человек посмот-

рит на стрелу, то сразу же ее узнает, нейронная сеть же способна распознать изображение этого объекта, если обучена на большом количестве выборок.

Полнота информации характеризует качество информации и определяет достаточность данных для принятия решений или создания новых данных. Неполный набор данных оставляет большую долю неопределенности. Избыточный набор данных затрудняет доступ к нужным данным, создает повышенный информационный шум. Следовательно, неполный и избыточный наборы затрудняют получение информации и принятие адекватного решения. Например, если стрела находится в темноте, то результат распознавания будет неопределенным, если же стрелу дополнительно осветить внешним источником, то существует вероятность, что изображение будет засвеченным.

Достоверность информации определяет степень соответствия информации реальному объекту с необходимой точностью. При работе с неполным набором данных достоверность информации характеризуется вероятностью. Например, если часть изображения стрелы попадает в тень, то нельзя определенно сказать, что это за объект: стрела, или, например, меч.

Адекватность информации показывает степень соответствия создаваемого с помощью информации образа реальному объекту. Полную адекватность достигнуть очень трудно, если обрабатываются полностью достоверные данные, то методы, применяемые к ним, все равно будут неадекватны. Нейронные сети могут давать результат с определенной погрешностью, поэтому объект меч, по форме похожий на стрелу, может быть ошибочно причислен к этому классу.

Доступность информации выражается в возможности получить необходимую информацию в требуемый момент времени. На доступность информации влияют одновременно доступность данных и доступность адекватных методов для их интерпретации. Для того чтобы распознать изображение стрелы, необходимо наличие как фотоприемника видеокамеры, так и персонального компьютера.

Актуальность информации характеризует степень соответствия информации текущему моменту времени, так как все информационные процессы существуют во времени. Информация, актуальная в данный момент времени, может быть абсолютно бесполезна через некоторое время. Если объекты постоянно меняются, а результат распознавания образов остается неизменным, то такая информация неактуальна.

Получение информации связано с информационными процессами, в ходе которых данные преобразуются из одного вида в другой с помощью различных методов. Рассмотрим основные виды операций с данными.

Сбор данных представляет собой накопление информации с целью обеспечения достаточной полноты для принятия решений. **Формализация данных** используется для приведения данных, поступающих из разных источников, к одинаковой форме для повышения уровня их доступности. **Фильтрация данных** применяется для отсеивания данных, в которых нет необходимости для принятия решений; при этом должен уменьшаться уровень «шума», а достоверность и адекватность данных должны возрастать. **Сортировка данных** упоря-

дочивает данные по какому-либо признаку с целью удобства использования, что повышает доступность информации. **Архивация данных** необходима для организации хранения данных в удобной и легкодоступной форме; служит для снижения экономических затрат по хранению данных и повышает общую надежность информационного процесса в целом. **Защита данных** представляет собой комплекс мер, направленных на предотвращение утраты, воспроизведения и модификации данных. **Транспортировка данных** необходима при приеме и передаче данных между удаленными участниками информационного процесса. При этом источник данных в информатике принято называть сервером, а потребителя – клиентом. **Преобразование данных** используется для перевода данных из одной формы в другую или из одной структуры в другую.

Исходя из вышесказанного, можно дать следующее определение информатики. **Информатика** – это техническая наука, изучающая и систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.

Ключевым в определении информатики является понятие вычислительной техники. Под вычислительной техникой все чаще понимают **компьютер**, представляющий собой программируемое электронное устройство, которое предназначено для автоматизации создания, хранения, обработки и транспортировки данных.

В связи с тем, что информатика является практической наукой, можно выделить следующие определяющие направления для практических приложений: архитектура вычислительных систем, интерфейсы вычислительных систем, программирование, преобразование данных, защита информации, автоматизация и стандартизация программно-аппаратных средств.

1.2 Позиционные системы счисления

При работе с данными, относящимися к различным типам, очень важно унифицировать форму их представления – для этого обычно используется прием **кодирования**, то есть выражение данных одного типа через данные другого типа. В качестве примеров можно привести телеграфный код Морзе, морскую флажковую азбуку, систему Брайля для слабовидящих людей. Совокупность приемов кодирования и наименования чисел называется **системой счисления**.

Числа кодируются с помощью символов. В зависимости от количества используемых символов выделяют позиционные и непозиционные системы счисления. Если для кодирования числа используется бесконечное множество символов, то система счисления называется непозиционной, например, римская система счисления. Бесконечный ряд чисел потребует бесконечного числа символов для записи римских чисел. Кроме того, такой способ записи чисел приводит к очень сложным правилам арифметики.

Позиционные системы счисления для кодирования чисел используют ограниченный набор символов, называемых цифрами, и величина числа зависит

не только от набора цифр, но и от того, в какой последовательности они записаны. Количество цифр, используемых для записи числа, называется основанием системы счисления.

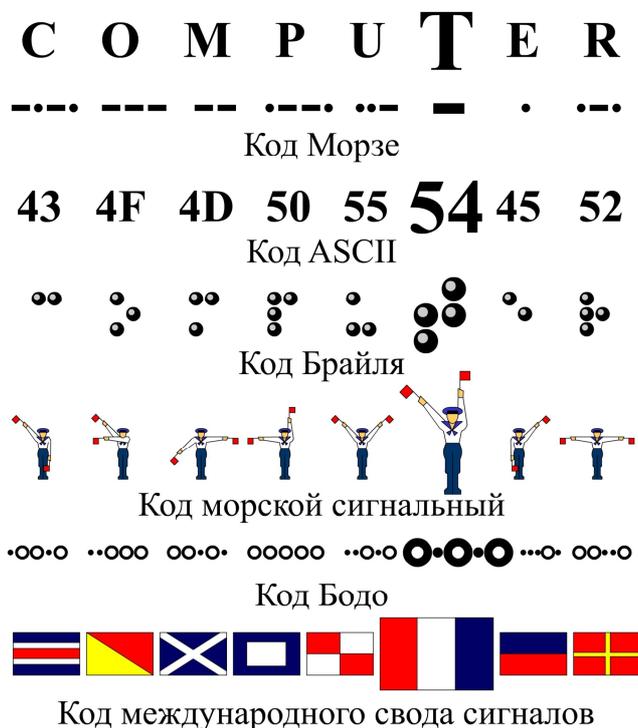


Рисунок 1.3 – Примеры кодирования слова «COMPUTER» в различных системах

Имена числительные во многих языках указывают на то, что у первобытного человека кодирование чисел осуществлялось преимущественно с помощью пальцев. Не случайно в древнерусской нумерации единицы называются «перстами», десятки – «составами», а все остальные числа – «сочинениями». По словам знаменитого русского путешественника Н. И. Миклухо-Маклая туземцы Новой Гвинеи считали так: «...папуас загибает один за другим пальцы руки, причем издает определенный звук, например, «бе, бе, бе...». Досчитав до пяти, он говорит «ибон-бе», что значит рука. Затем он загибает пальцы другой руки, снова повторяет «бе, бе...», пока не доходит до «ибон-али», что означает две руки. Затем он идет дальше, приговаривая «бе, бе...», пока не доходит до «самба-бе» и «самба-али», что означает одна нога и две ноги. Если нужно считать дальше, папуас пользуется пальцами рук и ног кого-нибудь другого».

От пальцевого счета берет начало пятеричная и десятеричная системы счисления. Именно эти системы счисления наиболее понятны человеку, потому что с рождения постигать счет приходится с помощью рук.

Однако существуют и другие системы счисления, успешно реализующиеся в отдельных отраслях техники, науки и культуры.

Томас Гэрриот впервые в мире подробно описал двоичную систему счисления, ставшую основной системой счисления, применяемой в компьютере.

Стостраничная рукопись начинается оставленной без комментариев таблицей, содержащей представления целых чисел вида 2^N с помощью знаков «+» и «-».

Вильгельм Готфрид Лейбниц также занимался исследованием свойств двоичной системы счисления. Он придавал ей мистический смысл и считал, что на ее базе можно создать универсальный язык для объяснения явлений мира и использования во всех науках, в том числе в философии.

Джордж Буль на основе двоичной системы счисления разработал раздел математической логики, занимающийся анализом высказываний, которые могут быть либо истинными, либо ложными. При этом высказывания должны строиться с помощью булевых функций, наиболее известными из которых являются конъюнкция («Логическое И»), дизъюнкция («Логическое ИЛИ»), отрицание («Логическое НЕ»), импликация («Следование»). Для того чтобы разобраться с этими функциями, построим таблицы истинности и диаграммы Эйлера, приведенные на рисунке 1.4.

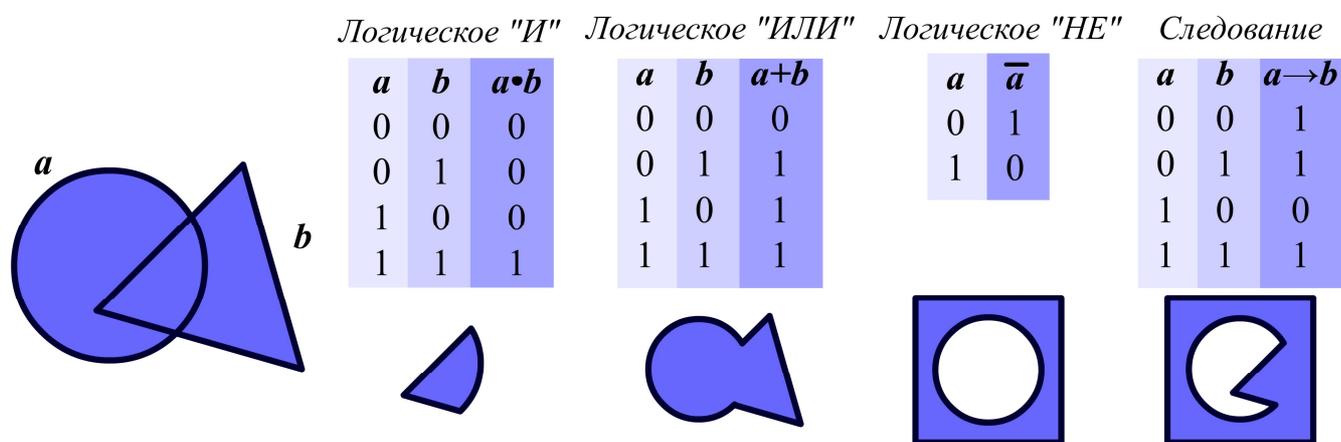


Рисунок 1.4 – Булевы функции

Прокомментируем каждую из этих булевых функций с помощью двух множеств *a* и *b*, которые могут принимать как истинные, так и ложные значения. Если над множествами *a* и *b* выполнить булеву функцию «Логическое И», то результат функции принимает истинное значение только тогда, когда оба значения множеств *a* и *b* истинные. Если над множествами *a* и *b* выполнить булеву функцию «Логическое ИЛИ», то результат функции принимает истинное значение только тогда, когда хотя бы одно значение множеств *a* и *b* истинное. Если над множеством *a* выполнить булеву функцию «Логическое НЕ», то результат функции принимает истинное значение тогда, когда значение множества *a* ложное, и результат функции принимает ложное значение тогда, когда значение множества *a* истинное. Если над множествами *a* и *b* выполнить булеву функцию «Следование», то результат функции принимает ложное значение только тогда, когда значение множества *a* истинное, а значение множества *b* ложное.

Двоичная система счисления получила широкое распространение в вычислительной технике из-за того, что имеет два устойчивых состояния: «0» и

«1», что соответствует отсутствию и наличию электрического сигнала. Чарльз Бэббидж одним из первых предложил использовать наряду с другими системами счисления систему счисления по основанию два в своей аналитической машине для счета, которая, к сожалению, не была реализована.

Следовательно, возникает парадоксальная ситуация, двоичная система счисления неудобна для восприятия человеком, но эффективно применяется в вычислительных машинах. Более привычна для пользователя десятичная система счисления, но работа с десятичными числами в вычислительных машинах громоздка и сложна.

Первым решил данную проблему Конрад Цузе при проектировании электромеханической вычислительной машины Z-1. Он разделил операции обработки данных и операции представления результатов вычислений. Для этого вычислительная машина Z-1 выполняла обработку вещественных чисел, представленных в двоичной системе счисления, а входные и выходные устройства машины преобразовывали числа из десятичной системы счисления в двоичную систему и обратно.

Пример перевода числа из десятичной системы счисления в двоичную систему и обратно представлен на рисунке 1.5.

$$\begin{array}{r}
 39 \mid 2 \\
 \hline
 19 \mid 2 \\
 \hline
 9 \mid 2 \\
 \hline
 4 \mid 2 \\
 \hline
 2 \mid 2 \\
 \hline
 1 \mid 2 \\
 \hline
 0
 \end{array}$$

$39_{10} = 100111_2$

$$\begin{aligned}
 & 100111_2 = 2^5 \cdot 1 + 2^4 \cdot 0 + 2^3 \cdot 0 + 2^2 \cdot 1 + \\
 & + 2^1 \cdot 1 + 2^0 \cdot 1 = 32 + 4 + 2 + 1 = 39_{10}
 \end{aligned}$$

Рисунок 1.5 – Пример преобразования из десятичной системы счисления в двоичную систему и обратно

Целые десятичные числа кодируются двоичным кодом. Для этого нужно взять целое число и делить его пополам до тех пор, пока в остатке не образуется нуль или единица. Совокупность остатков от каждого деления, записанная слева направо, начиная с последнего остатка, и образует двоичный аналог десятичного числа.

Если необходимо двоичные числа закодировать десятичными, то необходимо найти сумму каждого значащего разряда двоичного числа, умноженного на число два в степени, равной порядковому номеру разряда, начиная с нуля.

Можно заметить, что, несмотря на многофункциональность двоичного кода, его запись очень громоздка. Для устранения этого недостатка применяют восьмеричную и шестнадцатеричную системы счисления.

Восьмеричная система счисления имеет восемь знаков: 0 1 2 3 4 5 6 7. Шестнадцатеричная система счисления использует для представления чисел шестнадцать знаков: 0 1 2 3 4 5 6 7 8 9 A B C D E F.

"10"	"2"	"8"	"16"
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Рисунок 1.6 – Представление чисел в различных системах счисления

Перевод из двоичной системы счисления в восьмеричную систему осуществляется очень просто. Все разряды двоичного числа разделяются на триады, начиная справа, причем каждой триаде соответствует определенный знак восьмеричной системы.

Перевод из двоичной системы счисления в шестнадцатеричную систему осуществляется аналогично. Все разряды двоичного числа разделяются уже на четверки, начиная справа, причем каждой четверке соответствует определенный знак шестнадцатеричной системы. Сопоставление чисел двоичной системы счисления числам восьмеричной, десятичной и шестнадцатеричной систем счисления представлено на рисунке 1.6. На рисунке 1.7 представлен пример перевода двоичного числа в восьмеричную и шестнадцатеричную систему счисления.

$$\underbrace{100111011011001}_2 = 47331_8$$

$$\underbrace{0100111011011001}_2 = 4ED9_{16}$$

Рисунок 1.7 – Перевод двоичного числа в восьмеричную и шестнадцатеричную систему счисления

Для преобразования десятичного числа в восьмеричную или шестнадцатеричную систему счисления необходимо сначала перевести десятичное число в двоичную форму, а затем сделать преобразование двоичного числа в восьмеричную или шестнадцатеричную систему. Аналогично осуществляется перевод восьмеричного или шестнадцатеричного числа в десятичную систему счисления.

1.3 Единицы измерения количества информации

Наибольший вклад в теорию связи внес Клод Шеннон, занимавшийся изучением способов наилучшей передачи информации по телефонным каналам связи, находящимся под воздействием помех. В 1948 г. Клод Шеннон показал, что для этого необходимо предварительно преобразовать передаваемый аналоговый сигнал в цифровой код. Чтобы математически оценить количество передаваемой информации по каналам связи, ученый принял за единицу информации бит и определил ее количество через энтропию, которая характеризует количественную меру неопределенности ситуации:

$$I = 1 - H = 1 - \sum_{i=1}^N p_i \cdot \log_2 \frac{1}{p_i},$$

где I – количество информации, H – энтропия сообщения, N – количество сообщений, i – номер сообщения, p_i – вероятность i -го сообщения.

Следовательно, по Шеннону, количеством информации называют числовую характеристику информации, отражающую ту степень неопределенности, которая исчезает после получения сообщения.

С тех пор за единицу информации принимается один **бит** (**bit** – **binary digit** – двоичная цифра). Это количество информации, при котором неопределенность уменьшается вдвое. Бит является самой маленькой единицей измерения информации. На практике чаще применяются более крупные единицы, например, байт. **Байтом** называется последовательность из восьми бит. Одного байта достаточно для того, чтобы закодировать все клавиши клавиатуры компьютера. Один байт также является минимальной единицей адресуемой памяти компьютера.

Широко используются еще более крупные производные единицы измерения информации: килобайт, мегабайт, гигабайт, терабайт. Основные соотношения между этими единицами измерения представлены на рисунке 1.8.

1 Килобайт (Кбайт)	1024 байт	2^{10} байт
1 Мегабайт (Мбайт)	1024 Кбайт	2^{20} байт
1 Гигабайт (Гбайт)	1024 Мбайт	2^{30} байт
1 Терабайт (Тбайт)	1024 Гбайт	2^{40} байт
1 Петабайт (Пбайт)	1024 Тбайт	2^{50} байт
1 Экзабайт (Эбайт)	1024 Пбайт	2^{60} байт
1 Зеттабайт (Збайт)	1024 Эбайт	2^{70} байт
1 Йоттабайт (Йбайт)	1024 Збайт	2^{80} байт

Рисунок 1.8 – Основные соотношения между единицами измерения информации

Необходимо отметить, что по мере усовершенствования вычислительной техники появляются все новые и новые единицы измерения информации, отражающие прогресс компьютерных технологий в основном в сфере разработки устройств хранения данных.

Говоря о единицах измерения информации, нельзя не отметить характеристику, отражающую скорость прохождения битов информации, которая называется **битрейт**. Понятие битрейт используют при измерении эффективной скорости передачи информации по каналу связи, которая выражается в количестве битов, переданных в секунду. При этом не учитывается служебная информация: стартовые и стоповые биты, контрольные символы.

Скорость передачи информации, учитывающую полную пропускную способность канала, выражают в бодах. **Бодом** называется единица скорости передачи информации, равная одному биту в секунду. Для обозначения больших скоростей передачи применяют более крупные единицы: килобиты в секунду, мегабиты в секунду, гигабиты в секунду.

Тезаурус

Сигнал, данные, сообщение, информация, информатика, кодирование, формализация данных, фильтрация данных, сортировка данных, архивация данных, защита данных, транспортировка данных, преобразование данных, система счисления, двоичная система счисления, восьмеричная система счисления, десятичная система счисления, шестнадцатеричная система счисления, конъюнкция, дизъюнкция, отрицание, импликация, энтропия, бит, байт, килобайт, мегабайт, гигабайт, терабайт, петабайт, экзабайт, зеттабайт, йоттабайт, битрейт, бод.

Контрольные вопросы

1. Дайте определение термину «информация».
2. Назовите существующие свойства информации.
3. Перечислите основные виды операций с данными и поясните их смысл на конкретном примере.
4. В чем отличие данных от информации?
5. Сколько бит в мегабайте?
6. Почему двоичная система счисления получила широкое распространение в вычислительной технике?
7. Переведите десятичное число 145_{10} в восьмеричную и шестнадцатеричную системы счисления.
8. Назовите таблицы истинности булевых функций конъюнкция, дизъюнкция, отрицание и импликация.
9. Каким образом энтропия связана с количеством информации по определению Шеннона?
10. В каких случаях для измерения скорости передачи информации по каналу связи используют единицу измерения бод?

Список рекомендуемой литературы

1. Воройский, Ф. С. Информатика : энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах [Текст] / Ф. С. Воройский. – М. : ФИЗМАТЛИТ, 2006. – 768 с.
2. Информатика : базовый курс [Текст] / под ред. С. В. Симоновича. – СПб. : Питер, 2009. – 640 с.
3. Конев, Ф. Б. Информатика для инженеров [Текст] / Ф. Б. Конев. – М. : Высшая школа, 2004. – 272 с.
4. Могилев, А. В. Информатика [Текст] / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. – М. : Академия, 2004. – 848 с.
5. Полунов, Ю. Л. От абака до компьютера: судьбы людей и машин : книга для чтения по истории вычислительной техники [Текст] / Ю. Л. Полунов. – М. : Русская редакция, 2004. – 480 с. – т. 1.
6. Соболев, Б. В. Информатика [Текст] / Б. В. Соболев, А. Б. Галин, Ю. В. Панов. – Ростов н/Д : Феникс, 2007. – 446 с.
7. Степанов, А. Н. Информатика [Текст] / А. Н. Степанов. – СПб. : Питер, 2006. – 684 с.

ГЛАВА 2

ТЕХНИЧЕСКИЕ СРЕДСТВА РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

В настоящей главе рассматривается история развития информационных технологий, предпосылки создания современной архитектуры вычислительной техники, дается характеристика составу и назначению основных элементов персонального компьютера.

2.1 История развития информационных технологий

Изобретение новых вычислительных машин во все времена определялось потребностями общества, экономическими условиями и технологическими предпосылками. В определенное время человек обходился пальцами руки, чтобы сделать все необходимые вычисления, затем достаточно было абаков и логарифмических линеек для получения результата заданной точности и производительности. Однако интенсивно начали развиваться технологии создания вычислительных машин после появления в 1645 г. «Арифметической машины» Блеза Паскаля или, как называли ее современники, «Паскалево колесо», так как основным элементом была шестеренка.

Многие ученые и инженеры усовершенствовали конструкцию такой машины. Среди них выделяется Чарльз Бэббидж, великий математик и физик, известный также своими открытиями в оптике: он изобрел коронограф и сконструировал офтальмоскоп. Чарльз Бэббидж задался идеей создания «Аналитической машины», способной выполнять операции не только над числами, но и над аналитическими выражениями, для этого он в 1837 г. сформулировал несколько основополагающих принципов, которые стали поворотными в развитии цифровых вычислительных машин.

Во-первых, Чарльз Бэббидж предложил отделить устройство для хранения чисел, которое назвал «склад», от устройства, предназначенного для выполнения арифметических операций, которое назвал «мельница». Во-вторых, Бэббидж выделил устройства ввода и вывода данных, а также устройство управления работой машины. В-третьих, он предложил использовать в машине системы счисления по основанию два, три, пять, десять и сто. В-четвертых, Чарльз Бэббидж высказал соображения, чрезвычайно важные для современного программирования: предложил идеи «цикла», «библиотеки подпрограмм», операции «условной передачи управления».

Механический принцип построения устройств и использование десятичной системы счисления не позволили Чарльзу Бэббиджу полностью реализовать свои идеи, ему пришлось ограничиться скромными макетами, иначе по размерам машина сравнялась бы с локомотивом, и чтобы привести в движение ее устройства, понадобился бы паровой двигатель.

Реализовать идеи Бэббиджа на новой элементной базе первым смог немецкий инженер Конрад Цузе. В 1941 г. для построения вычислительной ма-

шины Z-3 он использовал в качестве логического элемента, работающего по принципу «да/нет», телефонное реле. Именно реле позволило реализовать в электромеханической вычислительной машине двоичную логику, что значительно упростило конструкцию машины и увеличило ее быстродействие.

Аналогичного результата в 1943 г. добился Говард Эйкен, сконструировав релейную вычислительную машину Mark-I. В структурном отношении эти вычислительные машины повторяли «Аналитическую машину», поэтому сам Говард Эйкен честно признавался: «Живи Бэббидж на 75 лет позже, я остался бы безработным». Однако электромеханические машины Z-3 и Mark-I подтвердили правильность сделанных выводов и создали основу для построения нескольких поколений цифровых электронных вычислительных машин.

Новые логические элементы раскрывали все новые возможности перед разработчиками ЭВМ; появление электронных ламп, транзисторов, интегральных схем предопределяло рост производительности и мощности вычислительных систем, но идеи, высказанные Бэббиджем, остаются актуальными и сегодня. Рассмотрим каждое поколение ЭВМ более подробно.

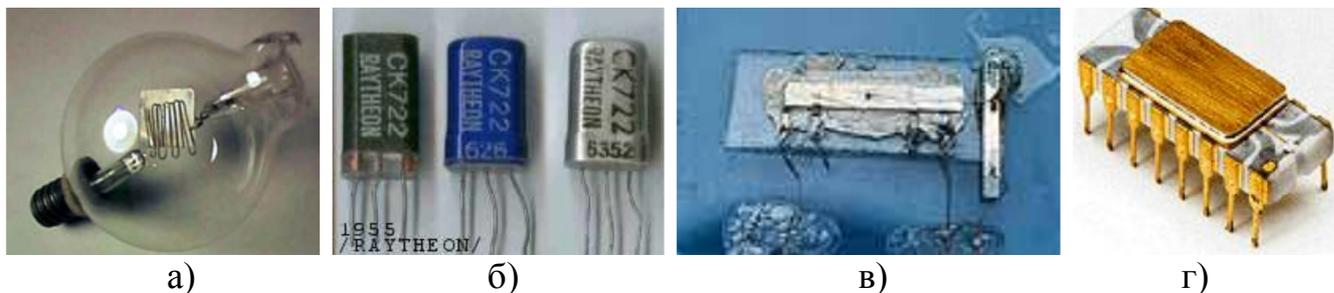


Рисунок 2.1 – Фотографии элементов электронных вычислительных машин:
а) одна из первых ламп Ли де Фореста; б) первые транзисторы;
в) интегральная схема Джека Килби; г) процессор Intel 4004

2.1.1 Первое поколение ЭВМ

Основным элементом первого поколения ЭВМ стала изобретенная американским инженером Ли Де Форестом **электронная лампа** (рисунок 2.1, а).

Первая электронная вычислительная машина была создана в 1945 г. под руководством Джона Моучли и Джона Эккерта и называлась «**ENIAC**» (Electronic Numerical Integrator and Computer – электронный числовой интегратор и калькулятор). «ENIAC» содержал 178468 электронных ламп шести различных типов, занимал площадь в 300 м² и в 1000 раз превосходил по быстродействию релейные вычислительные машины. Управление таких ЭВМ осуществлялось с помощью пульта управления, ввод информации производился с помощью перфокарт, программирование выполнялось посредством машинных языков, быстродействие таких систем составляло десятки тысяч операций в секунду.

Однако электронные лампы занимали много места, для реализации узлов вычислительной машины их требовалось огромное количество, работали они с напряжениями в десятки вольт и расходовали много энергии. Это приводило к перегреву элементов и выходу из строя вычислительной машины.

В 1946 г. Джон фон Нейман на основе критического анализа конструкции «ENIAC» предложил ряд новых идей организации ЭВМ с хранимой в памяти программой, в результате чего была создана архитектура ЭВМ, в целом сохранившаяся до настоящего времени.

2.1.2 Второе поколение ЭВМ

Появление второго поколения ЭВМ обязано изобретению в 1948 г. полупроводникового прибора – транзистора (рисунок 2.1, б). Изобретение транзисторов явилось знаменательной вехой в истории развития электроники, и его авторы Джон Бардин, Уолтер Бремен и Уильям Брэдфорд Шокли были удостоены в 1956 г. Нобелевской премии по физике.

Транзисторы отличались от электронных ламп малыми размерами, низким напряжением питания и малой потребляемой мощностью, что обусловило появление компактных компьютеров с большими функциональными возможностями.

Первый транзисторный компьютер «Традис» был сконструирован фирмой «Белл телефон лабораторис» и содержал 800 транзисторов. Характерной особенностью ЭВМ третьего поколения является применение в качестве устройств ввода информации перфолент, устройств хранения информации – магнитных барабанов. Программирование таких компьютеров осуществлялось с помощью алгоритмических языков программирования, быстрдействие таких систем составляло сотни тысяч операций в секунду.

2.1.3 Третье поколение ЭВМ

В транзисторном компьютере каждый компонент электронной схемы изготавливался отдельно, а затем компоненты соединялись посредством пайки. Такая конструкция была громоздкой и потребляла много энергии. Разрешил эту проблему в 1956 г. Джек Килби, разместивший на одном полупроводниковом кристалле все компоненты одного функционального узла (рисунок 2.1, в). Так появились интегральные схемы.

Первыми представителями ЭВМ третьего поколения стали компьютеры фирмы IBM – «System 360». Управление таких систем производилось с помощью операционных систем, ввод и вывод информации осуществлялись посредством видеотерминальных систем. Производительность компьютеров увеличилась до миллионов операций в секунду. Хотя технология производства интегральных схем стала доступной, цена компьютеров на их основе была высокой и недоступной для массового потребителя.

2.1.4 Четвертое поколение ЭВМ

Четвертое поколение ЭВМ связано с появлением технологии больших интегральных схем, при которой на одном кристалле размещалось более тысячи компонентов. Это привело к миниатюризации электронных устройств и появлению новой отрасли промышленности – микроэлектроники. В 1971 г. Эдвард Хофф, работающий в фирме «Intel», построил интегральную схему, аналогичную по своим функциям центральному процессору большой ЭВМ. Так появился первый микропроцессор Intel-4004, содержащий все основные компоненты центрального процессора (рисунок 2.1, г).

Появление этого процессора дало импульс массовому производству компьютеров: «Altair 8800» и «Apple-1». Привлекательность этих компьютеров для пользователей заключалась в том, что это было готовое к использованию устройство, содержащее все необходимое оборудование и программное обеспечение. В 1978 г. появился процессор Intel 8086, на основе которого IBM начала выпуск первых в мире персональных компьютеров. Производительность ЭВМ четвертого поколения превышает 10 миллионов операций в секунду.

Дальнейшая классификация вычислительных систем по их принадлежности к различным поколениям весьма условна. В настоящее время элементная база микропроцессорных систем активно развивается, но в ее основе по-прежнему лежат сверхбольшие интегральные схемы.

2.2 Архитектура компьютера

Современный компьютер состоит из многих функциональных узлов, каждый из которых представляет собой сложное электронное устройство. Набор компьютерных устройств, операций и связей между ними называется **архитектурой компьютера**. Рассмотрим более подробно существующие архитектуры ЭВМ.

2.2.1 Архитектура с фиксированным набором устройств

Архитектурой с фиксированным набором устройств называется такая организация компьютера, при которой устройства обработки данных расположены компактно и используют для обмена информацией внутренние шины передачи данных.

Компьютеры первого и второго поколения имели архитектуру с фиксированным набором устройств, так как введение любого дополнительного устройства требовало разработки нестандартных средств его подключения и вызывало увеличение потребления энергии, поэтому такие компьютеры использовались, в основном, при решении узкоспециализированных вычислительных задач. Нужно отметить, что такая архитектура в некоторых случаях применяется и сегодня, например, для построения печатных плат приборов, не требующих подключения дополнительных устройств.

2.2.2 Вычислительные системы с открытой архитектурой

В основе современного персонального компьютера лежит открытая архитектура – способ построения, регламентирующий и стандартизирующий только описание принципа действия компьютера и его конфигурации, что позволяет собирать его из отдельных узлов и деталей, разработанных и изготовленных независимыми производителями. Главным нововведением являлось подключение всех устройств, независимо от их назначения, к общей шине передачи информации.

Подключение устройств к шине осуществляется в соответствии со стандартом шины, который является свободно распространяемым документом.

Для того чтобы соединить друг с другом различные устройства компьютера, они должны иметь одинаковый интерфейс. **Интерфейсом** называется средство сопряжения двух устройств, в котором все физические и логические параметры согласуются между собой. Если интерфейс является общепринятым, например, утвержденным на уровне международных соглашений, то он называется стандартным. Для согласования интерфейсов периферийные устройства подключаются к шине не напрямую, а через контроллеры. **Контроллер** согласовывает сигналы устройства с сигналами шины и осуществляет его управление по командам, поступающим от центрального процессора. Контроллер подключается к шине специальными устройствами – **портами ввода-вывода**. **Протокол** устанавливает технические условия, которые должны быть обеспечены разработчиками устройств для успешного согласования их работы с другими устройствами.

2.2.3 Архитектура суперкомпьютеров

Под **суперкомпьютером** понимают мощную вычислительную систему с производительностью свыше 100 миллионов операций с плавающей точкой в секунду. Для измерения производительности работы суперкомпьютеров используется единица **Флопс** (Floating point Operations Per Second), показывающая, сколько операций с плавающей запятой в секунду выполняет вычислительная система.

Традиционной сферой применения суперкомпьютеров всегда были научные исследования: экспериментальная физика, химия, математика, также суперкомпьютеры активно используются для достижения военных целей: конструирование ракет, самолетов, танков и подводных лодок. Однако в последнее время вычислительные мощности суперкомпьютеров требуются для решения задач повседневной жизни: реализации специальных эффектов в графических приложениях для кино и телевидения, построения сверхбольших баз данных, предполагающих интенсивную обработку транзакций.

Выделяют следующие основные архитектуры суперкомпьютеров. **Асимметричная мультипроцессорная обработка** – архитектура суперкомпьютера, в которой каждый процессор имеет собственную оперативную память, при этом

процессоры взаимодействуют между собой, передавая друг другу сообщения через общую шину или с использованием межпроцессорных связей. **Симметричная мультипроцессорная обработка** – архитектура суперкомпьютера, в которой группа процессоров работает с общей оперативной памятью и пользуется общими внешними устройствами, при этом операционная система и приложения могут использовать любой доступный процессор. Две представленные архитектуры позволяют строить высокопроизводительные вычислительные системы, но имеют существенный недостаток – высокую стоимость.

Поэтому на сегодня самой популярной архитектурой суперкомпьютеров являются кластеры. **Кластер** представляет собой набор из нескольких компьютеров, соединенных через некоторую коммуникационную сеть. В качестве такой структуры может выступать обычная компьютерная сеть. Самый большой недостаток, по сравнению с рассмотренными архитектурами суперкомпьютеров, это невысокая скорость связи и большая латентность коммуникационной сети, что заставляет разработчиков создавать новые высокоскоростные интерфейсы.

Специально для учета суперкомпьютеров был учрежден проект TOP-500, основной задачей которого является составление рейтинга и описаний самых мощных ЭВМ мира.

2.3 Классификация компьютеров

После изобретения микропроцессора Intel 4004, имеющего малые габаритные размеры и невысокую стоимость, началось бурное развитие персональных компьютеров, предназначенных для обслуживания одного рабочего места. Сферы применения персональных компьютеров настолько разнообразны, что в 1999 г. пришлось разрабатывать международные сертификационные стандарты PC99, регламентирующие принципы их классификации (рисунок 2.2).

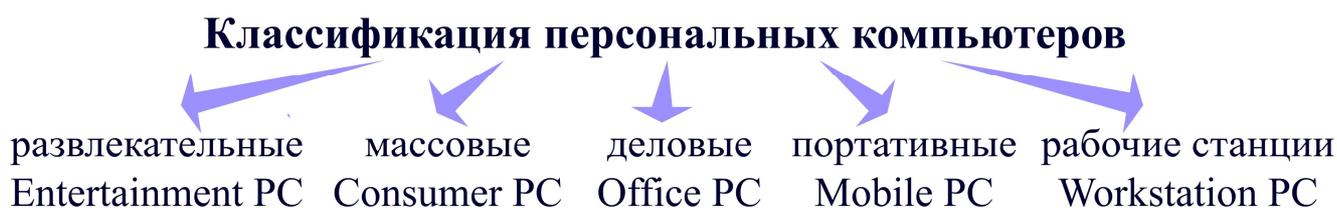


Рисунок 2.2 – Классификация персональных компьютеров

Каждая категория персональных компьютеров имеет свои особенности.

Развлекательные компьютеры должны иметь мощные средства обработки графической информации и звука. Такие компьютеры в основном используются для воспроизведения видеоинформации и запуска компьютерных игр.

Деловые компьютеры предназначены для решения определенных деловых задач. Компьютеры такого класса необходимы в основном офисным работникам, чтобы составлять отчеты, презентации, отправлять деловую корреспонденцию и работать с бухгалтерскими программами. Требования к таким персо-

нальным компьютерам выдвигаются весьма скромные, кроме обязательной поддержки работы в локальной и глобальной сети.

Массовые компьютеры являются компромиссным вариантом между развлекательным и деловым компьютером, связано это с тем, что многие персональные компьютеры становятся многофункциональными, способными решать множество самых разнообразных задач. Массовые компьютеры предназначены для среднестатистического пользователя, который может работать с большим количеством стандартных прикладных программ.

Портативные персональные компьютеры пользуются сегодня большим спросом, так как позволяют мобильно решать многие проблемы, при этом для такого персонального компьютера трудно определить какое-то одно рабочее место. Важным требованием, выдвигаемым к портативным компьютерам, является низкий вес, возможность длительной работы от аккумулятора, наличие средств связи и выхода в глобальную сеть Интернет.

Рабочие станции являются категорией персональных компьютеров, которые ориентированы на решение специфических задач, требующих выполнения значительного объема вычислений. К таким компьютерам предъявляются повышенные требования к производительности микропроцессора, объему оперативной памяти и устройствам хранения данных.

Одна из целей классификации персональных компьютеров состояла и в том, чтобы наметить пути дальнейшего их развития и совершенствования. Однако постепенно границы между разными категориями персональных компьютеров размылись, поэтому обновление этих стандартов было прекращено.



Рисунок 2.3 – Классификация персональных компьютеров по типоразмерам

Персональные компьютеры также можно классифицировать по типоразмерам. Различают настольные, портативные и карманные модели (рисунок 2.3).

Настольные модели распространены наиболее широко. Они являются принадлежностью рабочего места. Эти модели отличаются простотой изменения конфигурации, что позволяет настраивать компьютерную систему оптимально для решения именно тех задач, которые необходимо решить на отдельно взятом рабочем месте.

Портативные модели удобны для транспортировки. С портативным компьютером можно работать и при отсутствии рабочего места. Достоинством портативных компьютеров является возможность использовать их в качестве средства связи, особенно это актуально при применении в качестве канала связи сотовых сетей.

Карманные модели не могут сравниться по вычислительной мощности с настольными или портативными компьютерами, однако настроенные на решение определенной задачи, они являются незаменимым инструментом. Причем на карманные компьютеры сегодня устанавливаются собственная операционная система. Наиболее часто такие модели используют в качестве коммуникаторов, навигаторов, записных книжек, калькуляторов, радиоприемников или, даже, телевизоров.

2.4 Состав персонального компьютера

Состав персонального компьютера называется **аппаратной конфигурацией**. Как было сказано выше, современные компьютеры и вычислительные комплексы имеют блочно-модульную конструкцию, поэтому аппаратную конфигурацию, необходимую для решения конкретных задач, можно собирать из готовых устройств.

По способу расположения устройств относительно микропроцессора различают внутренние и внешние устройства. Внешними, как правило, являются большинство устройств ввода-вывода данных, которые называют периферийными устройствами, и некоторые устройства хранения данных.

Рассмотрим состав персонального компьютера более подробно.

2.4.1 Материнская плата

Материнской платой называют основную печатную плату персонального компьютера, на которую устанавливаются все системные компоненты. От качества материнской платы зависит надежность и скорость взаимодействия различных узлов компьютера.

Компоненты материнской платы подразделяются на внутренние и внешние элементы. Внутренние элементы устанавливаются на заводе-изготовителе, соединение которых выполняется предварительным травлением медной фольги, нанесенной на подложку материнской платы.

Внешние элементы, в соответствии с открытой архитектурой, представляют собой готовые устройства различных производителей, которые устанавливаются в разъемы материнской платы. К таким устройствам относятся процессор, оперативная память, видеокарта, сетевая плата.

Некоторые материнские платы могут иметь все элементы, необходимые для работы персонального компьютера. У большинства материнских плат состав внутренних элементов включает лишь основные узлы, а также встроенные видеоадаптер и звуковую карту. Материнская плата содержит множество внутренних компонентов, определяющих архитектуру персонального компьютера: разъем под процессор, кэш-память, контроллер винчестера, разъемы под оперативную память, контроллер клавиатуры и мыши, перезаписываемую базовую систему ввода-вывода, разъем для подключения видеокарты, сетевой адаптер, контроллеры внешних устройств и т.д. (рисунок 2.5).

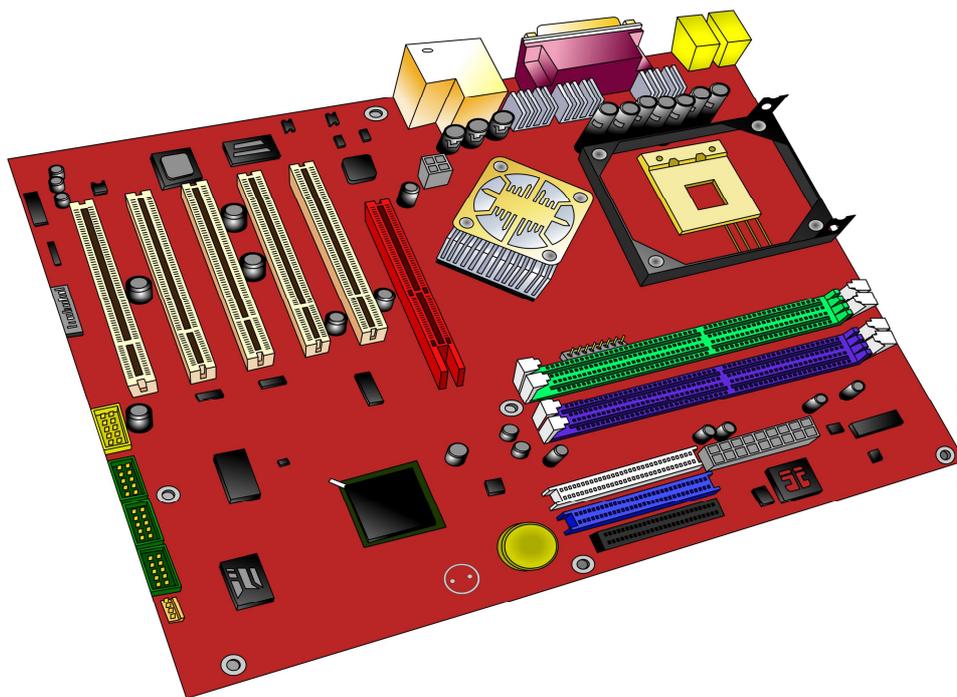


Рисунок 2.4 – Изображение материнской платы

Все материнские платы можно классифицировать по типу используемой оперативной памяти, по типу чипсета, по применяемому разъему микропроцессора и по форм-фактору (рисунок 2.6). Рассмотрим каждый из этих классификационных признаков более подробно.

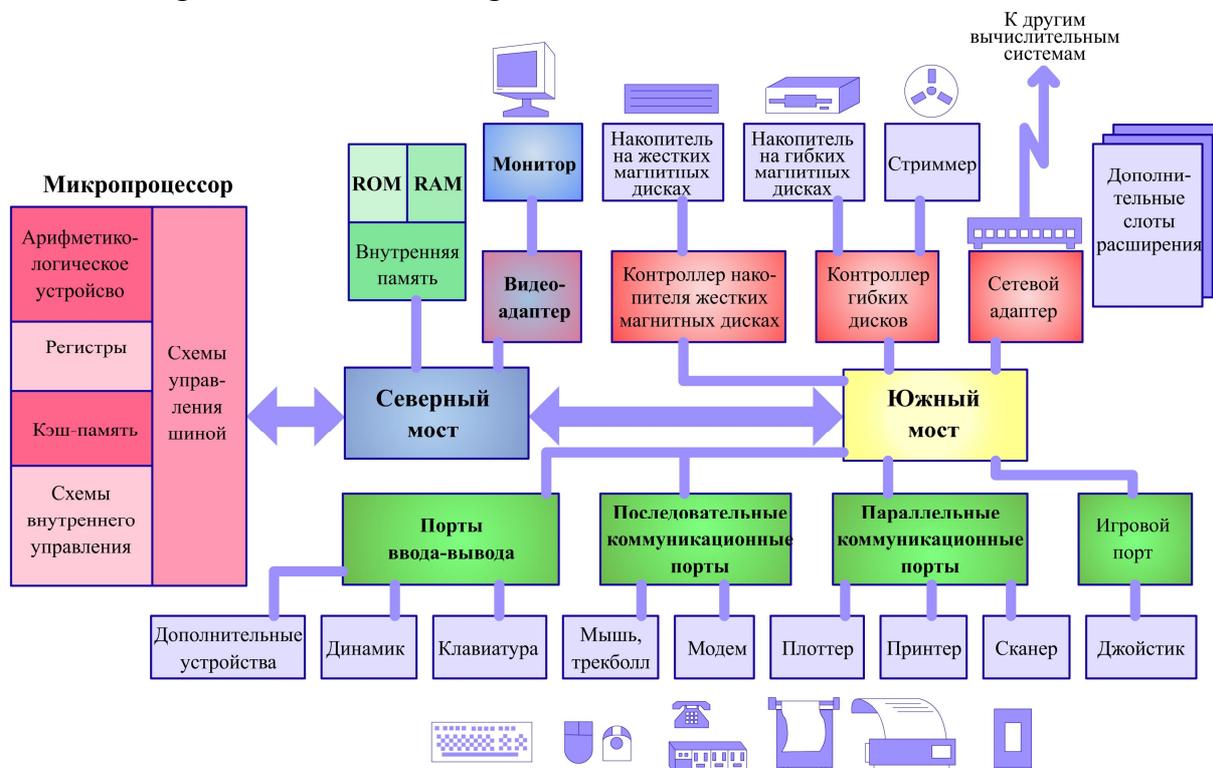


Рисунок 2.5 – Состав материнской платы



Рисунок 2.6 – Классификация материнских плат

Материнская плата обычно поддерживает один тип оперативной памяти. В зависимости от типа памяти количество разъемов для ее установки может быть различным. Спецификация каждой материнской платы также регламентирует максимальный размер оперативной памяти, с которой может работать чипсет.

Персональный компьютер состоит из устройств, которые подключены к материнской плате и занимаются приемом, обработкой и передачей информации. Логической организацией этой работы занимаются **чипсеты** (Chipset).

Современные чипсеты содержат в себе периферийный контроллер, контроллер DMA, контроллер для Enhanced IDE устройств, часы реального времени с CMOS-памятью, контроллер прямого доступа к памяти, таймер, систему управления памятью и шиной; поддерживают спецификацию шины PCI, синхронную и асинхронную кэш-память и клавиатурный контроллер.

Чипсет состоит из двух микросхем, называемых северным и южным мостами. Эти мосты выполняют функции связи различных шин и интерфейсов. Северный мост отвечает за взаимодействие с микропроцессором, оперативной памятью, видеокартой и южным мостом. Следовательно, северный мост контролирует работу самых производительных устройств персонального компьютера. Южный мост соединен с медленными узлами: шиной для подключения внешних устройств PCI, жестким диском, интерфейсом USB, портом клавиатуры и мыши PS/2.

Вид разъема для центрального процессора, подключаемого к материнской плате, бывает всего двух типов: Socket – плоский разъем для установки микропроцессора с выводами, перпендикулярными корпусу (наиболее популярный разъем), и Slot – щелевой разъем для установки микропроцессора с контактами по краю.

Форм-фактор материнской платы определяет ее геометрические размеры и параметры электропитания. Выбор форм-фактора определяется требованиями по энергопотреблению, а также по занимаемой площади, в зависимости от этого предпочтение отдается той или иной материнской плате.

2.4.2 Микропроцессор

Микропроцессором называют электронное устройство, которое состоит из одной или нескольких микросхем, осуществляющих выполнение арифметических, логических и операций управления, представленных в машинном коде.

Часть микропроцессора, выполняющая арифметические и логические операции, называется **ядром**.

Микропроцессор является центральным устройством компьютера, к которому направлены все информационные потоки. С устройствами компьютера микропроцессор связан несколькими группами проводников, которые называют процессорной шиной. В соответствии с принципами Гарвардской архитектуры, предполагающими раздельное хранение и обработку команд и данных, компьютер содержит три основные шины.

Адресная шина – часть процессорной шины, предназначенная для передачи адресных данных.

Шина данных – часть процессорной шины, предназначенная для передачи данных между различными устройствами компьютера.

Шина управления – часть процессорной шины, предназначенная для передачи сигналов управления.

Работа большинства современных микропроцессоров для персональных компьютеров основана на версии циклического процесса последовательной обработки информации, изобретенного Джоном фон Нейманом. Рассмотрим важнейшие этапы этого процесса (рисунок 2.7).

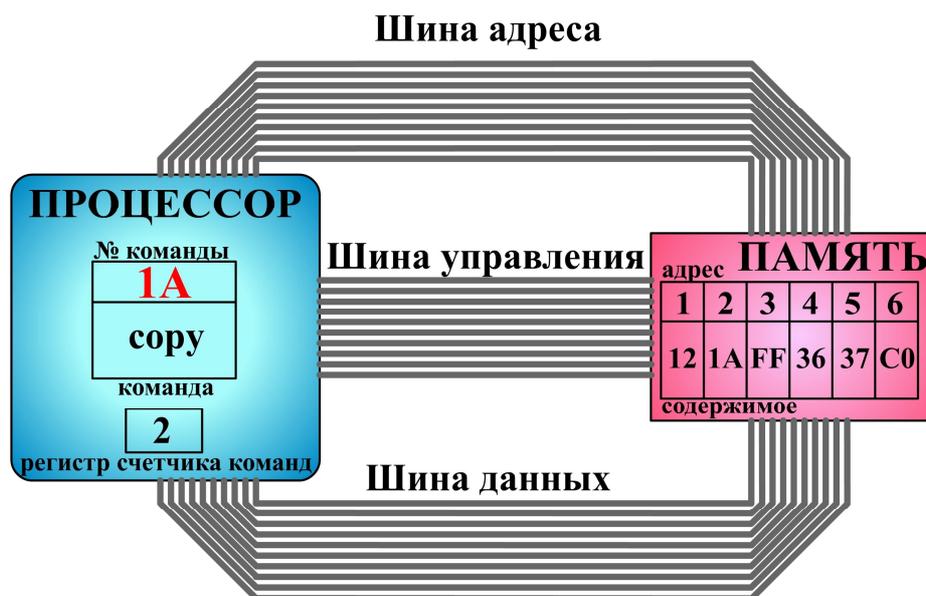


Рисунок 2.7 – Архитектура фон Неймана

Сначала микропроцессор выставляет число, хранящееся в регистре счетчика команд, на шину адреса и отдает памяти команду чтения. Выставленное число является для памяти адресом; память, получив адрес и команду чтения, выставляет содержимое, хранящееся по этому адресу, на шину данных и сообщает о готовности. Далее микропроцессор получает число с шины данных, интерпретирует его как команду из своей системы команд и исполняет ее. Если последняя команда не является командой перехода, процессор увеличивает на единицу число, хранящееся в счетчике команд, и начинается выполнение сле-

дующей команды. Очередность считывания команд изменяется в случае, если процессор считывает команду перехода, тогда адрес следующей команды может оказаться другим.

Для увеличения скорости обмена информацией между микропроцессором и оперативной памятью в микропроцессоре реализована **кэш-память**. При обращении микропроцессора к памяти сначала производится поиск необходимых данных в кэш-памяти, если необходимых данных не найдено, то только после этого процессор обращается к памяти. Так как скорость работы микропроцессора с кэш-памятью значительно выше, следовательно, повышается общее быстродействие вычислительной системы.

Разберемся более подробно с внутренней организацией процессоров. В современных микропроцессорах используется многоуровневая система кэширования. Кэш-память первого уровня L1, самая быстродействующая память, так как находится на самом кристалле микропроцессора, но из-за ограничений в размерах кристалла микропроцессора L1 имеет небольшую емкость. L1 разделяется на две части: кэш инструкций (L1I) и кэш данных (L1D).



Рисунок 2.8 – Многоуровневая система кэширования

Кэш-память второго уровня L2 имеет более высокое быстродействие по сравнению с оперативной памятью, но более низкое по сравнению с кэш-памятью первого уровня L1. Однако ее размер больше, чем размер L1, и там располагаются как команды, так и данные. Кэш-память второго уровня L2 может быть расположена на кристалле микропроцессора или выполнена в виде отдельных микросхем.

Для того чтобы исполнить команды, поступающие из памяти, **декодер** микропроцессора преобразовывает их во внутренний код, состоящий из элементарных операций. Декодер является очень важной частью современного микропроцессора, так как от его быстродействия зависит то, насколько постоянным будет поток команд, поступающих на исполняющие блоки.

После того, как команды пройдут через все уровни кэш-памяти и декодер, они поступают на исполняющие устройства, к которым относятся арифметико-логические устройства, блок вычислений с плавающей запятой и регистры.

Арифметико-логические устройства отвечают за два типа операций: арифметические действия с целыми числами, логические операции с целыми числами.

Блок вычислений с плавающей запятой занимается выполнением команд, работающих с числами с плавающей запятой, а также содержит дополнительный набор команд расширений.

Арифметико-логические устройства и блок вычислений с плавающей запятой могут состоять из нескольких устройств, которые способны работать параллельно.

Регистры представляют собой ячейки памяти, расположенные в процессорном ядре. Скорость работы микропроцессора с регистрами во много раз превосходит скорость работы с кэш-памятью любого уровня. Поэтому большинство команд микропроцессора предусматривают осуществление действий именно над содержимым регистров, а не над содержимым памяти. Однако общий объем регистров микропроцессора чрезвычайно мал.

Большинство современных микропроцессоров относятся к классу конвейерных суперскалярных процессоров с внеочередным исполнением операций. Рассмотрим кратко каждую из этих характеристик.

Конвейерная организация микропроцессора означает, что многие сложные действия разбиваются на этапы с небольшим временем выполнения.

Суперскалярная организация означает, что на каждом этапе обрабатываются сразу несколько потоков инструкций – от выборки инструкций из кэш-памяти до их завершения.

Внеочередное исполнение операций означает, что операции не обязаны выполняться в исполняющих устройствах строго в том порядке, который определен в программном коде.

Для увеличения производительности в современных микропроцессорах используются еще две очень эффективные технологии: предсказание ветвлений и предвыборка данных.

Блок предсказаний ветвлений анализирует поступающие команды и пытается предсказать, на какой участок кода укажет команда условного перехода еще до того, как она будет исполнена. **Блок предвыборки данных** пытается предсказать, какие данные необходимо загрузить в кэш-память еще до того, как они могут понадобиться. В результате работы этих блоков в кэш-памяти оказываются необходимые данные до того, как их затребовал микропроцессор, что значительно ускоряет процесс их обработки, так как нет необходимости обращаться к оперативной памяти.

Рассмотрим общую структуру процессора и взаимодействие его исполняющих блоков при прохождении машинной инструкции (рисунок 2.9).

Код программы, наиболее часто исполняемый в данный момент времени, размещен в кэш-памяти инструкций (L1I). В случае отсутствия в этой кэш-памяти нужных инструкций, они считываются из кэш-памяти второго уровня (L2).

Так как микропроцессоры являются одними из самых сложных полупроводниковых устройств с числом транзисторов порядка одного миллиарда, то необходимо кратко охарактеризовать этапы его изготовления.

Для создания современных микропроцессоров используется сложный технологический процесс, включающий более 300 отдельных этапов.

Изготовление процессоров начинается с производства подложек, заключающегося в выращивании монокристалла, который разрезается на «блины» с помощью кольцевой алмазной пилы и полируется. Затем происходит легирование монокристалла, что позволяет менять электрические свойства определенных областей и слоев, а не всей структуры кристалла. Далее осуществляется создание маски, которая пропускает излучение высокой интенсивности только на определенные участки кристалла. С помощью фотолитографии на кремниевой подложке, на которую предварительно наносят диоксид кремния, а затем фоторезистивный материал, формируется нужная структура. При травлении и очистке подложка переходит на новый этап, где удаляется ослабленный фоторезистивный материал, что позволяет получить доступ к диоксиду кремния.

После этого готовые подложки тестируются на установках зондового контроля. При положительных результатах тестирования с помощью разрезания из подложки получают отдельные ядра. Затем функциональное ядро связывают с процессорной упаковкой, используя клейкий материал. Далее осуществляют проводные соединения, связывающие контакты и сам кристалл. После чего ядро заключают в керамическую или пластиковую упаковку, что позволяет предотвратить повреждение, и оснащается распределителем тепла. Последний этап подразумевает тестирование микропроцессора, происходящее при повышенных температурах в соответствии со спецификациями микропроцессора.

Отдельно хотелось бы остановиться на процессе фотолитографии, наиболее интересном этапе изготовления микропроцессоров для оптотехников.

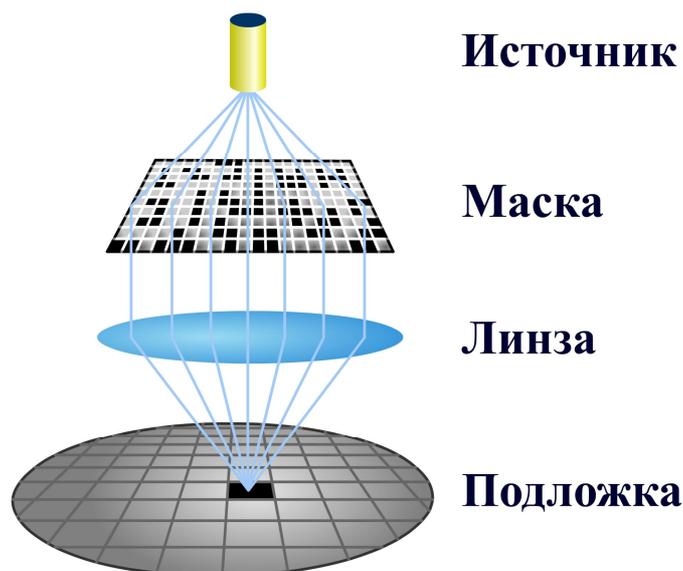


Рисунок 2.10 – Схема осуществления процесса фотолитографии

Через маску подложка облучается источником энергии. В качестве источников энергии могут выступать пучки электронов для создания масок, рентгеновские лучи и пучки ионов для исследовательских целей, в промышленном производстве используют жесткое ультрафиолетовое излучение и газовые лазеры.

Жесткое ультрафиолетовое излучение с длиной волны 13,5 нм облучает фоторезистивный материал, проходя через маску. Время проецирования и фокусировка являются определяющими факторами получения требуемого результата при фотолитографии. Дефокусировка приведет к тому, что останутся лишние частицы фоторезистивного материала, поскольку некоторые отверстия в маске не будут облучены должным образом. Если время проецирования будет слишком маленьким, то структура из фоторезистивного материала будет слишком широкой. С другой стороны, чрезмерное время проецирования создает слишком узкую структуру из фоторезистивного материала.

Специальная шаговая проекционная установка перемещает подложку в нужное положение. Затем может проецироваться строчка или один участок, соответствующий одному кристаллу процессора.

2.4.3 Оперативная память

Под **оперативной памятью** понимают набор микросхем, предназначенный для временного хранения данных, когда компьютер включен. Оперативная память предоставляет возможность произвольного доступа к любой своей ячейке в произвольном порядке. При выключении компьютера все данные, которые находились в оперативной памяти, теряются.

Оперативная память является очень важным устройством компьютера, потому что постоянно взаимодействует с микропроцессором. Следовательно, от производительности оперативной памяти во многом будет зависеть скорость работы компьютера в целом.

Классификация микросхем оперативной памяти приведена на рисунке 2.11. Рассмотрим каждый из этих критериев более подробно.



Рисунок 2.11 – Классификация микросхем оперативной памяти

Одним из самых распространенных критериев классификации памяти является ее **объем**, чем больше объем оперативной памяти, тем выше производительность вычислительной системы, так как скорость работы микропроцессора с внешними устройствами хранения данных значительно ниже.

Структура памяти определяется количеством ячеек памяти и разрядностью каждой ячейки. Под **разрядностью** ячейки понимают количество бит, с которыми операция чтения или записи может быть выполнена одновременно. Разрядность памяти обычно согласуется с разрядностью процессорной шины.

По типу памяти различают статическую и динамическую память.

В статической памяти ячейки построены на различных вариантах схем с двумя устойчивыми состояниями, называемыми триггерами. После записи бита в ячейку она может пребывать в этом состоянии при наличии питания сколь угодно долго. Рассмотренная выше кэш-память является статической памятью.

В динамической памяти ячейки построены на основе областей с накоплением зарядов, практически не потребляющих энергии при хранении. При записи бита в такую ячейку в ней формируется электрический заряд, который сохраняется в течение нескольких миллисекунд, затем происходит регенерация, то есть стирание и последующая запись или перезапись данных.

Статическая память работает на порядок быстрее динамической памяти, однако динамическая память имеет значительно больший объем, по сравнению со статической памятью.

Время доступа к данным, находящимся в оперативной памяти, определяется, в основном, скоростью чтения и записи. Скорость чтения и записи информации в память ограничивается пропускной способностью самой памяти. Латентность является не менее важной характеристикой с точки зрения быстродействия подсистемы памяти, чем скорость чтения и записи данных, но совершенно другой по сути. Под **латентностью** понимают задержки, возникающие при доступе к любому произвольно взятому адресу. Следовательно, при передаче больших объемов данных с одних адресов определяющей является скорость чтения и записи данных, если же передаются малые порции данных с большого количества адресов, то на первый план выходит латентность.

Форм-фактор памяти определяется ее интерфейсом. Выделяют несколько интерфейсов, представляющих на сегодня интерес.

SDRAM представляет собой динамическое оперативное запоминающее устройство с синхронным интерфейсом, использует внутреннее разделение массива памяти на два независимых банка, что позволяет увеличивать разрядность, совмещая выборку из одного банка с установкой адреса в другом банке.

DDR SDRAM представляет собой синхронное динамическое оперативное запоминающее устройство с удвоенной скоростью передачи данных. Достигается это за счет приема и передачи данных два раза за такт на обоих фронтах тактовых импульсов.

DRDRAM является разновидностью синхронной памяти, снабженной специальным интерфейсом. Каждая микросхема такой памяти имеет внутреннюю многобанковую структуру, чем обеспечивается ее высокая пропускная

способность. Для связи памяти с контроллером памяти используется специальная быстродействующая шина.

2.4.4 Видеокарта

Видеокарта представляет собой устройство, предназначенное для обработки видеосигналов, поступающих с внешних устройств, а также для вывода графики информации на экран монитора.

Видеокарта является одной из самых мощных вычислительных систем компьютера, освобождающей центральный микропроцессор от большого объема операций с графическими данными. Микропроцессор, видеокарта и оперативная память определяют общую производительность компьютера. Рассмотрим более подробно принципы работы видеокарты (рисунок 2.12).

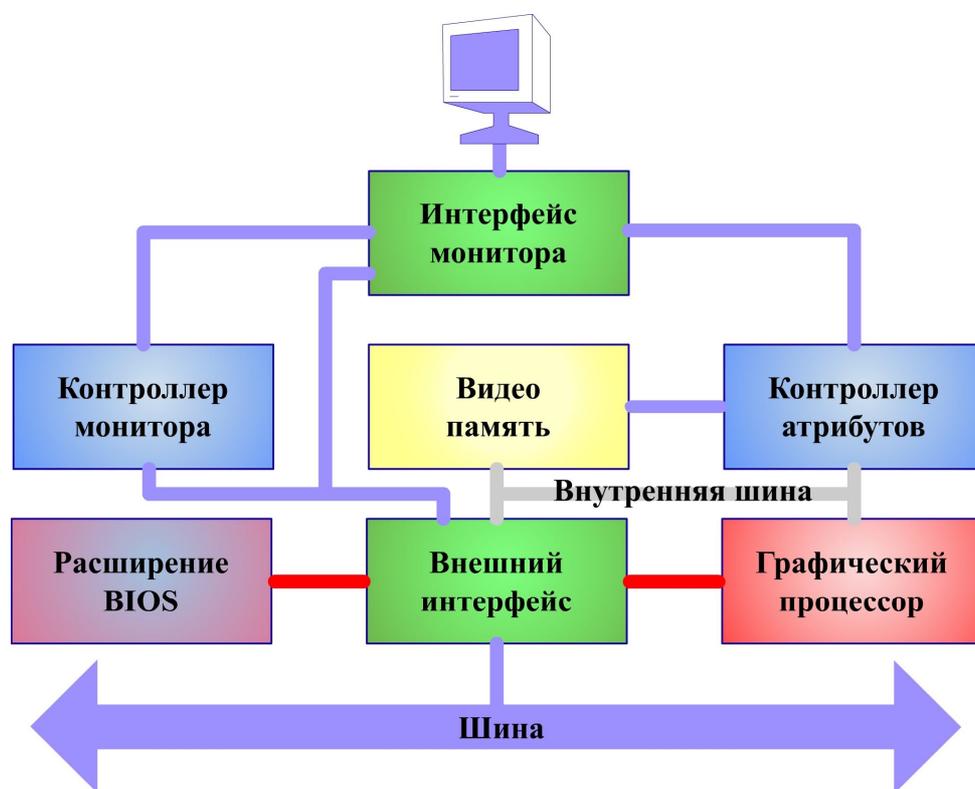


Рисунок 2.12 – Функциональная схема видеокарты

Обязательным элементом видеокарты является контроллер монитора, в задачу которого входит согласованное формирование сигналов сканирования видеопамати и сигналов вертикальной и горизонтальной синхронизации монитора.

Видеопамать является специальной областью памяти, из которой контроллер монитора организует циклическое чтение содержимого для регенерации изображения. Видеокарты поддерживают архитектуру унифицированной памяти, при которой под видеопамать выделяется область оперативной памяти. Однако это приводит к снижению производительности компьютера в целом.

Для повышения производительности применяются микросхемы видеопамати со специальной архитектурой.

Контроллер атрибутов управляет цветовой информацией, хранящейся в видеопамати. В текстовом режиме он обрабатывает информацию текущего знакоместа, а в графическом – текущего выводимого пикселя. В состав контроллера атрибутов входят регистры палитр, которые служат для преобразования цветов, закодированных битами видеопамати, в реальные цвета на экране.

Графический процессор является средством повышения производительности обработки графической информации в видеопамати. Многие функции работы с видеоданными реализованы на аппаратном уровне.

Внутренняя шина видеокарты предназначена для высокопроизводительного обмена данными между видеопаматью, графическим процессором и внешним интерфейсом. Блок интерфейса монитора формирует выходные сигналы соответствующего типа.

Модуль расширения BIOS хранит код драйверов видеокарты и обеспечивает возможность ее совместимости с другими устройствами компьютера.

Блок внешнего интерфейса связывает адаптер с одной из шин компьютера. Современные графические адаптеры используют, в основном, высокопроизводительные шины, такие как AGP и PCI-Express.

Для оценки эффективности работы видеокарт используют два показателя: количество кадров, обрабатываемых в секунду, и поддержка библиотек трехмерного моделирования (3D).

Количество обрабатываемых в секунду кадров (**FPS**, frames per second) является обобщенной характеристикой многих технических параметров видеокарты, таких, как частота ядра графического процессора, технология изготовления чипсета, разрядность внутренней шины, означает производительность обработки графической информации.

На сегодняшний день существует два стандарта аппаратной поддержки библиотек трехмерного моделирования.

Библиотека Direct3D предоставляет прямой доступ к памяти видеокарты для обработки изображения, микширования звука, работы с джойстиком, клавиатурой и мышью, создания анимационных эффектов.

Библиотека OpenGL была создана специально для программ трехмерного моделирования, содержит средства работы с трехмерной графикой.

2.4.5 Устройства хранения данных

В 1837 г. Чарльз Бэббидж предложил принципиально новую структурную организацию вычислительной машины, в которой отдельно выделил устройство для хранения чисел – «склад». При этом долгосрочное хранение данных и программ осуществлялось с помощью перфокарт. Качественный шаг вперед в технологии долгосрочного хранения информации был сделан в 1973 г. инженерами фирмы IBM, разработавшими первый накопитель на жестких магнитных дисках.

2.4.5.1 Жесткий диск

Жесткий диск (Hard Disk Drive, HDD) представляет собой устройство, предназначенное для долговременного хранения данных. Часто в литературе жесткие диски называют винчестерами. По способу записи и чтения данных жесткие диски относятся к магнитным накопителям, состоящим из механического блока и платы электроники.

Плата электроники состоит из процессора, необходимого для подготовки записываемых и обработки считанных сигналов, постоянного запоминающего устройства с программой работы жесткого диска и кэш-памяти.

В герметичном механическом блоке с очищенным воздухом установлен шпиндель с несколькими магнитными дисками. Магнитные диски представляют собой пластины из алюминия, керамики или стекла, на которые нанесен тонкий слой высококачественного ферромагнетика. Под дисками находится двигатель, вращающий шпиндель. Запись и чтение данных осуществляется с помощью магнитных головок.

Рассмотрим физические основы записи и чтения данных на магнитный диск. Магнитное покрытие диска представляет собой множество мельчайших областей самопроизвольной намагниченности. Под воздействием внешнего магнитного поля собственные магнитные поля доменов ориентируются в соответствии с его направлением. После прекращения действия внешнего поля на поверхности диска образуются зоны остаточной намагниченности, что соответствует записи данных на диск. Участки остаточной намагниченности, оказавшись при вращении диска напротив зазора магнитной головки, наводят в ней электродвижущую силу, изменяющуюся в зависимости от величины намагниченности.

На поверхности дисков в результате записи информации образуются намагниченные участки в форме концентрических окружностей, которые называются **дорожками**. Каждая дорожка в свою очередь разбивается на **сектора**. Совокупность дорожек, расположенных друг под другом на всех поверхностях дисков, образует **цилиндр**.

Существует несколько параметров, характеризующих эффективность работы жесткого диска (рисунок 2.13).

Скорость вращения шпинделя определяет скорость обмена данными. Размер жесткого диска равен максимальному количеству данных, которое способно хранить запоминающее устройство. Плотность записи данных характеризует количество секторов, расположенных на одном цилиндре. Протокол передачи данных определяет используемый интерфейс передачи данных, например, IDE, SCSI, IEEE или USB. Среднее время доступа описывает процесс позиционирования магнитной головки на нужное место жесткого диска. Размер кэш-памяти определяет общую производительность работы жесткого диска с устройствами персонального компьютера. Среднее время наработки на отказ характеризует надежность работы жесткого диска в течение длительного времени.



Рисунок 2.13 – Основные параметры жестких дисков

Отдельно необходимо отметить современные технологии в изготовлении жестких дисков. Например, в винчестерах последнего поколения применяется система шумоподавления, которая обеспечивает низкий уровень шума во время работы диска, и технология самостоятельного следящего анализа и отчетности (S.M.A.R.T.), которая предназначена, например, для диагностики жесткого диска, управления кэш-памятью, определения сбоев в работе магнитных головок или температуры диска.

В некоторых случаях необходимо получить устройство хранения данных, возможности которого должны превышать допустимые параметры обыкновенных жестких дисков. Этого можно добиться за счет избыточности и параллельного использования множества обычных жестких дисков, применяя технологию RAID (Redundant Array of Inexpensive Disks — избыточный массив недорогих дисков). В этом случае с помощью RAID-контроллера к компьютеру подключается группа обычных односторонних жестких дисков, работающая как один виртуальный диск с улучшенными свойствами.

2.4.5.2 Оптические накопители данных

Оптические накопители данных на сегодняшний день чрезвычайно востребованы в связи с возможностью хранения значительных объемов данных и низкой стоимостью самих накопителей.

Первым оптическим устройством хранения данных стали компакт-диски, данные с которых считывались с помощью специальных устройств CD-ROM (Compact Disc Read-Only Memory – постоянное запоминающее устройство на основе компакт-диска). Типовой CD-ROM состоит из платы электроники, шпиндельного двигателя, системы оптической головки и системы загрузки диска.

Плата электроники содержит необходимые управляющие микросхемы. Шпиндельный двигатель приводит компакт-диск во вращение с постоянной линейной скоростью. Сохранение постоянной линейной скорости требует изменения угловой скорости диска в зависимости от положения оптической головки.

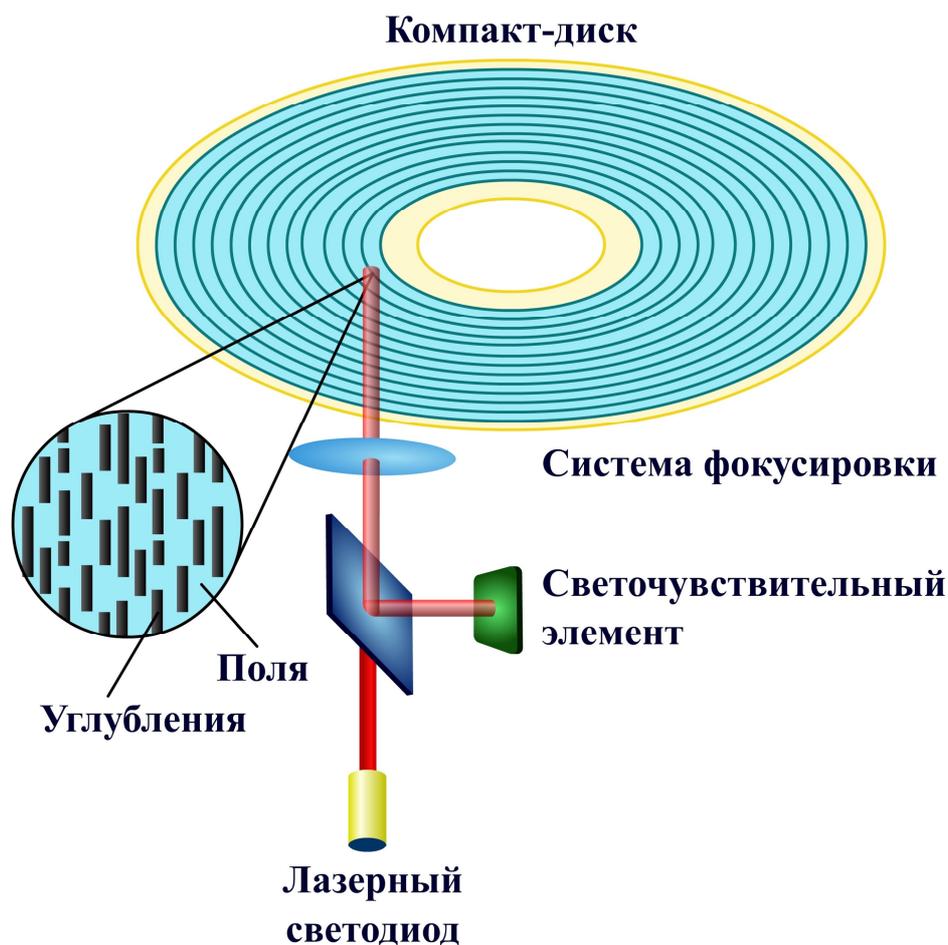


Рисунок 2.14 – Считывание данных с компакт-диска

Система оптической головки состоит из самой головки и устройства ее перемещения. В головке размещены лазерный излучатель на основе инфракрасного лазерного светодиода, система фокусировки, фотоприемник и предварительный усилитель. Система фокусировки представляет собой подвижную линзу, приводимую в движение электромагнитным устройством.

Компакт-диск **CD-R** состоит из трех основных слоев: подложки из поликарбоната, на которой размещается рельеф диска, нанесенного на нее отражающего покрытия и защитного слоя. Рельеф диска состоит из спиральной дорожки, идущей от центра к периферии, вдоль которой расположены углубления, образованные после воздействия лазера. Данные кодируются чередованием углублений и полей между ними. При многократной записи на компакт-диск **CD-RW** на подложке расположен регистрирующий слой, вещество которого меняет свое состояние под воздействием лазера и становится аморфным или кристаллическим. Однако у таких дисков низкая отражающая способность.

Считывание информации с компакт-диска происходит за счет регистрации изменений интенсивности отраженного от рельефа диска излучения слабого лазера. Светочувствительный элемент фиксирует, отразился луч от полей, был рассеян или поглощен в месте углубления. После этого электриче-

ский сигнал со светочувствительного элемента передается на плату электроники, где преобразуется в двоичные данные (рисунок 2.14).

Компакт-диски первоначально были предназначены для хранения звуковой информации, в то время как следующий вид оптических дисков DVD направлен на хранение видеoinформации. Первоначально аббревиатура DVD означала цифровой видеодиск (Digital Video Disc), однако, затем из-за расширения функций таких дисков аббревиатура стала читаться иначе – цифровой многофункциональный диск (Digital Versatile Disk).

Принцип работы DVD заключается в том, что луч лазера вызывает кристаллографические изменения в активном слое оптического диска, в результате облучения вещество меняет свое состояние с кристаллического на аморфное. Большая плотность записи достигается тем, что диаметр углублений на дорожке сокращен, и дорожки нанесены с меньшим шагом друг относительно друга.

По конструкции различают несколько типов DVD дисков.

Односторонний диск с однослойной записью имеет максимальную емкость 4,7 Гбайт. Такой диск состоит из 0,6 мм пленки, покрытой алюминием и наклеенной на чистую подложку. Технология напыления та же, что для обычного компакт-диска.

Односторонний диск с двухслойной записью имеет максимальную емкость уже 8,5 Гбайт. В этом случае создается полупрозрачный слой, который отражает до 30% лазерного излучения. Это позволяет считывать информацию с верхнего слоя. И, в то же время, полупрозрачный слой пропускает достаточно излучения, чтобы сигнал от нижнего уровня с высокой отражательной способностью тоже читался. Две половины диска соединяет клеевая прослойка (рисунок 2.15). Считывание данных с внутреннего или внешнего слоя производится с помощью перефокусировки оптической системы.

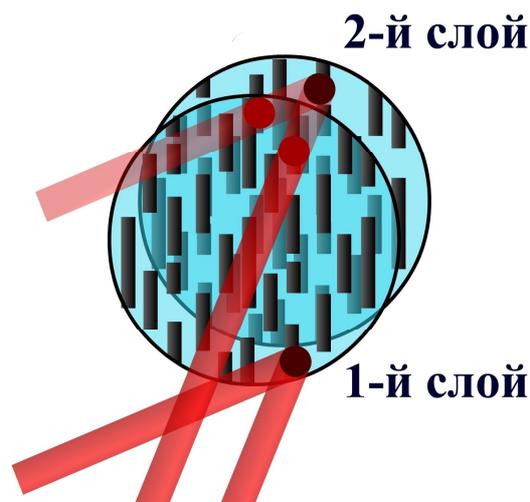


Рисунок 2.15 – Фрагмент одностороннего диска с двухслойной записью

Также существует двухсторонний диск с однослойной записью и двухсторонний диск с двухслойной записью. Однако перспектив развития у таких DVD технологий нет, так как максимально возможный объем хранимой информации не может превысить 17 Гбайт.

С целью увеличения емкости оптических носителей разработана технология записи и чтения данных лазерным излучением сине-фиолетового диапазона **BLU-RAY DISK**. Применение лазера с меньшей длиной волны позволило минимизировать рассеяние луча и увеличить апертуру оптической системы. Толщина защитного слоя при этом была уменьшена, ширина дорожки сократилась вдвое по сравнению с DVD, благодаря чему повысились емкость дисков и скорость чтения и записи данных.

2.4.5.3 Голографическая память

Устройства хранения данных, основанные на использовании **голографической технологии**, являются перспективными для работы с большими массивами данных. Отличительным признаком голографической технологии от других оптических методов является возможность использования объемной записи по всей толщине записывающего слоя при помощи различных углов наклона лазера, а также большая плотность записи.

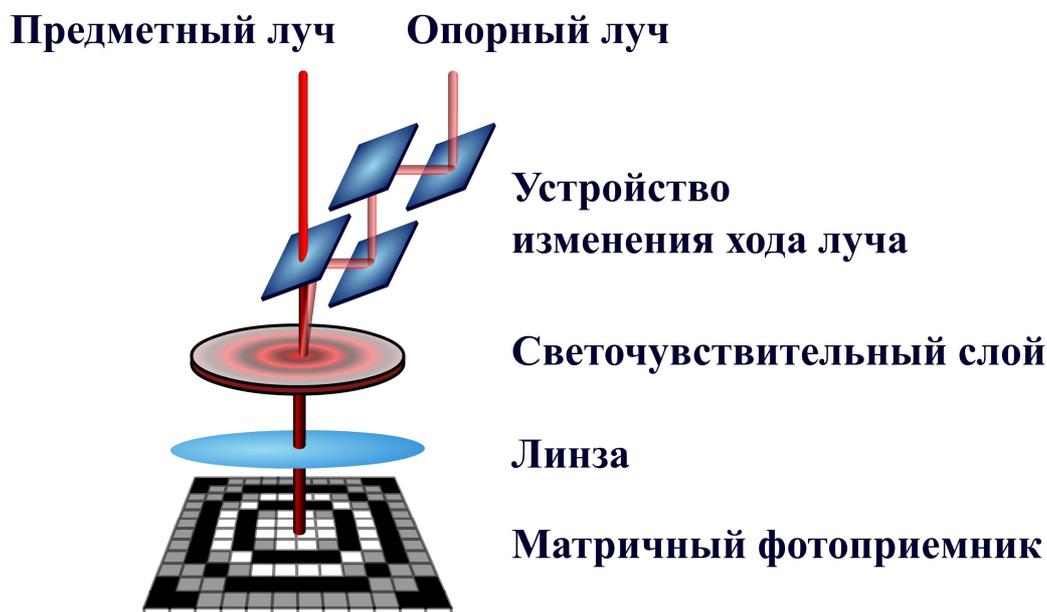


Рисунок 2.16 – Принцип работы голографической памяти

В голографических устройствах хранения запись данных производится следующим образом. На начальном этапе происходит разделение луча лазера на опорный луч и предметный луч, являющийся носителем данных. Оба луча направляются внутрь светочувствительного записывающего слоя, где и происходит их взаимодействие. В результате этого взаимодействия образуется ин-

терференционная картина, которая и является основой голограммы, и запоминается в виде набора вариаций коэффициента отражения внутри этого слоя.

При чтении данных светочувствительный слой освещается опорным лучом, который при взаимодействии с интерференционной картиной воспроизводит записанную страницу в виде образа из светлых и темных областей, фиксируемых на матричном фотоприемнике. При чтении данных опорный луч должен падать на светочувствительный слой под тем же самым углом, при котором производилась запись этих данных.

Основные проблемы, стоящие перед разработчиками объемной записи данных, связаны с созданием светочувствительных материалов, удовлетворяющих жестким требованиям голографической технологии.

2.4.5.4 Флэш-память

Флэш-памятью называют тип твердотельной полупроводниковой энергонезависимой перезаписываемой памяти. Флэш-память хранит данные в массиве транзисторов с плавающим затвором, причем каждый транзистор работает с одним битом. Существует два основных вида организации флэш-памяти: на логических элементах «Или-Не» (NOR) и логических элементах «И-Не» (NAND), отличающиеся архитектурой размещения транзисторов и их контактов.

Транзистор, представляющий собой ячейку флэш-памяти, имеет два изолированных затвора: управляющий и плавающий, способный удерживать электроны. Также в ячейке имеются сток и исток. При программировании между ними с помощью электрического поля создается поток электронов. Некоторые из них преодолевают слой изолятора и попадают на плавающий затвор, на нем они могут храниться в течение нескольких лет. Для стирания данных на управляющий затвор подается высокое отрицательное напряжение, и электроны с плавающего затвора переходят на исток. К недостаткам флэш-памяти можно отнести ограниченное количество циклов перезаписи, довольно высокую стоимость и низкую скорость по сравнению с жесткими дисками. Однако благодаря своей компактности и низкому энергопотреблению этот тип памяти получил широкое применение.

2.4.6 Устройства ввода данных

Существует множество систем, задачей которых является ввод информации в компьютер. Такие устройства отличаются друг от друга назначением, способом работы и ввода данных.

2.4.6.1 Клавиатура

Клавиатурой называют совокупность расположенных в определенном порядке клавиш, используемых для ввода данных в компьютер и их редактирования.

На сегодняшний день наиболее популярной является расширенная клавиатура, содержащая 101 и более клавиш. Все клавиши можно разделить на четыре группы: буквенно-цифровая группа, предназначенная для ввода символьных данных, функциональная группа, необходимая для выполнения определенных функций, группа клавиш управления курсором, служащих для управления позиции ввода данных, и группа клавиш дополнительной панели, дублирующих действие основных клавиш.

Клавиатура представляет собой унифицированное устройство со стандартным разъемом и последовательным интерфейсом связи с материнской платой. Драйвер клавиатуры входит в состав базовой системы ввода-вывода, и потому компьютер реагирует на нажатия клавиш сразу после включения. Клавиатура содержит внутренний контроллер, осуществляющий сканирование матрицы клавиш, управление индикаторами, внутреннюю диагностику и связь с материнской платой. В качестве датчиков нажатия клавиш применяют кнопки на основе токопроводящей резины, емкостные датчики и датчики на эффекте Холла. Клавиатура на сегодня является надежным, долговечным и дешевым устройством.

2.4.6.2 Мышь

В 1967 г. Дуглас Энгельбарт запатентовал «Индикатор координат X-Y для системы вывода изображений», который сегодня известен как мышь. **Мышью** называют координатное устройство, являющееся дополнением к клавиатуре, необходимое при пользовании графического интерфейса.

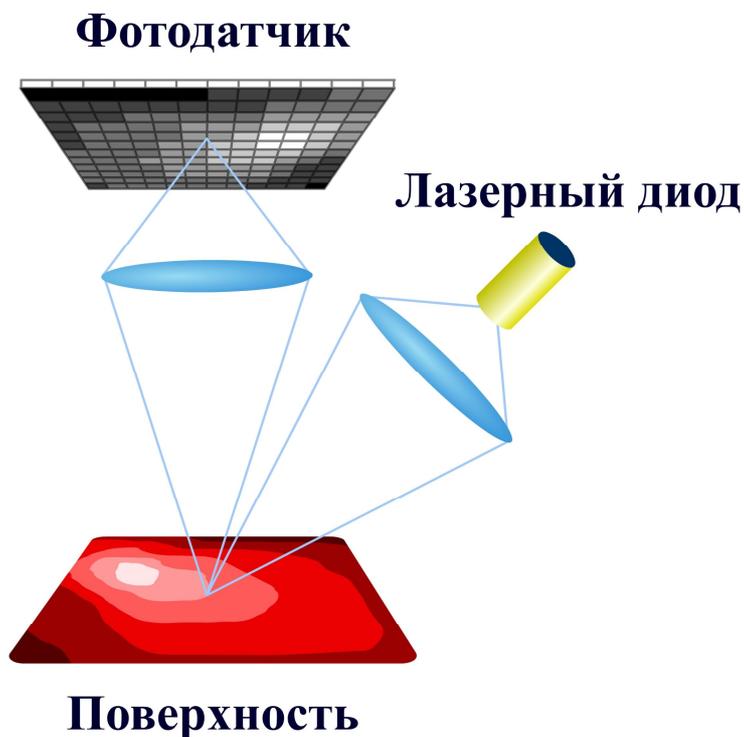


Рисунок 2.17 – Принцип формирования изображения мыши

Наиболее популярной является оптическая мышь на основе лазерных диодов. В состав такого устройства входит лазерный диод, используемый в качестве осветительного устройства, фотодатчик, представляющий собой мини-видеокамеру низкого разрешения, и блок электроники (рисунок 2.17).

Когерентные свойства лазерного излучения обуславливают высокую контрастность получаемого изображения при отражении от поверхности. Изображение, регистрирующееся на фотодатчике, проявляет детали практически любой поверхности.

Фотодатчик является матрицей фоточувствительных элементов. Фотодатчик последовательно регистрирует изображения освещенной поверхности, по которой перемещается мышь. После этого в блоке электроники происходит сравнение полученных изображений с целью определения расстояния и направления перемещения мыши (рисунок 2.18).

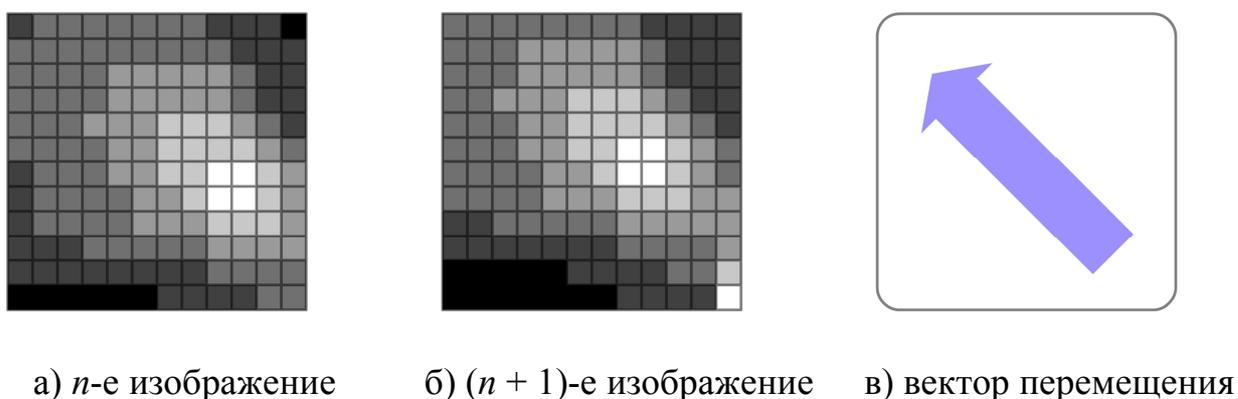


Рисунок 2.18 – Принцип определения расстояния и направления перемещения мыши

Несколько характерных величин в совокупности дают представление об итоговой производительности мыши. К таким величинам относятся разрешение, характеризующее точность захваченного изображения, частота кадров, определяющая количество считанных изображений в единицу времени, и максимальная скорость, равная такой скорости перемещения мыши, при которой система адекватно отражает это перемещение.

2.4.6.3 Сканер

Сканером называют устройство, предназначенное для считывания графических данных с поверхности носителя при помощи оптических средств, кодирования и ввода данных в компьютер. Сканер имеет несколько основных характеристик, рассмотрим их более подробно (рисунок 2.19).

Оптическое разрешение является параметром, определяющим максимальную разрешающую способность сканера без интерполяции. Разрешение измеряют количеством раздельно фиксируемых точек изображения на единицу

длины. Для повышения разрешения выполняют **интерполяцию** полученного изображения, при этом происходит генерация новых точек с присвоением им промежуточных значений по отношению к соседним точкам.



Рисунок 2.19 – Основные характеристики сканеров

Аппаратное разрешение определяется шагом перемещения фоточувствительных элементов сканера относительно поверхности носителя, размер шага перемещения зависит от точности механической части сканера.

Разрядность цвета, также называемая глубиной цвета, характеризует максимальное количество цветов, которое может воспроизвести сканер. Стандартом является формат TrueColor, в котором каждая точка кодируется тремя байтами, или 24 битами, на представление каждого основного цвета (красного, зеленого и синего) отводится по восемь бит. Увеличение числа разрядов внутри сканера открывает возможность цветовой коррекции изображения без внесения искажений.

Различные модели сканеров подключаются к компьютеру через разные интерфейсы, например, LPT, USB, SCSI или нестандартные интерфейсы. Нестандартные интерфейсы применяются в сканерах, требующих установки дополнительной интерфейсной платы. Интерфейс SCSI используется для высокоскоростной передачи больших объемов отсканированных данных. Большинство современных сканеров общего назначения снабжаются интерфейсом USB.

Тип оптической системы зависит от типа фоточувствительных элементов, применяемых в сканере. Выделяют два основных типа: CCD (Charge-Coupled Device) и CIS (Contact Image Sensor). Элемент CIS состоит из линейки датчиков, непосредственно воспринимающих световой поток от оригинала, причем размер фоточувствительных элементов сканирующей линейки равен размеру считываемой области. Такие сенсорные датчики более дешевые, однако качество получаемого изображения низкое. Элемент CCD представляет собой прибор с зарядовой связью, на который проецируется изображение объекта. Более дорогая оптическая система позволяет формировать качественное изображение.

Классификация сканеров приведена на рисунке 2.20. Ручной сканер применяется в качестве мобильного устройства ввода, однако его использование ограничено очень узким кругом задач. Более часто используются настольные сканеры.



Рисунок 2.20 – Классификация сканеров

Сканер со сканирующей головкой на плоттере применяется для больших по протяженности объектов, при этом головка перемещается по носителю данных по двум координатам, после чего происходит совмещение изображений. В планшетном сканере сканирующая головка перемещается относительно носителя данных с помощью шагового двигателя. В проекционном сканере перемещается только сканирующее устройство, позволяющее получить изображение без взаимного перемещения носителя данных и сканирующего элемента. В рулонном сканере отдельные листы документов протягиваются через роликовое устройство.

Рассмотрим более подробно принцип работы наиболее используемого планшетного сканера (рисунок 2.21).

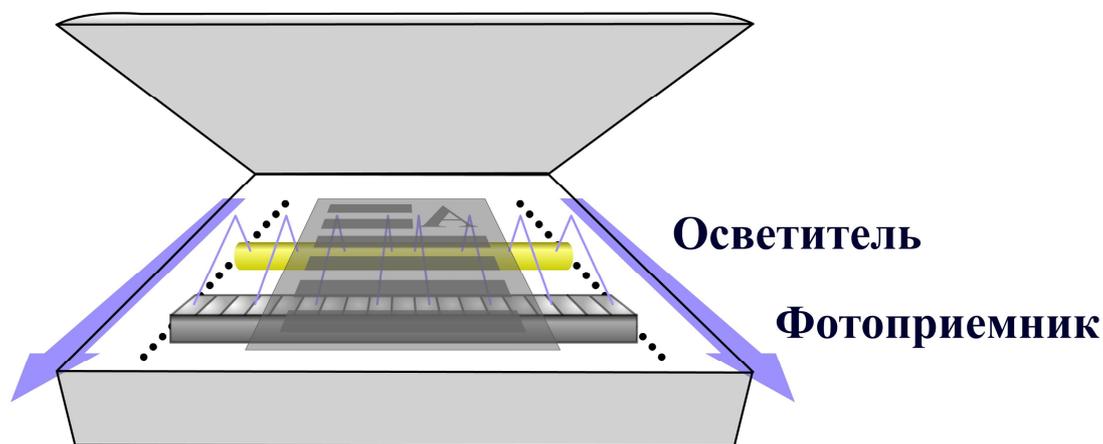


Рисунок 2.21 – Планшетный сканер

При сканировании фрагмент носителя данных освещается белым светом, отраженный свет фокусируется на линейном фотоприемнике, на котором формируется изображение строки носителя данных и передается в память сканера. После этого каретка с осветителем и линейным фотоприемником сдвигается, и сканируется следующая строка. В памяти сканера делается предварительная обработка изображения, которое через интерфейс передается в компьютер.

2.4.6.4 Веб-камера

Веб-камерой называют цифровую видеокамеру, предназначенную для ввода изображений в компьютер в реальном режиме времени и передачи изображения по компьютерной сети.

Первая веб-камера была установлена в 1991 г. в Кембриджском университете, с тех пор эти средства ввода данных в компьютер нашли широкое применение. Веб-камеры используются для проведения видеоконференций и научных исследований, применяются в развлекательных приложениях и охранных системах, внедряются в программы дистанционного обучения и идентификации доступа.

Конструктивно веб-камера состоит из объектива, оптического фильтра, матрицы фоточувствительных элементов, блока электроники, включающего в себя схемы цифровой обработки и сжатия изображения, микропроцессора, оперативной памяти, флэш-памяти и интерфейса передачи данных по компьютерной сети в компьютер (рисунок 2.22).

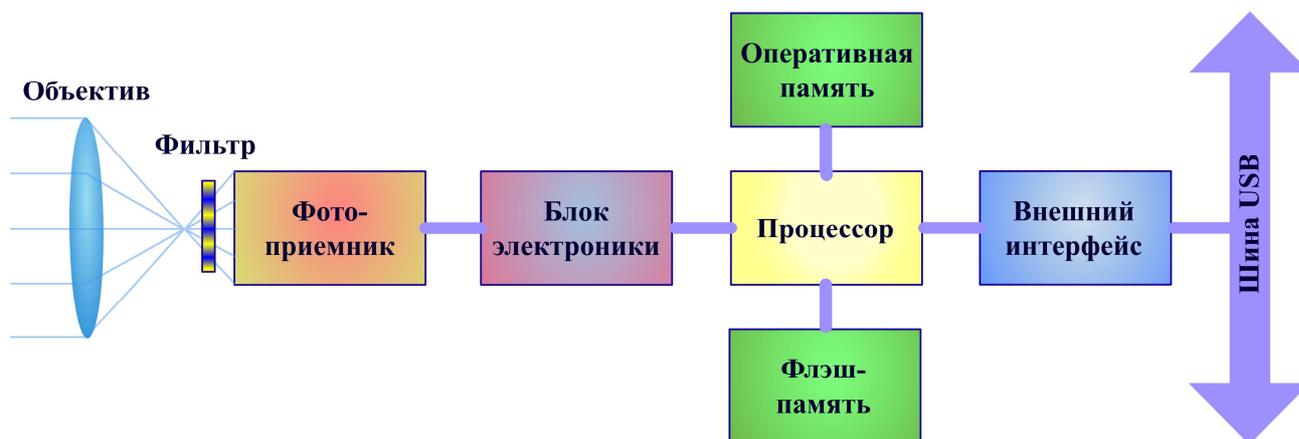


Рисунок 2.22 – Устройство веб-камеры

В качестве фотоприемника в веб-камерах в большинстве случаев применяются ПЗС-матрицы, которые позволяют преобразовывать световую энергию в электрические сигналы, доступные для дальнейшей обработки.

Оптические фильтры предназначены для отсеивания инфракрасной составляющей света, обеспечивая видеосистеме правильную цветопередачу.

Блок электроники осуществляет преобразование аналогового электрического сигнала в цифровой формат и дальнейшее его преобразование в один из форматов сжатия для сокращения размера видеоданных.

Процессор осуществляет операции по выводу оцифрованных и сжатых видеоданных. В оперативной памяти хранятся временные данные, необходимые для работы процессора. Во флэш-памяти размещаются программы веб-камеры, которые можно перезаписывать.

Видеоданные передаются в компьютер, в основном, через интерфейс USB, также в большинстве веб-камер предусмотрена отправка данных по компьютерной сети Ethernet.

Дополнительно в веб-камере могут быть реализованы такие функции, как детектор движения, защита данных паролем, передача аудиосигнала и сигнала тревоги.

2.4.7 Устройства вывода данных

Существует большое количество систем, предназначенных для вывода данных из персонального компьютера. К ним в большей части относятся устройства визуального отображения данных и вывода данных на материальные носители. Во многих таких устройствах происходит преобразование световой энергии, что должно быть интересно для оптотехников. Рассмотрим устройства вывода данных более подробно.

2.4.7.1 Монитор

Монитором называется устройство отображения данных для их визуального восприятия, а также контроля и управления работой компьютера.

Выделяют несколько классификационных признаков мониторов, к основным признакам относятся тип экрана монитора, возможность цветопередачи, размер экрана и разрешающая способность (рисунок 2.23).



Рисунок 2.23 – Основные классификационные признаки мониторов

Мониторы могут работать в двух режимах: текстовом и графическом. В текстовом режиме работы экран монитора разбивается на отдельные участки, называемые знакоместом, в которые могут быть выведены изображения заранее заданных символов.

В графическом режиме экран монитора разделяется на множество точек, называемых пикселями, изменением яркости которых можно создавать произвольные изображения. При этом количество пикселей, расположенных по горизонтали и вертикали, определяет разрешающую способность монитора.

Возможность цветопередачи определяет качество отображения цвета и его оттенков. Выделяют цветные и монохромные мониторы. В большинстве случаев используются цветные мониторы, однако иногда для научных или про-

изводственных целей, когда необходимо выводить лишь текстовую черно-белую информацию, применяют в качестве таких индикаторов монохромные мониторы.

В системах цветовоспроизведения мониторов используется аддитивная цветовая модель RGB, введенная в 1860 г. Максвеллом. Эта система основана на применении трех первичных цветов: красного (Red), зеленого (Green) и синего (Blue). При смешивании этих цветов можно получить практически любой оттенок. Так, в мониторе три световых пучка проходят через красный, зеленый и синий светофильтры и смешиваются, что соответствует формированию одного пикселя цветного изображения.

Размер экрана монитора также является важным параметром, позволяющим отображать все большее количество информации. Современные технологии производства мониторов направлены на увеличение размера экрана монитора. Однако необходимо понимать, что качество изображения зависит не только от размера монитора, но и от разрешающей способности, контрастности, яркости и, конечно же, типа монитора.

Имеется множество технологий создания мониторов, но наиболее доступными и используемыми типами мониторов являются мониторы на электронно-лучевой трубке, жидкокристаллические мониторы, LED-мониторы и плазменные мониторы.

2.4.7.1.1 Монитор на электронно-лучевой трубке

Мониторы на электронно-лучевой трубке (CRT, Cathod Ray Tube) основаны на использовании электровакуумного прибора – электронно-лучевой трубки, в которой происходит непрерывная бомбардировка электронами люминесцентного экрана. Управление шириной пучка электронов, его движением по поверхности экрана и интенсивностью производится с помощью электромагнитного поля (рисунок 2.24).

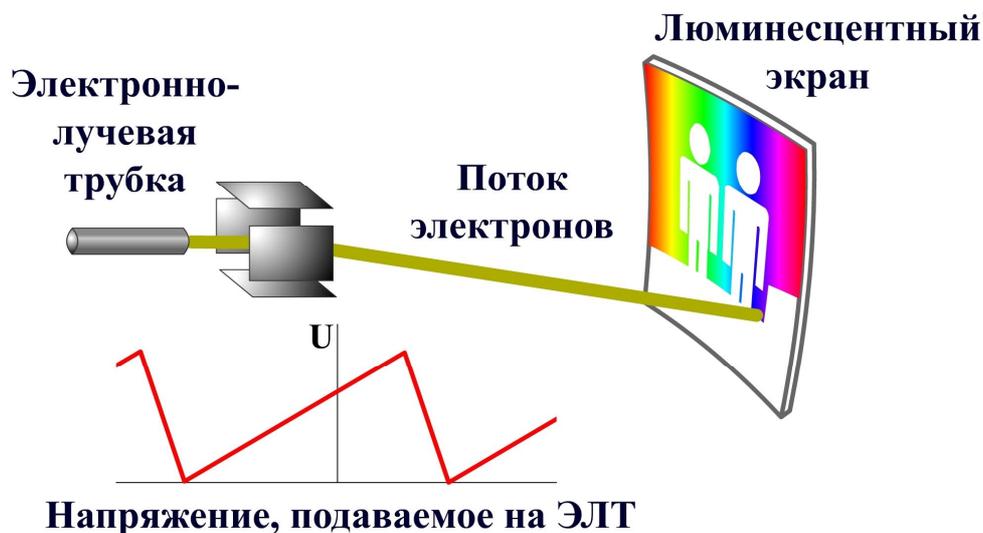


Рисунок 2.24 – Принцип работы монитора на электронно-лучевой трубке

Для вывода цветного изображения используется матричная структура, установленная перед люминофором. Электронно-лучевая трубка направляет поток электронов на один из участков матрицы, воспроизводящих базовый цвет, например, красный, зеленый или синий, после чего близко расположенные участки матрицы воспринимаются как один пиксель, цвет которого определяется смешением базовых цветов.

При высокой разрешающей способности и хорошей цветопередаче мониторы на электронно-лучевой трубке имеют крупные размеры и высокий уровень вредных электромагнитных излучений, что заставило производителей переходить на другие технологии.

2.4.7.1.2 Жидкокристаллический монитор

Принцип получения изображения в **жидкокристаллических мониторах** (Liquid Crystal Display, LCD) основан на том, что жидкие кристаллы способны изменять под действием электрического поля свою структуру и положение плоскости поляризации света, то есть управлять количеством проходящего через них светового излучения. Свет генерируется осветителем и проходит через поляризационные фильтры, окружающие слой жидких кристаллов. Цветное изображение получается с помощью применения цветных фильтров, в большинстве случаев используются красный, зеленый и синий фильтры модели RGB (рисунок 2.25).

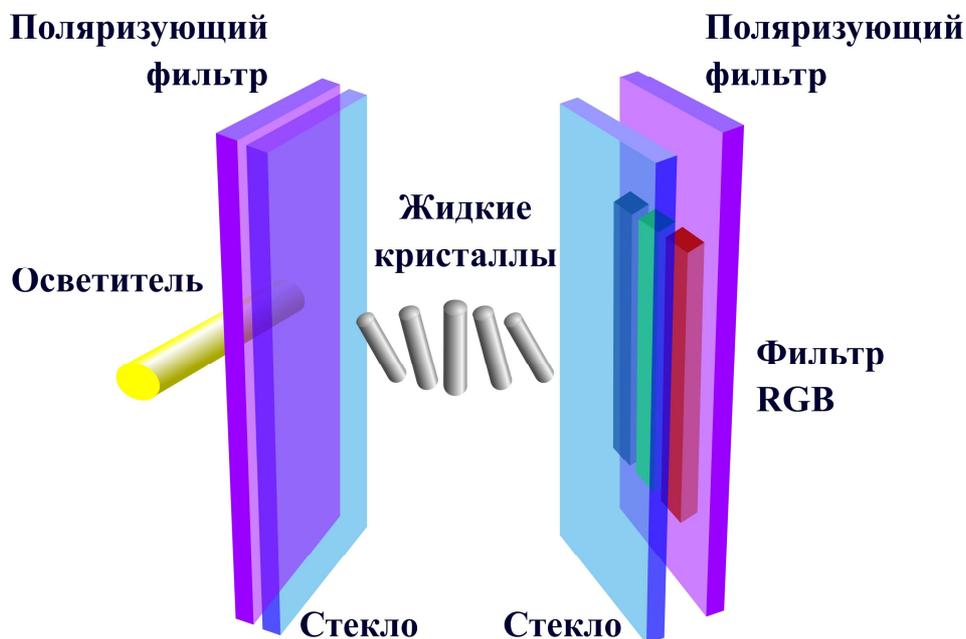


Рисунок 2.25 – Принцип работы жидкокристаллического монитора

Жидкокристаллические мониторы имеют компактные размеры и низкое потребление, практически не оказывают вредного воздействия на пользователя.

2.4.7.1.3 Плазменный монитор

Формирование изображения в **плазменном мониторе** (PDP, Plasma Display Panel) происходит в пространстве между двумя стеклянными пластинами, заполненном смесью газов.

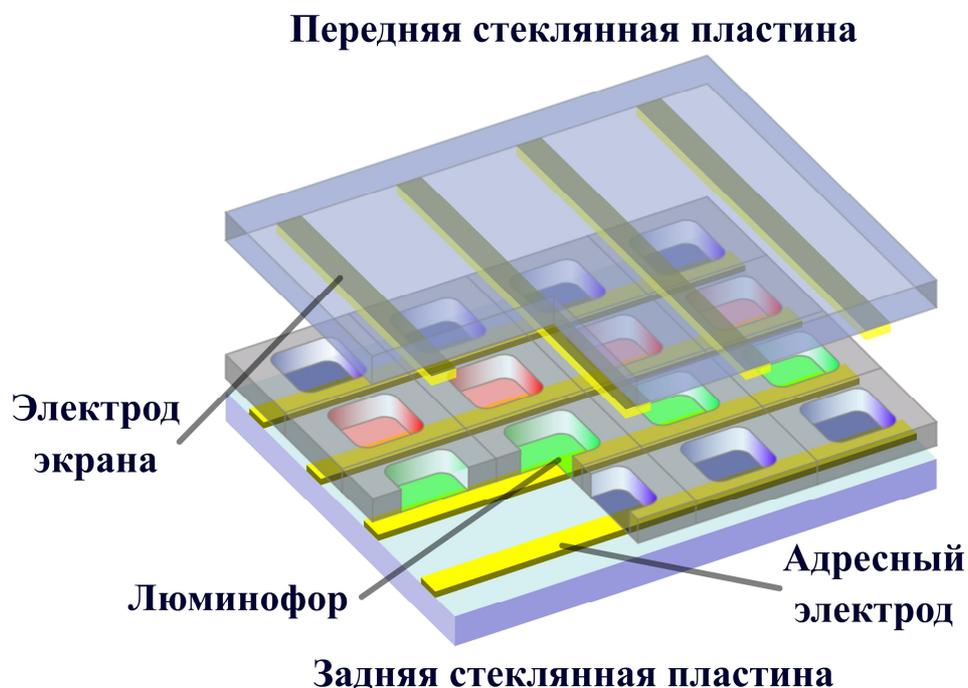


Рисунок 2.26 – Принцип работы плазменного монитора

На переднюю пластину нанесены тончайшие прозрачные электроды, на задней пластине расположены адресные электроды. Задняя пластина имеет микроскопические ячейки, заполненные люминофорами трех основных цветов – красного, синего и зеленого. При разряде смесь газов излучает ультрафиолетовый свет, который воздействует на люминофор, заставляя его светиться в видимом спектре (рисунок 2.26).

2.4.7.1.4 Органический светодиодный монитор

Органические светодиодные мониторы (OLED, Organic Light Emitting Diodes) создаются на основе тонкопленочных полимерных и молекулярных органических материалов, которые сами являются источниками светового излучения, что обеспечивает более высокий диапазон яркости и меньшее потребление энергии.

Органические светодиодные мониторы состоят из катода, анода и находящегося между ними органического материала, содержащего эмиссионный и проводящий слой. При подаче напряжения между катодом и анодом, происходит насыщение эмиссионного слоя электронами, а проводящего слоя – дырками,

которые под действием электростатических сил движутся навстречу друг другу и рекомбинируют, что сопровождается электромагнитным излучением в видимом спектре.

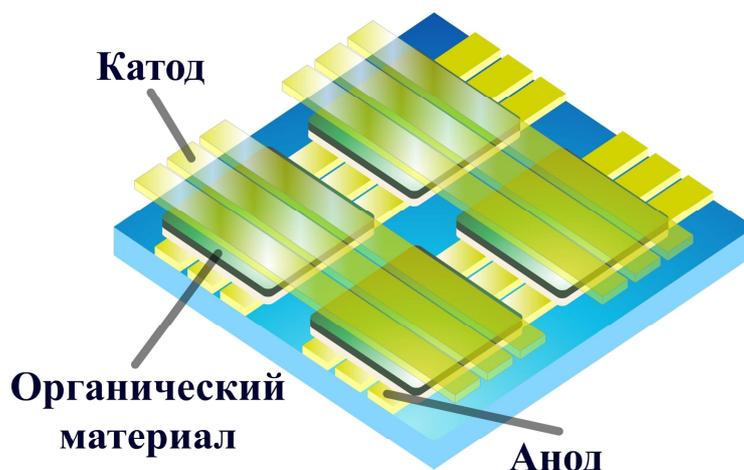


Рисунок 2.27 – Принцип работы органического светодиодного монитора с пассивной матрицей

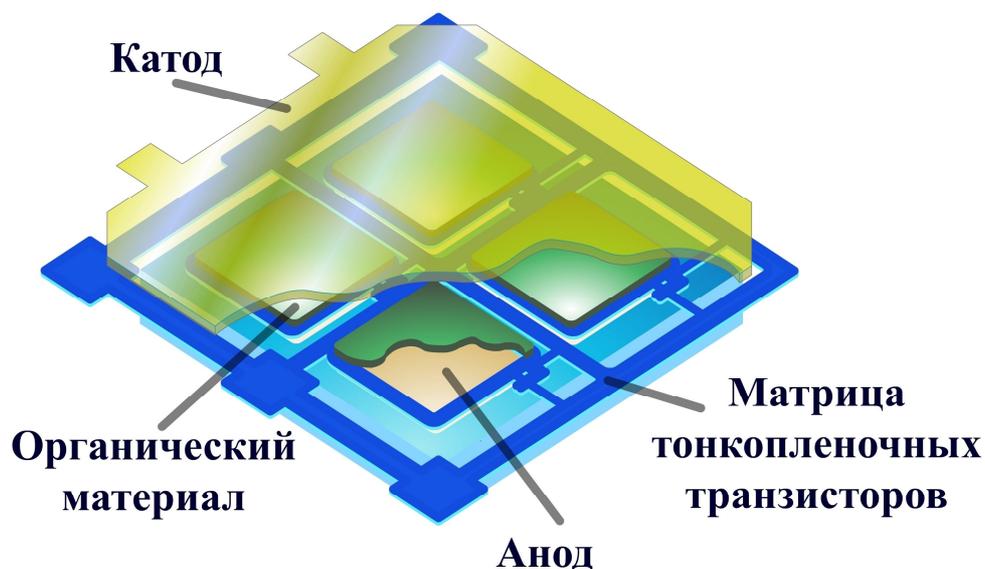


Рисунок 2.28 – Принцип работы органического светодиодного монитора с активной матрицей

Существуют органические светодиодные мониторы с пассивной матрицей и активной матрицей.

В органических светодиодных мониторах с пассивной матрицей для формирования яркости пикселей изображения необходимо последовательно подавать напряжение на все катоды и аноды, на пересечении которых органический материал излучает свет (рисунок 2.27).

В органических светодиодных мониторах с активной матрицей дополнительно используется матрица тонкопленочных транзисторов, которые позволяют одновременно управлять яркостью пикселей изображения, однако имеют сложную схему управления (рисунок 2.28).

2.4.7.1.5 Стереоскопический монитор

Стереоскопические мониторы могут быть созданы по одной из вышепредставленных технологий, однако должны обеспечивать высокое разрешение. Дополнительно к такому монитору поставляются стереоскопические очки. Современные модели обладают беспроводным интерфейсом подключения к компьютеру. Трехмерное изображение формируется за счет совместной работы стереоскопических очков и видеокарты компьютера путем чередования изображений для каждого глаза. Если на монитор выводится кадр, предназначенный для левого глаза, то очки закрывают правый глаз, и наоборот, если на монитор выводится кадр, предназначенный для правого глаза, то очки закрывают левый глаз. Пользователь при высокой частоте обновления экрана видит стереоскопическое изображение, то есть создается 3D-эффект.

2.4.7.2 Принтер

Принтером называют печатающее устройство, предназначенное для вывода графических данных на материальный носитель. Принтер выводит на печать растровое изображение, представляющее собой набор пикселей.

В принтерах используются различные технологии печати: матричные, струйные, лазерные, с термопереносом восковой мастики, с термосублимацией, с изменением фазы красителя.

Принцип действия **матричных принтеров** основан на перемещении каретки, на которой расположена матрица иглонок и красящая лента. Управляемые электромагнитом иглолки поочередно ударяют по ленте, краситель с которой переносится на бумагу.

Струйные принтеры основаны на мгновенном впрыскивании капли чернил через микроскопическое отверстие, ряды которых расположены на движущейся вдоль бумаги каретке. Для мгновенного впрыска чернил используются термическая или пьезоэлектрическая технология. При термической технологии капля чернил выталкивается пузырьком газа, образующимся при быстром нагреве капли специальным термоэлементом. При пьезоэлектрической технологии капля выталкивается за счет резкого изменения размеров пьезоэлемента при подаче на него напряжения.

В **принтерах с термопереносом восковой мастики** термопластичное красящее вещество, нанесенное на тонкую подложку, попадает на бумагу именно в том месте, где нагревательными элементами печатающей головки обеспечивается необходимая температура.

В принтерах с термосублимацией красителя переход красящего вещества из твердого состояния в газообразное происходит, минуя стадию жидкости. Таким образом, порция красителя сублимирует с подложки и осаждается на бумаге.

Восковые стерженьки в принтерах с изменением фазы красителя для каждого первичного цвета красителя постепенно расплавляются специальным нагревательным элементом и попадают в отдельные резервуары. Расплавленные красители подаются оттуда специальным насосом в печатающую головку, работающую обычно на основе пьезоэффекта. Капли воскообразного красителя на бумаге застывают практически мгновенно, обеспечивая необходимое с ней сцепление.

На сегодняшний день наиболее используемыми являются **лазерные принтеры**, обеспечивающие высокую скорость и качество печати. Принцип работы таких принтеров интересен оптотехникам, так как в них для формирования изображения на материальном носителе применяется лазер.

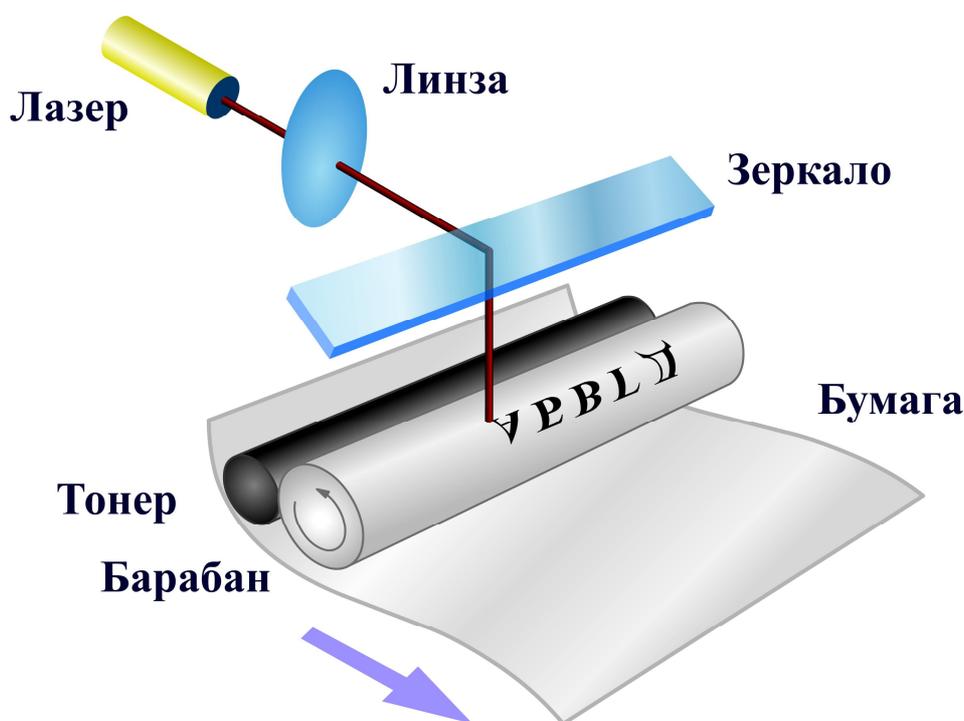


Рисунок 2.29 – Принцип работы лазерного принтера

Ключевым элементом лазерного принтера является барабан, обладающий свойством сохранять в любой точке своей поверхности электрический заряд. Луч лазера, попадая на барабан, воздействует на отдельные точки поверхности барабана, образуя на барабане изображение, состоящее из заряженных и незаряженных участков. Далее на барабан рассыпается тонер, который прилипает к заряженным участкам и осыпается с незаряженных участков. После этого прилипший к заряженным участкам тонер переносится на предварительно заря-

женную бумагу и поступает в узел закрепления, где под воздействием высокой температуры и давления тонер вплавляется в бумагу (рисунок 2.29).

Существует также светодиодная LED технология печати, являющаяся усовершенствованной лазерной технологией. Вместо одиночного лазерного диода используется набор светодиодов, количество которых определяет горизонтальное разрешение принтера. В таких принтерах, основанных на светодиодной технологии, отсутствует механическое управление источником света, что повышает надежность и скорость печати.

В 1951 г. Э. Мюллер предложил субтрактивную цветовую модель CMYK, состоящую из трех первичных цветов: сине-зеленого, пурпурного, желтого и одного дополнительного, черного цвета. Такая система имеет преимущества в полиграфии и цветной фотографии, так как основана не на прохождении света через светофильтр, а на его отражении от носителя данных. Поэтому в цветных принтерах используется именно цветовая модель CMYK.

2.4.7.3 Плоттер

Плоттером называют устройство, позволяющее представлять выводимые из компьютера данные на бумажный носитель в форме рисунка или графика. Изображение для плоттера хранится в виде массива описаний векторов. Для каждого вектора задается его направление, длина и атрибуты, к которым относятся цвет, толщина и тип линии. Поэтому плоттеры, в отличие от принтеров, являются векторными устройствами.

В зависимости от принципов построения пишущих головок различают фитильные, шариковые, карандашно-перьевые, струйные, пузырьковые, трубчатые, электростатические и лазерные плоттеры.

По принципу работы выделяют два типа плоттеров: планшетные и рулонные плоттеры. В планшетном плоттере лист бумаги укладывается на плоский стол и неподвижно закрепляется. Над столом в одном направлении перемещается каретка, вдоль которой перемещается пишущая головка, которая приводится в движение двумя шаговыми двигателями, обеспечивающими ее перемещение по всей поверхности листа. В рулонном плоттере имеется горизонтальный барабан, на который укладывается лист бумаги и прижимается к барабану валиками. Пишущая головка перемещается по направляющей только вдоль оси барабана. Направление вращения барабана перпендикулярно перемещению пишущей головки. Рулонные плоттеры позволяют выводить чертежи крупного формата, не занимая при этом большой площади.

Современные плоттеры, в которых применена технология струйной печати, все же являются растровыми принтерами, головка которых имеет ряд сопел. При выводе бумага прокатывается в них по барабану всего один раз, в одном направлении, и за этот проход растровым способом выводится все изображение. Растеризация изображения производится во внутренней памяти плоттера.

Большинство современных плоттеров управляются с помощью графического языка программирования HPGL (Hewlett Packard Graphics Language), яв-

ляющегося стандартом в промышленности и совместимого с многими прикладными программами. Помимо рисующих плоттеров существуют и режущие плоттеры, которые называют катерами. В таких плоттерах вместо пишущей головки применяется режущая головка с механическим или лазерным резакром. В качестве листового материала применяется кожа, ткань, винил или бумага.

Тезаурус

Поколения ЭВМ, архитектура компьютера, интерфейс, контроллер, порт ввода-вывода, протокол, аппаратная конфигурация, материнская плата, чипсет, микропроцессор, адресная шина, шина данных, шина управления, кэш-память, регистр, оперативная память, латентность, видеокарта, жесткий диск, винчестер, сектор, дорожка, цилиндр, флэш-память, клавиатура, мышь, сканер, разрешение, интерполяция, веб-камера, монитор, принтер, плоттер.

Контрольные вопросы

1. Назовите базовый элемент второго поколения ЭВМ.
2. В чем измеряется производительность суперкомпьютеров?
3. Каково назначение чипсета материнской платы?
4. Опишите внутреннюю организацию современного микропроцессора.
5. Перечислите основные параметры микросхем оперативной памяти.
6. Какова основная функция контроллера атрибутов видеокарты?
7. Как происходит чтение данных с компакт-диска?
8. Каким образом определяется направление перемещения мыши?
9. В чем отличие аппаратного разрешения сканера от оптического?
10. Опишите принцип работы лазерного принтера.

Список рекомендуемой литературы

1. Воройский, Ф. С. Информатика : энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах [Текст] / Ф. С. Воройский. – М. : ФИЗМАТЛИТ, 2006. – 768 с.
2. Гук, М. Ю. Аппаратные средства IBM PC. Энциклопедия [Текст] / М. Ю. Гук. – СПб. : Питер, 2006. – 1072 с.
3. Информатика : базовый курс [Текст] / под ред. С. В. Симоновича. – СПб. : Питер, 2009. – 640 с.
4. Полунов Ю. Л. От абака до компьютера: судьбы людей и машин : книга для чтения по истории вычислительной техники [Текст] / Ю. Л. Полунов. – М. : Русская редакция, 2004. – 480 с. – т. 1.
5. Соболев, Б. В. Информатика [Текст] / Б. В. Соболев, А. Б. Галин, Ю. В. Панов. – Ростов н/Д : Феникс, 2007. – 446 с.
6. Фигурнов, В. Э. IBM PC для пользователя [Текст] / В. Э. Фигурнов. – М. : ИНФРА-М, 2006. – 638 с.
7. Эспинуолл, Дж. Железо ПК. Трюки. 100 советов и рекомендаций профессионала [Текст] / Дж. Эспинуолл. – СПб. : Питер, 2005. – 256 с.

ГЛАВА 3 ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

В настоящей главе рассматриваются вопросы, связанные с системным и служебным программным обеспечением, их назначением и структурой. Описывается классификация и основные возможности операционных систем, раскрываются методы обработки текстовой и графической информации, методы работы с электронными таблицами, презентациями и базами данных.

3.1 Понятие программного обеспечения

Работа компьютера невозможна без программного обеспечения. Состав программного обеспечения компьютера называют **программной конфигурацией**. В программной конфигурации существует межпрограммный интерфейс, при котором все программное обеспечение разделяется на несколько взаимодействующих между собой уровней: базовое, системное, служебное, прикладное и инструментальное программное обеспечение (рисунок 3.1).



Рисунок 3.1 – Уровни программного обеспечения

Каждый последующий уровень программного обеспечения опирается на предыдущий уровень, при этом происходит расширение функциональных возможностей всей системы.

3.1.1 Базовый уровень программного обеспечения

Самым низким уровнем программного обеспечения является **базовый уровень**. Программы, составляющие базовый уровень, обеспечивают взаимодействие операционной системы с аппаратной частью компьютера. Как правило, базовое программное обеспечение входит в базовую конфигурацию компьютера и физически представляет собой микросхему, расположенную на материнской плате компьютера. Такие микросхемы называют постоянным запоминаю-

щим устройством. Запись программ в это устройство происходит в процессе производства материнской платы. Изменения в базовом программном обеспечении в процессе эксплуатации возможно только в случае использования перепрограммируемой микросхемы.

Основой базового программного обеспечения является базовая система ввода-вывода (Basic Input-Output System – BIOS), которая представляет собой программу, отвечающую за управление всеми компонентами, установленными на материнской плате. Кроме того, после каждого включения компьютера эта система производит тестирование всего установленного на материнской плате оборудования: проверку работоспособности системы управления электропитанием, инициализацию системных ресурсов и регистров микросхем, тестирование оперативной памяти, подключение клавиатуры и мыши, тестирование портов, инициализацию контроллеров, определение и подключение жестких дисков. В процессе тестирования базовая система ввода-вывода сравнивает системную информацию с данными, хранящимися в энергонезависимой памяти.

Важной функцией программного обеспечения базового уровня является загрузка операционной системы компьютера, которая может производиться со всех типов носителей.

3.1.2 Системный уровень программного обеспечения

Системный уровень программного обеспечения организует взаимодействие между базовым программным обеспечением, различными устройствами и программным обеспечением более высокого уровня, например, прикладными программами.

Системное программное обеспечение выполняет несколько очень важных задач, от которых зависит эксплуатационные характеристики всей вычислительной системы в целом. В первую очередь, системное программное обеспечение обеспечивает интерфейс между всеми ресурсами компьютера и пользователем. Под ресурсами в данном случае понимается весь комплекс программно-аппаратных средств, установленных на компьютере. Так, с помощью программ системного уровня реализуется ввод, редактирование, хранение, передача и вывод данных. Так же системное программное обеспечение выполняет функции по подключению и настройке нового оборудования и программ, поиска драйверов устройств и необходимых обслуживающих утилит.

Наличие системного программного обеспечения является обязательным условием работы компьютера. Основные программы, формирующие уровень системного программного обеспечения, называют **ядром операционной системы**.

3.1.3 Служебный уровень программного обеспечения

Программы, выполняющие работу по техническому обслуживанию вычислительной системы, называют **служебным программным обеспечением**,

или **утилитами**. Служебное программное обеспечение может взаимодействовать с системным программным обеспечением, в частности с операционной системой, непосредственно с пользователем компьютера или напрямую обращаться к базовому программному обеспечению.

Служебное программное обеспечение может быть интегрировано в операционную систему или быть независимой автономной программой, обслуживающей какой-либо ресурс компьютера.

Основная задача служебного программного обеспечения состоит в автоматизации работ по проверке, наладке и настройке компьютера. Классификация служебного программного обеспечения приведена на рисунке 3.2.



Рисунок 3.2 – Классификация служебного программного обеспечения

Диспетчеры файлов выполняют большинство операций, связанных с обслуживанием файловой структуры, к ним относятся копирование, перемещение и переименование файлов, создание и удаление файлов и каталогов, поиск файлов и навигация в файловой структуре.

Средства коммуникации обеспечивают работу пользователя с компьютерной сетью. Они позволяют устанавливать соединения с удаленными компьютерами, обслуживают передачу данных, обеспечивают пересылку факсимильных сообщений.

Средства диагностики необходимы для автоматизации процессов мониторинга и устранения неполадок компьютера. Такие программы выполняют необходимые проверки и выдают полученную информацию в наглядном виде. Так же с помощью этого программного обеспечения можно оптимизировать работу компьютера.

К средствам просмотра и воспроизведения относят универсальные программные средства, позволяющие просматривать документы различных видов.

Средства контроля представляют собой программы, позволяющие следить за процессами, происходящими в компьютерной системе. Такие программы часто называют мониторами. Наблюдение обычно ведется в реальном режиме времени, хотя возможен и контроль результатов с регистрацией в специальном файле.

Мониторы установки производят контроль над процессами установки и удаления программного обеспечения компьютера. Монитор установки любой программы при удалении сканирует системный реестр на наличие межпрограммных связей. Если такие связи существуют, и удаление программы влияет

на работоспособность системы, то монитор предупреждает об этом пользователя. Как правило, мониторы установки входят в состав операционной системы.

Архиваторы предназначены для создания архивов данных, это упрощает их хранение. При этом повышается эффективность использования носителя за счет того, что архивные файлы обычно имеют повышенную плотность записи информации. Архиваторы часто используют для создания резервных копий важных данных.

3.1.4 Прикладной уровень программного обеспечения

Прикладное программное обеспечение используется для решения конкретных задач из какой-либо предметной области. В связи с тем, что сегодня компьютеры применяют во всех областях деятельности человека, то и количество задач, решаемых с помощью прикладного программного обеспечения, велико. Следует отметить, что программное обеспечение этого уровня наиболее популярно и широко распространено среди пользователей компьютеров. В своей работе прикладное программное обеспечение опирается на системные программы, причем, чем более совершенна операционная система, тем больше функциональных возможностей имеется у прикладных программ. Классификация основных программ прикладного уровня приведена на рисунке 3.3.



Рисунок 3.3 – Классификация прикладных программ

Текстовые процессоры служат не только для создания и редактирования текстовых данных, но и для форматирования и преобразования текстовой информации. Текстовый процессор должен обеспечивать взаимодействия текста, графики, таблиц и других объектов, составляющих документ. К дополнительным функциям текстовых процессоров относят средства автоматической проверки орфографии, автоматического форматирования и рецензирования документов.

Электронные таблицы нацелены на преобразование больших массивов данных различных видов в соответствии с их структурой. Электронные таблицы предоставляют большое количество методов для обработки данных различных типов и поддерживают автоматическое изменение содержимого ячеек при вычислениях.

Графические редакторы предназначены для создания и обработки графических изображений. Графические редакторы разделяются на программы для обработки растровых и векторных изображений.

Электронные презентации необходимы для создания проектов, слайды которых поддерживают вставку текста, графической информации, видеофильмов и анимации. Такие программы применяются для визуализации результатов работы пользователей.

Системы управления базами данных предназначаются для создания, обработки и хранения больших массивов данных. Базы данных позволяют импортировать и экспортировать данные, обеспечивают возможность доступа к данным и предоставления средств поиска и фильтрации.

Представленная классификация, конечно же, не является полной, однако именно этот перечень прикладных программ обеспечивает качественную работу на персональном компьютере любого пользователя. Существует большое количество других узкоспециализированных программ, таких, как системы автоматизированного проектирования, браузеры, бухгалтерские системы, финансово-аналитические программы, геоинформационные системы, программы видеомонтажа и распознавания текста, математические пакеты.

3.2 Операционные системы

Важнейшим программным обеспечением компьютера является операционная система. **Операционная система** компьютера представляет собой комплекс взаимосвязанных программ, который действует как интерфейс между приложениями и пользователем с одной стороны и аппаратурой компьютера с другой стороны.

3.2.1 Основные понятия операционных систем

К главным функциям операционной системы относятся осуществление диалога с пользователем; ввод-вывод и управление данными; распределение ресурсов компьютера; запуск программ на выполнение; всевозможные вспомогательные операции; передача информации между различными внутренними устройствами компьютера; поддержка работы периферийных устройств (рисунок 3.4).



Рисунок 3.4 – Функции операционной системы

Каждая операционная система имеет свой командный язык. Для управления внешними устройствами компьютера используются **драйверы**, представ-

ляющие собой специальные системные программы. Драйверы стандартных устройств образуют в совокупности базовую систему ввода-вывода.

Операционная система состоит из ядра операционной системы и вспомогательных модулей.

Программы ядра выполняют базовые функции операционной системы: управление процессами, памятью, устройствами ввода-вывода. Эти функции наиболее часто используются, и скорость их выполнения определяет производительность системы в целом. Поэтому функции ядра постоянно находятся в оперативной памяти компьютера.

Вспомогательные модули операционной системы включают программы, решающие отдельные задачи управления и сопровождения системы, редакторы, компиляторы, компоновщики, отладчики, библиотеки процедур для упрощения разработки приложений.

Современная операционная система обладает несколькими характеристиками. Одной из таких характеристик является мобильность.

Под **мобильностью** понимают возможность переноса кода операционной системы от процессора одной модели к процессору другой модели.

Другой ключевой характеристикой операционной системы является **многозадачность**, представляющая собой способ организации вычислительного процесса, при котором на одном процессоре одновременно выполняются сразу несколько программ.

Многозадачность предназначена для повышения эффективности вычислительной системы: увеличение количества задач, выполняемых системой в единицу времени; повышение удобства работы пользователя в операционной системе.

Появление на рынке многопроцессорных систем сделало актуальным такую характеристику операционной системы, как мультипроцессорная обработка.

Мультипроцессорная обработка представляет собой способ обработки вычислительного процесса в системах с несколькими процессорами, при котором несколько задач могут одновременно выполняться на разных процессорах.

Существуют мультипроцессорные системы с симметричной и асимметричной архитектурой. Симметричная архитектура мультипроцессорной системы предполагает однозначность всех процессов и единообразие их включения в общую схему. Симметричные мультипроцессорные системы обычно разделяют одну большую оперативную память между процессорами, которые располагаются в одном корпусе. В асимметричной архитектуре разные процессы могут отличаться своими характеристиками и функциональной ролью. Требование единого корпуса для процессоров в данном случае отсутствует.

В операционных системах также используется многопоточковая обработка данных. **Поток** рассматривается как последовательный переход процессора от одной команды к другой, а операционная система в этом случае распределяет между потоками время и ресурсы компьютера.



Рисунок 3.5 – Состояния потока в многозадачной операционной системе

В многозадачной системе поток может находиться в одном из трех основных состояний: выполнение – состояние, во время которого поток обладает всеми необходимыми ресурсами и выполняется процессором; ожидание – состояние, при котором поток заблокирован по внутренним причинам; готовность – состояние, во время которого поток заблокирован по внешним причинам.

Переход между потоками организуется с помощью механизма прерываний. **Прерывание** переводит процессор на выполнение потока команд, отличного от того, который выполнялся до сих пор, с последующим возвратом к исходному потоку. Прерывание происходит в произвольной точке потока команд, которую нельзя спрогнозировать заранее.

Прерывания делятся на три класса: внешние – возникают в результате действий пользователя или поступления сигналов от устройств; внутренние – происходят при появлении аварийной ситуации в ходе исполнения некоторой инструкции; программные – возникают при выполнении особой команды процессора, которая имитирует прерывание.

3.2.2 Файловая система

Операционная система предоставляет пользователю упрощенную модель для работы с данными. Как правило, пользователь персонального компьютера не вникает в процессы создания, обработки или удаления данных, эти функции ложатся на операционную систему, которая, в свою очередь, перекладывает эти обязанности на файловую систему.

Файловая система отвечает за организацию хранения и доступа к данным на каких-либо носителях. Пользователь получает доступ к данным, не обращаясь к конкретным областям жесткого диска, он работает с другим укрупненным объектом хранения данных – файлом.

Файлом называют последовательность произвольного числа байтов, которая обладает уникальным собственным именем. Обычно в отдельном файле хранят данные, относящиеся к одному типу. В этом случае тип данных определяет **тип файла**.

Для операционной системы имя файла не имеет никакого значения, а важно, к какой области диска происходит обращение. Современные файловые системы поддерживают очень длинные имена, состоящие не только из буквенных символов, но знаков и цифр.

Все файлы в операционной системе расположены в каталогах. Различают два вида каталогов в файловой системе. С одной стороны, каталог представляет

собой файл, содержащий системную информацию о группе составляющих его файлов, такой каталог называют корневым. С другой стороны, каталог представляет собой группу файлов, объединенных пользователем по каким-либо личным соображениям. Каталог может содержать в себе каталоги более низкого уровня, таким образом, создаются уровни иерархии каталогов, а сама файловая система имеет иерархическую структуру.



Рисунок 3.6 – Состав файловой системы

Файловая система включает в себя совокупность всех файлов на диске. Также в состав файловой системы входят наборы служебных структур данных, используемых для управления файлами, например, каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске. Комплекс системных программных средств, реализующих управление файлами, в частности операции по созданию, уничтожению, чтению, записи, именованию файлов, установке атрибутов и уровней доступа, поиску, тоже является частью файловой системы.

В современных операционных системах возможна работа сразу с несколькими файловыми системами.

3.2.2.1 Файловая система FAT

Файловая система FAT (File Allocation Table) была разработана фирмой Microsoft в 1977 году для операционной системы MS DOS. Эта файловая система была предназначена для работы с гибкими дисками размером менее 1 Мбайт и не предусматривала поддержки жестких дисков. В настоящее время FAT поддерживает файлы и разделы размеров до 2 Гбайт.

Структура раздела файловой системы FAT изображена на рисунке 3.7. В блоке параметров BIOS содержится необходимая информация о физических характеристиках жесткого диска.



Рисунок 3.7 – Структура файловой системы FAT

Данная файловая система не может контролировать отдельно каждый сектор, поэтому она объединяет смежные сектора в **кластеры**. Это позволяет уменьшить общее количество единиц, обслуживаемых файловой системой. Размер кластера в FAT определяется при форматировании диска. **Кластер** представляет собой минимальное пространство, которое может занимать файл на диске. Это приводит к тому, что часть пространства диска расходуется впустую. В состав операционной системы должны входить различные утилиты, предназначенные для уплотнения данных на диске.

Ключевым понятием FAT является таблица размещения файлов. В этой таблице хранится информация о кластерах логического диска. Каждому кластеру в FAT соответствует отдельная запись, которая показывает статус кластера: свободен, занят данными файла или помечен как сбойный. Если кластер занят под файл, то в соответствующей записи в таблице размещения файлов указывается адрес кластера, содержащего следующую часть файла. Из-за этого FAT называют файловой системой со связанными списками.

Каждому файлу и подкаталогу в FAT соответствует элемент, содержащий имя файла, его атрибуты, например, архивный, скрытый, системный и «только для чтения», дату и время создания или внесения в него последних изменений, а также прочую информацию.

Файловая система FAT всегда заполняет свободное место на диске последовательно от начала к концу. Если необходимо создать новый файл или дополнить уже существующий, то файловая система ищет на диске первый свободный кластер в таблице размещения файлов. Если в процессе работы одни файлы были удалены, а другие изменились в размере, то появляющиеся в результате пустые кластеры будут рассеяны по диску. Если кластеры, содержащие данные файла, расположены не подряд, то файл оказывается фрагментированным. Сильно фрагментированные файлы значительно снижают эффективность работы, так как головки чтения-записи жесткого диска при поиске очередной записи файла должны будут перемещаться от одной области диска к другой. В состав операционных систем, поддерживающих FAT, обычно входят специальные утилиты дефрагментации диска, предназначенные повысить производительность файловых операций.

Производительность FAT сильно зависит от количества файлов, хранящихся в одном каталоге. Если в каталоге содержится большое количество файлов, то выполнение операции считывания списка файлов может занять значительное время. Это связано с тем, что в FAT каталог имеет линейную неупорядоченную структуру, и имена файлов в каталогах идут в порядке их создания. В результате, чем больше в каталоге записей, тем медленнее работают программы, так как при поиске файла требуется просмотреть последовательно все записи в каталоге.

Длина элементов FAT составляет 12, 16 или 32 байта. Разрядность FAT будет зависеть от емкости диска и размера кластера. FAT32 может использоваться с дисками значительно большей емкости, так как в ней обеспечивается меньшее отношение размера кластера к размеру раздела. Чем меньше размер

кластера, тем меньше места требуется для хранения файла и, как следствие, диск реже становится фрагментированным.

Хранение информации о файлах в таблице размещения файлов осуществляется следующим образом.

Например, файл требует для размещения два кластера. Операционная система может отметить каждый из двух элементов как занятый кластер, но при этом она не узнает, какой кластер первый, а какой второй. Трудно также определить, какому файлу принадлежат эти два кластера. Следовательно, простой отметки занятых кластеров недостаточно.

Предположим, что когда первый кластер выделяется файлу, можно сохранить номер кластера вне FAT. Тогда при увеличении файла и выделении ему второго кластера можно использовать элемент FAT для первого кластера, чтобы отметить, какой кластер выделен в качестве второго. Когда файл вновь увеличивается, требуя новый кластер, можно найти в FAT свободный кластер и поместить его номер в предыдущий элемент FAT.

Этот процесс можно продолжать до исчерпания свободных кластеров. Последний распределенный файлу кластер должен иметь специальное значение, показывающее, что после него нет кластеров. Такое значение должно представлять собой конец файла. Пример записи файла представлен на рисунке 3.8.

Таким образом, каждый кластер указывает на следующий кластер, занятый файлом, образуя цепь кластеров. Элемент, занятый файлом, содержит значение, указывающее на следующий кластер; элемент последнего кластера содержит индикатор конца файла. Теперь для нахождения всех кластеров файла нужно знать только номер первого кластера файла. Этот номер хранится в соответствующих данных корневого каталога.

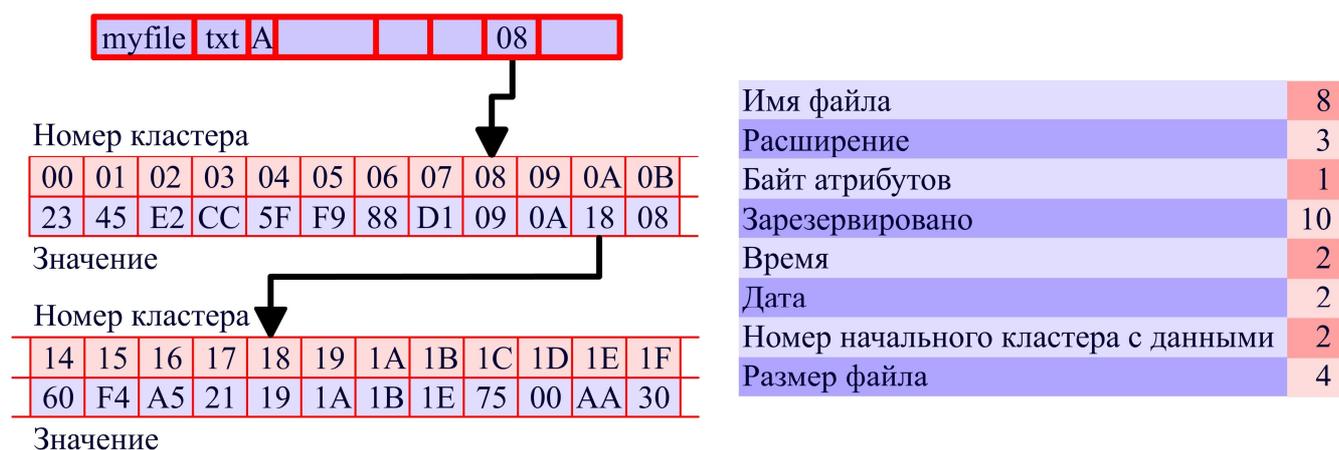


Рисунок 3.8 – Принцип работы файловой системы FAT

Файловая система FAT32 является усовершенствованной версией файловой системы FAT, поддерживающая жесткие диски объемом до 2 терабайт. В данный момент FAT32 поддерживается операционными системами Windows.

3.2.2.2 Файловая система NTFS

Файловая система NTFS (New Technology File System) разрабатывалась специально для операционной системы Windows NT, в ней значительно расширены возможности по управлению доступом к отдельным файлам и каталогам, введено большое число атрибутов, реализована отказоустойчивость, средства динамического сжатия файлов. NTFS обладает возможностью самостоятельного восстановления в случае сбоя операционной системы или оборудования, так что диск остается доступным, а структура каталогов не нарушается.

Наименьшей единицей работы файловой системы NTFS, так же, как и в FAT, является кластер, но в отличие от FAT файловая система NTFS поддерживает почти любые размеры кластеров, стандартом считается кластер размером 4 Кбайт.

Диск в файловой системе NTFS условно делится на две части. Первые 12% диска отводятся под главную файловую таблицу MFT (Master File Table), представляющую собой пространство, в котором хранятся метафайлы главной файловой таблицы. Запись каких-либо данных в эту область невозможна. MFT всегда держится пустой, это делается для того, чтобы главный служебный файл не фрагментировался при своем росте. Остальные 88% диска представляют собой обычное пространство для хранения файлов. Структура файловой системы NTFS изображена на рисунке 3.9.

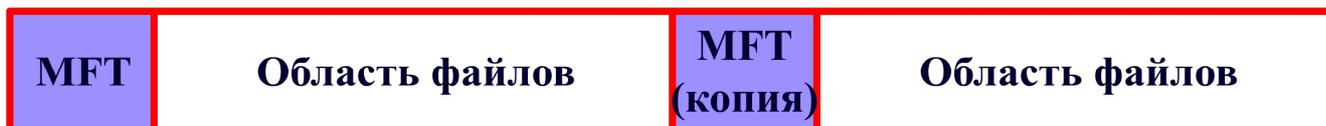


Рисунок 3.9 – Структура файловой системы NTFS

Свободное место диска включает в себя все свободное пространство, при этом учитывается и свободное место в главной файловой таблице. MFT используется следующим образом: если основного пространства для записи файлов недостаточно, то MFT сокращается, таким образом, освобождается место для записи файлов. Если место в основной области для хранения файлов освобождается, то MFT вновь расширяется.

Особенность файловой системы NTFS заключается в том, что каждый элемент системы представляет собой файл. В виде файла представлена также и служебная информация. Все существующие файлы в NTFS представлены записью в главной файловой таблице MFT. Первые 16 записей таблицы, что составляет около одного мегабайта дискового пространства, выделены для описания самой главной файловой таблицы. Далее размещается зеркальная копия MFT. Это необходимо в том случае, если первая запись MFT разрушена. Затем NTFS считывает вторую запись. Местоположение сегментов данных MFT и зеркального файла MFT хранится в секторе начальной загрузки. Копия сектора начальной загрузки находится в логическом центре диска. Третья запись MFT содер-

жит файл регистрации, применяемый для восстановления файлов. Семнадцатая и последующие записи главной файловой таблицы используются файлами и каталогами.

Устройства самого файла в NTFS также имеет некоторые особенности. Вся информация в файле представлена в форме потоков. Первый поток служит для хранения непосредственно данных файла, остальные потоки файла содержат его атрибуты. Такая структура файла позволяет свободно прикрепить к файлу дополнительную информацию, например, имя автора файла или его содержание.

Корневой каталог в NTFS представляет собой специальный файл, хранящий ссылки на другие файлы и каталоги. Данные на диске имеют иерархическую структуру. Файл каталога поделен на блоки, каждый из которых содержит имя файла, базовые атрибуты и ссылку на элемент MFT. Внутренняя структура каталога представляет собой бинарное дерево. Бинарное дерево располагает имена файлов таким образом, чтобы поиск файла осуществлялся более быстрым способом – с помощью получения двухзначных ответов на вопросы о положении файла. Вопрос, на который бинарное дерево способно дать ответ, таков: в какой группе, относительно данного элемента, находится искомое имя – выше или ниже? Файловая система обращается с таким вопросом к среднему элементу, и каждый ответ сужает зону поиска в среднем в два раза. Область поиска, суженная в два раза, начинается исследоваться аналогичным образом, начиная опять же со среднего элемента. Это позволяет ускорить процесс поиска файла в несколько раз.

Одним из преимуществ файловой системы NTFS является наличие **журнала транзакций**. В журнале транзакций регистрируются все операции, влияющие на структуру дискового пространства, включая создание файла и прочие команды, изменяющие структуру каталогов. Журнал транзакций применяется для восстановления NTFS после сбоя системы. NTFS является восстанавливаемой файловой системой, для этого используется модель обработки транзакций. Смысл такой обработки заключается в том, что каждая операция ввода-вывода, изменяющая файл в системе NTFS, рассматривается системой как транзакция и может выполняться как неделимый блок. При модификации файла пользователем сервис регистрации фиксирует всю информацию, необходимую для повторения или отката транзакции. Если транзакция завершена успешно, производится модификация файла. Если нет, NTFS производит откат транзакции. Такой механизм защиты файловой системы является устойчивым к различным экстремальным ситуациям, например, запись информации на поврежденный участок диска или отключение питания в момент записи файла.

NTFS позволяет хранить файлы размером до 16 эксабайт. Все файлы хранятся на диске в сжатом виде, средство сжатия файлов встроено в файловую систему. Процесс сжатия и распаковки файлов происходит в реальном режиме времени. Сжатие является атрибутом файла и по желанию пользователя может быть снято с файла или каталога. При уплотнении файла применяется пофайловое уплотнение, таким образом, порча небольшого участка диска не приводит к

потере информации в других файлах. Сжатие файлов имеет очень высокую скорость, поэтому пользователь его не замечает. Недостатком такой организации хранения информации является огромная виртуальная фрагментация сжатых файлов.

3.2.3 Классификация операционных систем

Единой классификации операционных систем не существует. Наиболее общая классификация представлена на рисунке 3.10.



Рисунок 3.10 – Классификация операционных систем

Деление на клиентскую и серверную операционные системы является условным. Производители операционных систем стремятся выпускать общий дистрибутив, содержащий как серверную, так и клиентскую части. Однако существуют отдельные системы, например, Windows 2003 Server.

Большинство современных операционных систем распространяется за деньги, а часто продается вместе с оборудованием. Существуют бесплатные операционные системы, создатели которых базируются на принципах свободного доступа к коду, например, операционная система Linux.

По признаку адаптации операционные системы разделяются на оригинальные и локализованные версии. Оригинальная версия подразумевает, что операционная система работает с языком, на котором была создана изначально. Если операционная система адаптирована под какую-то страну, например, операционная система переводится с английского языка на русский, такую операционную систему называют локализованной.

Существуют два интерфейса операционной системы: текстовый интерфейс и графический интерфейс. Текстовый интерфейс представляет собой командную строку, с помощью которой организуется взаимодействие пользователя с операционной системой. Графический интерфейс обеспечивает взаимодействие пользователя с системой с помощью окон, значков и рисунков. Основная работа в такой системе идет с помощью мыши.

Однозадачные операционные системы в один момент времени могут работать только с одной программой, многозадачные же операционные системы могут одновременно работать с несколькими приложениями.

Операционные системы также делятся на однопользовательские и многопользовательские. Первые из них позволяют работать с операционной системой лишь одному пользователю, вторые – нескольким, с различными правами и привилегиями.

Архитектура операционных систем предопределяет количество информации, которое система может обрабатывать в единицу времени. Существуют 16-разрядные, 32-разрядные и 64-разрядные операционные системы.

3.2.4 Обзор современных операционных систем

Компания w3schools (<http://www.w3schools.com>) ежемесячно публикует рейтинг операционных систем, который строится по данным о количестве персональных компьютеров, подключенных к глобальной сети Интернет и использующих ту или иную операционную систему (рисунок 3.11). Эта информация позволяет отслеживать изменение популярности систем, а также выявлять тенденции их дальнейшего развития. Дадим характеристику наиболее востребованным операционным системам.

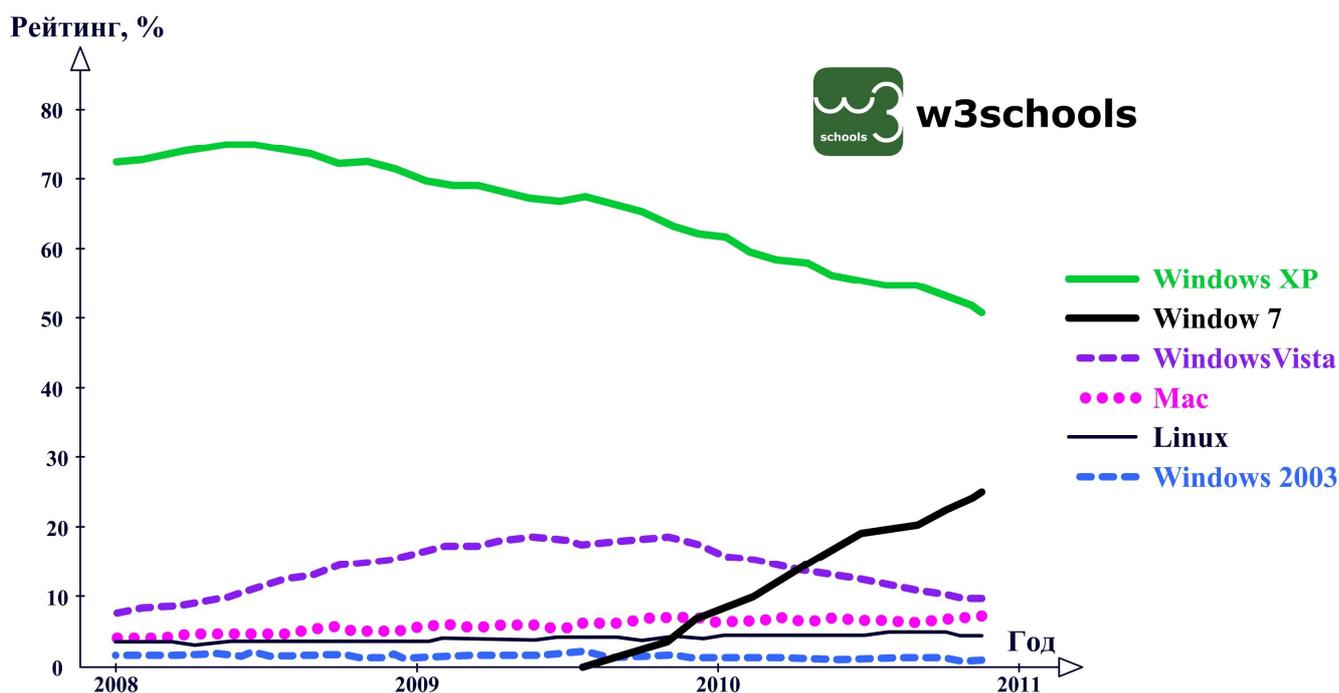


Рисунок 3.11 – Рейтинг операционных систем

Самой популярной операционной системой на сегодняшний день является **Windows XP**, именно ей отдают предпочтение больше половины пользователей персональных компьютеров. Операционная система Windows XP выпущена в 2001 году, ядро этой системы создано на основе Windows NT/2000, это

делает ее значительно более эффективной, безопасной и стабильной в работе. Windows XP использует файловые системы FAT16, FAT32 и NTFS. Существует две клиентских версии данной системы: Windows XP Professional и Windows XP Home Edition.

На настоящее время четверть всех пользователей применяют в своей работе операционную систему **Windows 7**, вышедшую в 2009 году. В этой операционной системе используется улучшенный графический интерфейс Windows Aero, включающий полупрозрачные окна, отображение всех открытых окон в двухмерном или трехмерном виде. По сравнению с предшествующими операционными системами Windows 7 имеет высокую производительность, позволяющую ей потреблять меньший объем памяти и выполнять фоновые службы только тогда, когда они требуются пользователю. Windows 7 быстрее запускает программы и затрачивает меньше времени на переход в спящий режим, возобновление работы и повторное подключение к беспроводным сетям. Данная операционная система является полностью 64-разрядной, что позволяет использовать все преимущества новейших технологий на мощных 64-разрядных компьютерах. Как видно из графика, Windows 7 быстро набирает популярность и, скорее всего, уже в следующем году превзойдет результат Windows XP по количеству пользователей.

Операционная система **Windows Vista** вышла в свет в 2006 году и была предназначена для корпоративных клиентов, принадлежащих к линейке операционных систем Windows NT. Windows Vista является 64-разрядной операционной системой. В Windows Vista используется трехмерный пользовательский интерфейс, усовершенствованный механизм поиска, распознавание речи, обеспечение максимальной безопасности информации пользователя. Однако Windows Vista обладает рядом существенных недостатков, которые не позволили ей занять место лидера в линейке операционных систем. Основным недостатком Windows Vista является повышенные требования к ресурсам вычислительной системы, в то время как производительность Windows Vista такая же, как и Windows XP; несовместимость с программным обеспечением; нехватка драйверов для устройств.

Операционная система **Mac** была представлена в 2001 году фирмой Apple. Она представляет собой многозадачную операционную систему с графическим пользовательским интерфейсом для персональных компьютеров Apple Macintosh. Mac включает в себя приложения для работы со шрифтами, просмотра PDF-документов и DVD-фильмов, редактирования текста, вычислений, управления принтерами, подключения к глобальной сети Интернет, изменения системных настроек и многие другие утилиты для настройки компьютера Macintosh.

Операционная система **Linux** была создана в 1991 году и базируется на следующих принципах: распространяется бесплатно, ее исходные коды полностью открыты; любой желающий может не только писать программы для этой операционной системы, но и модифицировать саму систему, как пожелает. Linux является современной Unix-подобной многопользовательской сетевой

операционной системой с оконной графической системой X Window System для персональных компьютеров и рабочих станций, которая выпускается как для 32-разрядных, так и для 64 разрядных компьютеров. Linux поддерживает стандарты открытых систем и протоколы глобальной сети Интернет. На персональных компьютерах Linux используется довольно редко, так как она сложна в установке и настройке.

Операционная система **Windows Server 2003** выпущена в 2003 году. Она предназначена для работы на серверах с поддержкой среды разработки приложений Visual Studio .NET 2003 и 64-разрядной версии системы управления базами данных SQL Server 2000 Enterprise Edition.

3.2.5 Принципы работы операционной системы Microsoft Windows XP

Операционная система Windows XP является графической операционной системой. Основными средствами управления этой системой является мышь и клавиатура. Стартовый экран Windows XP представляет собой системный объект, называемый «Рабочим столом». Рабочий стол является графической средой, в которой отображаются объекты и элементы управления Windows XP (рисунок 3.12).



Рисунок 3.12 – «Рабочий стол» Microsoft Windows XP

В исходном состоянии на «Рабочем столе» можно наблюдать несколько системных значков, например, «Мой компьютер», «Мои документы», «Сетевое

окружение», «Internet Explorer», «Корзина» и «Панель задач». **Значки** представляют собой графическое представление объектов Windows. **Ярлык** же является только указателем на объект, применение ярлыков позволяет экономить место на жестком диске. «Панель задач» является одним из основных элементов управления.

На панели задач находится кнопка «Пуск», которая позволяет производить управление работой компьютера. При нажатии на кнопку «Пуск» можно загрузить недавно использовавшиеся программы, получить доступ к системным объектам, запустить на выполнение любое программное обеспечение, установленное на компьютере.

В меню «Пуск» имеется важнейший элемент управления операционной системой Windows XP «Панель управления» (рисунок 3.13). С помощью «Панели управления» можно выполнить настройку внешнего вида элементов «Рабочего стола», экранных заставок, меню «Пуск» и «Панели задач». Можно настроить подключение к локальной сети или глобальной сети Интернет, установить или удалить программное обеспечение, подключить или отключить оборудование, изменить параметры учетных записей пользователей, отредактировать значение даты, времени и используемого языка.

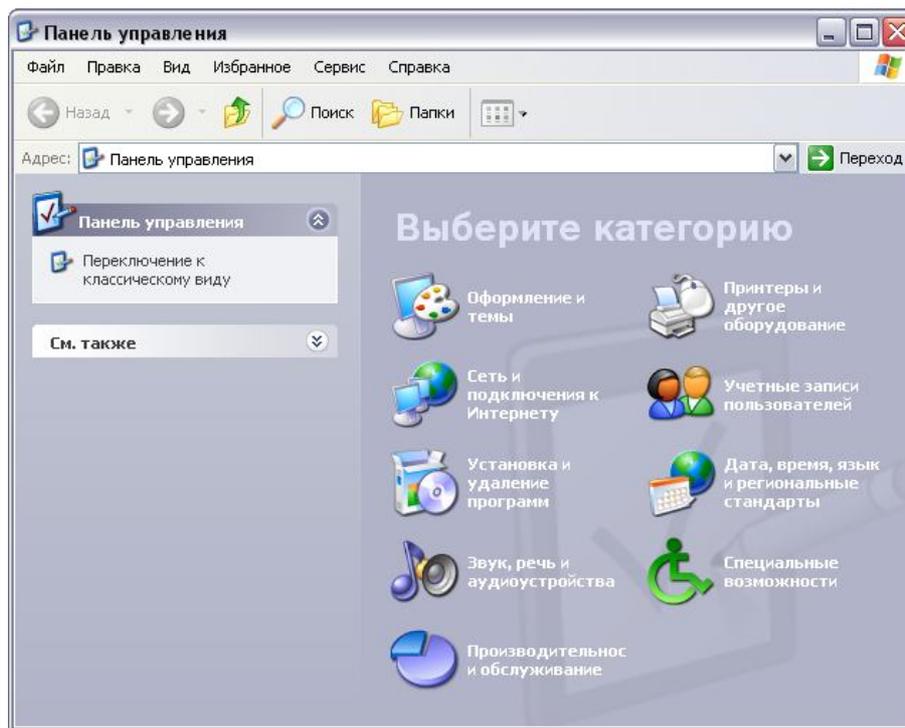


Рисунок 3.13 – «Панель управления» Microsoft Windows XP

Концепцией работы операционной системы Windows XP является использование окон, представляющих собой контейнеры, содержимое которых графически отображает содержимое папки. Любую папку можно открыть в своем окне. Любые операции с файлами и папками можно осуществить либо на «Рабочем столе», либо в каком-либо окне.

В состав любого окна Windows входят строка заголовка, кнопки сворачивания, восстановления и закрытия окна, кнопка системного меню, строка меню, панель инструментов, полосы прокрутки, рабочее поле и строка состояния.

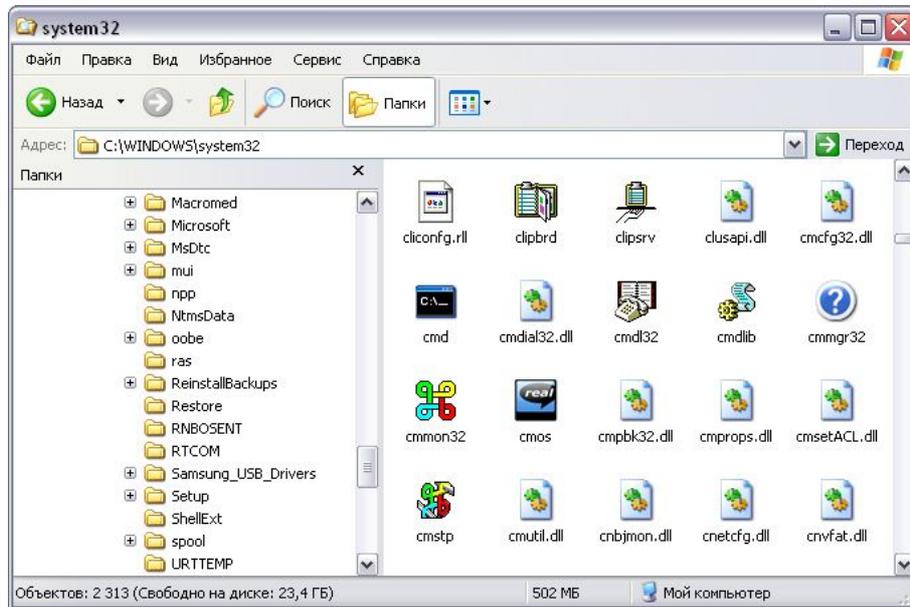


Рисунок 3.14 – Интерфейс программы «Проводник» Microsoft Windows XP

Для управления файлами и папками операционной системы Windows XP можно использовать стандартную программу «Проводник», которая относится к категории диспетчеров файлов. Окно «Проводника» имеет не одну рабочую область, а две: левую панель, называемую панелью папок, и правую панель, называемую панелью содержимого (рисунок 3.14). Навигацию по файловой структуре выполняют на левой панели «Проводника», на которой показана структура папок. Содержимое раскрытой папки отображается на правой панели. Вид рабочей области можно настраивать, указав размер значков или параметры сортировки. Более детально параметры файлов и папок можно установить с помощью окна «Свойства папки», запустив его из ниспадающего меню «Сервис» системного меню окна «Проводника».

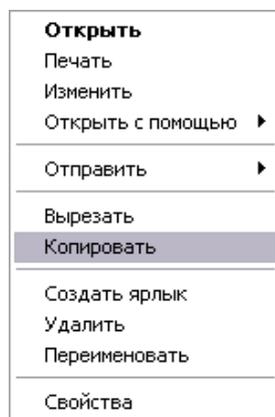


Рисунок 3.15 – Контекстное меню Microsoft Windows XP

Для любого объекта операционной системы Windows XP можно вызвать **контекстное меню**, которое содержит основной набор команд для работы именно с этим объектом. Для вызова контекстного меню нужно щелкнуть на объекте правой клавишей мыши. Пункты меню выбираются нажатием левой кнопки мыши. На рисунке 3.15 представлено контекстное меню файла, которое содержит определенный перечень команд: редактирование, переименование, копирование файла, можно отправить этот файл на другой носитель или распечатать.

Для каждого файла или папки можно получить окно его свойств, выбрав соответствующий пункт в контекстном меню. В этом окне отображается название и тип файла, стандартное приложение для обработки файлов такого типа, размещение файла на диске, а также его размер, дату создания и последнего изменения файла. У каждого файла имеются атрибуты, которые можно отредактировать в данном окне, например, сделать его скрытым или архивным.

3.3 Обработка текста с помощью Microsoft Office Word 2007

Текстовый редактор **Microsoft Office Word 2007** является одной из мощнейших программ по обработке текстовой информации, включающей в себя большой набор как стандартных, так и специализированных функций работы с текстом.

Интерфейс **Microsoft Office 2007** значительно отличается от предыдущих версий программы. Всю верхнюю часть окна любого приложения данного пакета занимает лента, кнопка «Office» и панель быстрого доступа. При выборе закладки ленты главного меню на ней размещаются инструменты, относящиеся к этой предметной области. При нажатии на кнопку «Office» появляется меню, предоставляющее возможность открыть, сохранить, распечатать документ **Microsoft Office** или произвести настройку приложения.

Для того чтобы создать новый документ в приложении Microsoft Office Word 2007, необходимо в меню кнопки «Office» выбрать элемент «Создание документа». Можно создать пустой новый документ или документ на основе готовых шаблонов. После этого можно начать работу с документом. Создадим документ Microsoft Office Word 2007, в котором приведем выдержки из учебника М. Борна и Э. Вольфа «Основы оптики» (рисунок 3.16).

После того как набран текст в документе Microsoft Office Word 2007, необходимо произвести его форматирование. Под **форматированием** понимают преобразование текста в вид, в котором он должен выводиться на экран монитора или печать. Наиболее часто процесс форматирования включает в себя редактирование шрифта, настройку абзацев, работу со стилями. Для форматирования текстовой информации нужно на ленте Microsoft Office на закладке «Главная» использовать группы «Шрифт», «Абзац» или «Стили».

С помощью элементов группы «Шрифт» можно изменять размер, тип и начертание шрифта, регистр текста, его цвет или цвет выделенного фрагмента.

Элемент «Очистить формат» позволяет удалять измененные параметры форматирования.

С помощью элементов группы «Абзац» можно настраивать выравнивание текста, устанавливать межстрочный интервал, уменьшать или увеличивать абзацный отступ, преобразовывать текст в нумерованный или маркированный список. Элемент «Отобразить все знаки» включает или выключает непечатаемые символы, что чрезвычайно полезно для выявления погрешностей форматирования.

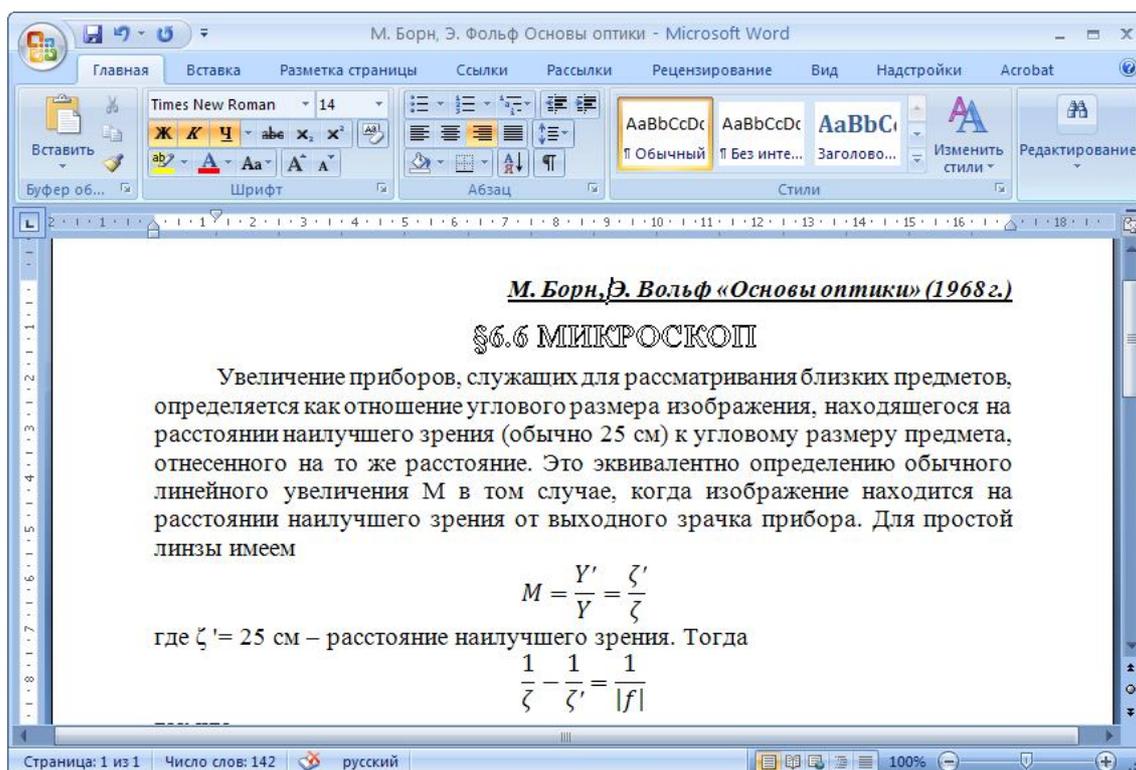


Рисунок 3.16 – Форматирование текста в Microsoft Office Word 2007

Параметры форматирования позволяют изменять вид создаваемых документов в широких пределах, однако число возможных вариантов форматирования столь велико, что установка всех этих параметров вручную может занимать значительное время. Для решения этой задачи используют стили. **Стили** представляют собой наборы команд форматирования. При создании стиля пользователь указывает значения отдельных параметров форматирования, которые должны быть включены в создаваемый стиль, для последующего применения всех этих параметров совместно посредством выбора имени этого стиля. Стили определяют форматирование символов, текстовых фрагментов, абзацев, строк таблиц или уровней структуры документа. Все доступные стили документа Microsoft Office Word 2007 можно получить с помощью элементов группы «Стили».

При форматировании текстовой информации полезно использовать на закладке «Главная» группы «Буфер обмена» элемент «Формат по образцу», кото-

рый переносит параметры форматирования указанного объекта на выделяемый фрагмент. Чтобы перенести все параметры форматирования на новый абзац необходимо сделать следующее. Во-первых, установить курсор в любом месте абзаца, параметры форматирования которого необходимо использовать. Во-вторых, нажать элемент «Формат по образцу». В-третьих, выделить текст, на который надо перенести форматирование.

В большинстве случаев необходимо установить параметры страницы, если стандартные установки не подходят для нового документа. Для этого нужно использовать группы «Темы», «Параметры страницы», «Фон страницы», «Абзац» и «Упорядочить», расположенные на закладке «Разметка страницы».

С помощью группы «Параметры страницы» можно установить книжную или альбомную ориентацию листа и его размер, поля документа, настроить параметры колонок, вставить разрывы страницы, раздела или колонки, расставить переносы в словах. С помощью группы «Фон страницы» настраивается цвет фона страницы и его границ. В группе «Абзац» можно указать значения перемещения левой или правой границы абзаца от края документа и определить интервал между абзацами.

Много функциональных возможностей по обработке текстовой информации представлено на ленте Microsoft Office на закладке «Вставка». Эта закладка содержит группы «Таблицы», «Иллюстрации», «Связи», «Колонтитулы», «Текст» и «Символы».

Рассмотрим пример использования таблиц в Microsoft Office Word 2007, которые представляют собой мощный инструмент форматирования текстовой информации. Для того чтобы вставить таблицу в документ, необходимо на закладке «Вставка» выбрать группу «Таблица», после чего указать количество строк и столбцов у создаваемого объекта. После выполнения этих действий необходимо выделить таблицу и приступить к ее форматированию. Особенностью Microsoft Office является динамическое добавление закладок на ленте Microsoft Office в зависимости от контекста, что делает обработку текстовой информации более удобной и качественной. Например, при работе с таблицами дополнительно появляются закладки «Конструктор» и «Макет» (рисунок 3.17).

На закладке «Конструктор» расположены группы «Параметры стилей таблицы», «Стили таблицы» и «Нарисовать границу». С помощью элементов этих групп можно применить к таблице один из существующих стилей форматирования, настроить цвет и тип границ таблицы. На закладке «Макет» имеются группы «Таблица», «Строки и столбцы», «Объединить», «Размер ячейки», «Выравнивание» и «Данные».

С помощью элементов этих групп происходит вставка новых столбцов и строк в таблицу, объединение и разбиение ячеек, настройка высоты и ширины ячеек, автоматическое выравнивание высоты группы строк или ширины группы столбцов, показ сетки таблицы, разбиение таблицы на две части. Также в таблице можно установить определенное выравнивание текста и его направление. При необходимости можно осуществить сортировку имеющейся в таблице информации или вставить в одну из ячеек формулу для выполнения простого рас-

чета, например, определения суммы в столбце или нахождения среднего значения в строке. В Microsoft Office Word 2007 имеется возможность преобразовать табличную информацию в текст.

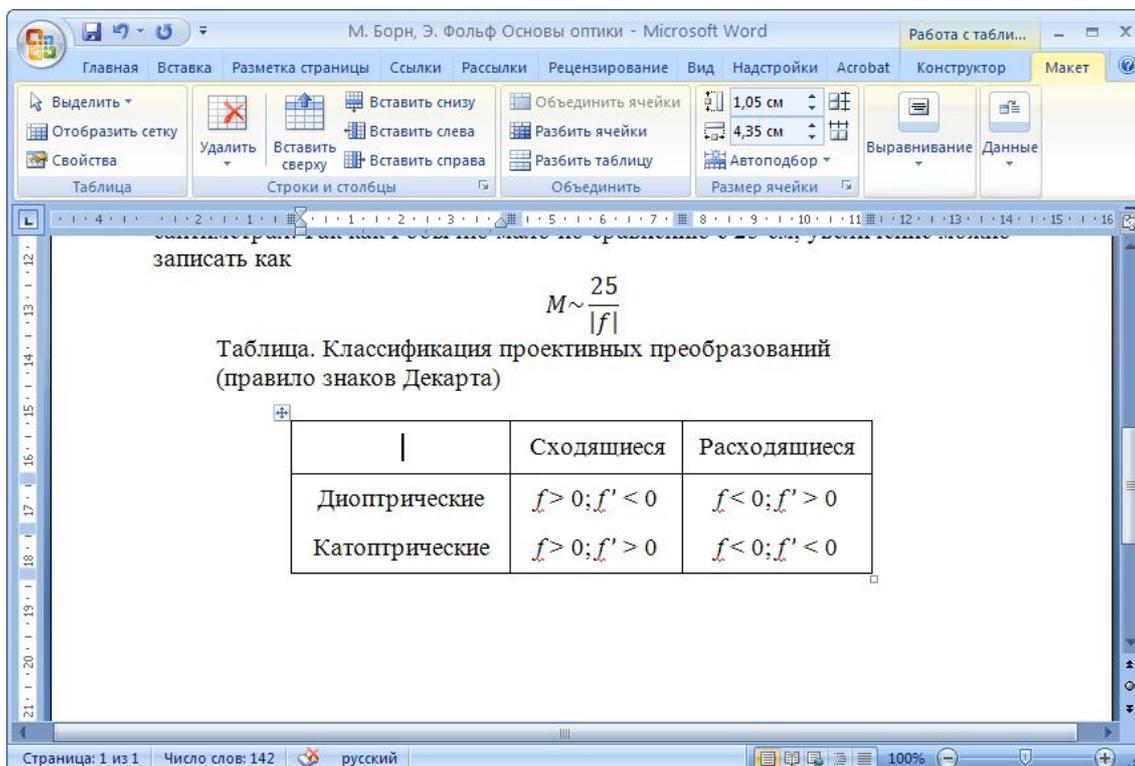


Рисунок 3.17 – Форматирование таблиц в Microsoft Office Word 2007

Многие пользователи, применяющие Microsoft Office Word 2007 в своей профессиональной деятельности, сталкиваются с потребностью использования формул для построения текстовых документов. Для того чтобы вставить формулы, необходимо на ленте Microsoft Office на закладке «Вставка» в группе «Символы» выбрать элемент «Формула». После этого в документе появляется объект для вставки формул, а на ленте Microsoft Office добавляется новая закладка «Конструктор», которая позволяет работать с этим объектом (рисунок 3.18).

Закладка «Конструктор» содержит группы «Сервис», «Символы» и «Структуры». С помощью элементов группы «Сервис» можно вставить стандартные заготовки формул, например, бином Ньютона, квадратное уравнение или разложение суммы, которые в процессе работы можно редактировать. Также в данной группе имеются элементы преобразования информации, находящейся в формулах, к линейному виду, когда все символы для упрощения записываются в одну строчку, или к профессиональному виду, который используется в научных текстах.

Элементы группы «Символы» позволяют вставлять в формулы специальные математические знаки, такие, как знаки арифметических операций, операций сравнения, знаки дифференциала или оператора Лапласа, знак существова-

ния или объединения. Также с помощью этой группы осуществляется вставка в формулы букв греческого алфавита.

Если для решения задачи недостаточно стандартных формул, описанных выше, можно использовать элементы группы «Структуры». С помощью этих элементов можно вставлять в формулу дроби, добавлять нижние и верхние индексы, использовать в формуле радикалы, интегралы, пределы и оператор суммирования. С помощью различных скобок можно разграничить информацию в формуле, чтобы сделать ее более читабельной. Имеется возможность вставлять математические операторы, матрицы и стандартные функции, что значительно сэкономит время пользователя при создании формул.

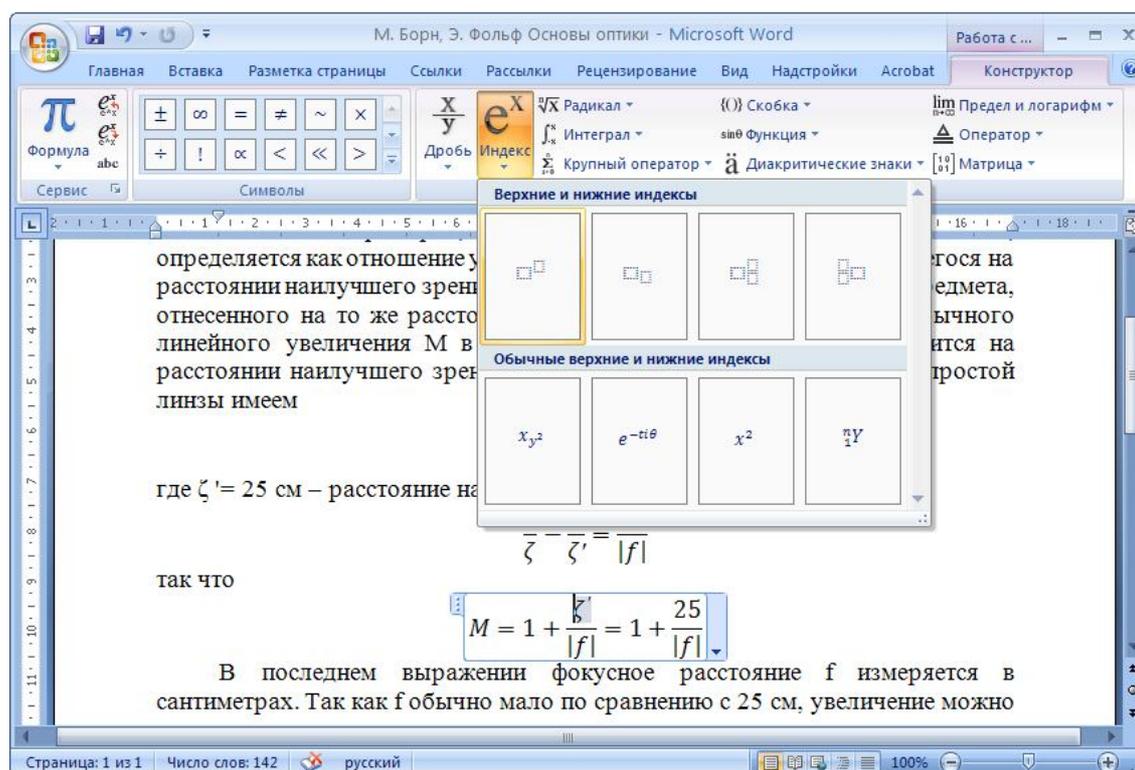


Рисунок 3.18 – Создание формул в Microsoft Office Word 2007

Из дополнительных возможностей форматирования наиболее часто используется вставка колонтитулов, в которых обычно размещаются номера страниц и текущая дата. При использовании колонтитулов в документе можно вставлять в них различную текстовую информацию для четных или нечетных страниц и для первой страницы документа.

Для того чтобы вставить колонтитул, необходимо на ленте Microsoft Office на закладке «Вставка» в группе «Колонтитулы» выбрать соответствующий элемент. После вставки колонтитул доступен для редактирования, при этом появляется закладка «Конструктор». Для вставки нумерации страниц необходимо использовать элемент «Номер страницы» закладки «Конструктор», после этого выбирается вариант размещения номера на странице и настраивается формат самого номера страниц.

3.4 Создание электронных таблиц с помощью Microsoft Office Excel 2007

Программа **Microsoft Office Excel 2007** является лидером среди средств создания и ведения электронных таблиц. Книга Microsoft Office Excel 2007 содержит несколько листов, на каждом из которых расположена своя таблица, состоящая из строк и столбцов, на их пересечении находится ячейка. Каждая ячейка имеет свой собственный адрес, равный сочетанию названия столбца и номера строки, например, на пересечении столбца F и строки 5 находится ячейка с адресом F5. В ячейки можно записывать данные различных форматов, например, числовые, текстовые, денежные или финансовые данные. Также в ячейках могут использоваться формулы для обработки значений других ячеек.

Интерфейс программы Microsoft Office Excel 2007 очень похож на интерфейс Microsoft Office Word 2007, хотя имеются отличия. Принципиальным отличием является присутствие под лентой Microsoft Office поля имени текущей ячейки и строки формул, а также наличие в рабочей области электронной таблицы.

Для того чтобы создать новую книгу в приложении Microsoft Office Excel 2007, необходимо в меню кнопки «**Office**» выбрать элемент «Создать». Можно создать пустую новую книгу или книгу на основе готовых шаблонов. После этого можно начать работу с книгой. Создадим электронную книгу, в которой произведем численный расчет закона Бугера-Ламберта-Бера (рисунок 3.19).

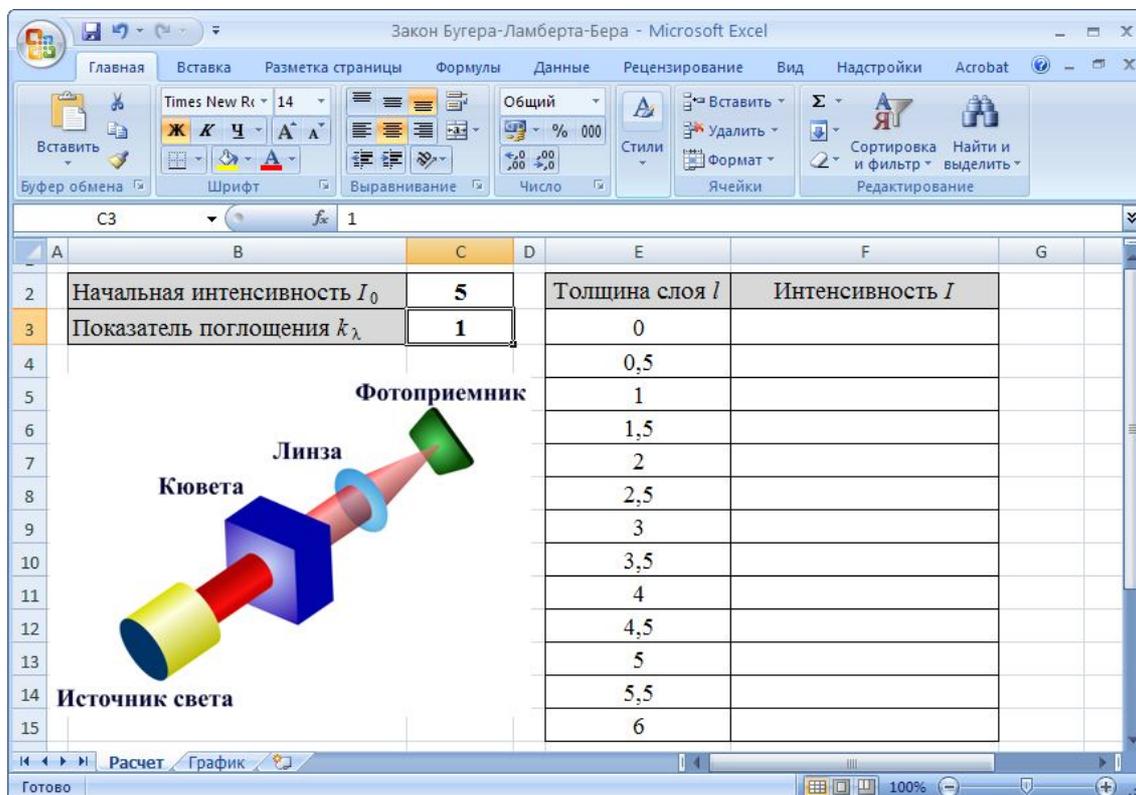


Рисунок 3.19 – Ввод информации в Microsoft Office Excel 2007

Работа с листом начинается с помещения первичных данных в соответствующие ячейки. Ввод любой информации осуществляется одним из двух способов. Первый способ заключается в выборе курсором мыши нужной ячейки, нажатии на клавиатуре кнопки F2 и ввода необходимой информации непосредственно в ячейку. Второй способ предполагает выбор ячейки с последующим вводом информации в строке формул. Аналогично можно редактировать информацию в ячейках. Так, например, в ячейки B2 и B3 помещаются текстовые значения «Начальная интенсивность I_0 » и «Показатель поглощения k_λ » соответственно, в ячейки C2 и C3 вводятся числовые значения этих величин. Ячейки E2 и F2 содержат текстовые значения «Толщина слоя l » и «Интенсивность I ».

Программа Microsoft Office Excel 2007 позволяет работать не только с одной ячейкой, но и с диапазоном ячеек, над которым можно выполнять групповые операции, например, суммирования, поиска максимального или минимального значения. Адрес диапазона ячеек определяется адресом левой верхней ячейки и адресом правой нижней ячейки, которые разделены символом «:», например, значения толщины слоя будут находиться в диапазоне E3:E15 и изменяться от 0 до 6 с шагом 0,5, то есть увеличиваться в арифметической прогрессии. Вводить эти значения можно вручную, однако, в Microsoft Office Excel 2007 имеется более эффективный способ ввода значений членов арифметической прогрессии. Для этого необходимо ввести первые два члена арифметической прогрессии, числовое значение 0 поместить в ячейку E3, числовое значение 0,5 – в ячейку E4. Далее необходимо выделить обе эти ячейки, навести курсор мыши на маркер автозаполнения, находящийся в правом нижнем углу выделенного фрагмента, затем нажать левую кнопку мыши и, не отпуская ее, переместить курсор мыши до конца диапазона.

После того, как все начальные данные введены, можно отформатировать рабочую область, хотя эти этапы можно выполнять в любой последовательности. Для этого на ленте Microsoft Office нужно выбрать закладку «Главная». Группа «Шрифт» аналогична по своим функциям соответствующей группе Microsoft Office Word 2007 и позволяет выполнять изменение размера, типа и начертания шрифта, регистра текста, его цвета или цвета выделенного фрагмента, а также тип, цвет и вид линии границы ячейки.

С помощью группы «Выравнивание» можно задавать выравнивание информации в ячейках по горизонтали и по вертикали. Также можно определять ориентацию текста в ячейке, уменьшить или увеличить отступ текста от левого края и объединить несколько ячеек в одну.

Группа «Число» помогает установить нужный тип данных, хранящихся в ячейке, а также для некоторых форматов данных задать нужные параметры отображения информации, такие, как количество знаков после запятой для числового формата или денежная единица для денежного формата.

Главным преимуществом программы Microsoft Office Excel 2007 является поддержка выполнения всевозможных расчетов и вычислений с помощью большого количества встроенных функций. Это возможно благодаря применению формул. Несомненным удобством проведения вычислений в программе

Microsoft Office Excel 2007 является то, что в формулах используются не конкретные числовые значения, а адреса ячеек. Поэтому при изменении значения в какой-либо ячейке, адрес которой указан в формуле, автоматически изменится и результат вычисления формулы. **Формулой** называют математическое выражение, на основании которого вычисляется значение некоторой ячейки.

Созданную один раз формулу в Excel можно копировать из одной ячейки в другие ячейки с помощью функции автозаполнения. При этом происходит корректировка адресов ячеек в формулах. Если формулу скопировать из одной ячейки в другую, то и все адреса ячеек, используемые в формуле, изменятся. Адреса ячеек, которые корректируются при копировании формул, называются **относительными**. Для того чтобы адреса ячеек в формуле при копировании не изменялись, необходимо ставить перед именем столбца и номером строки символ «\$», например, $\$F\5 . В этом случае такие адреса называются **абсолютными**. Любая формула начинается с символа «=». В формуле можно использовать арифметические операции, числа, адреса ячеек и функции. Для вставки функций необходимо на ленте Microsoft Office выбрать закладку «Формулы». Эта закладка служит для удобного создания и использования формул в ячейках электронной таблицы и содержит группы «Библиотека функций», «Определенные имена», «Зависимости формул» и «Вычисления».

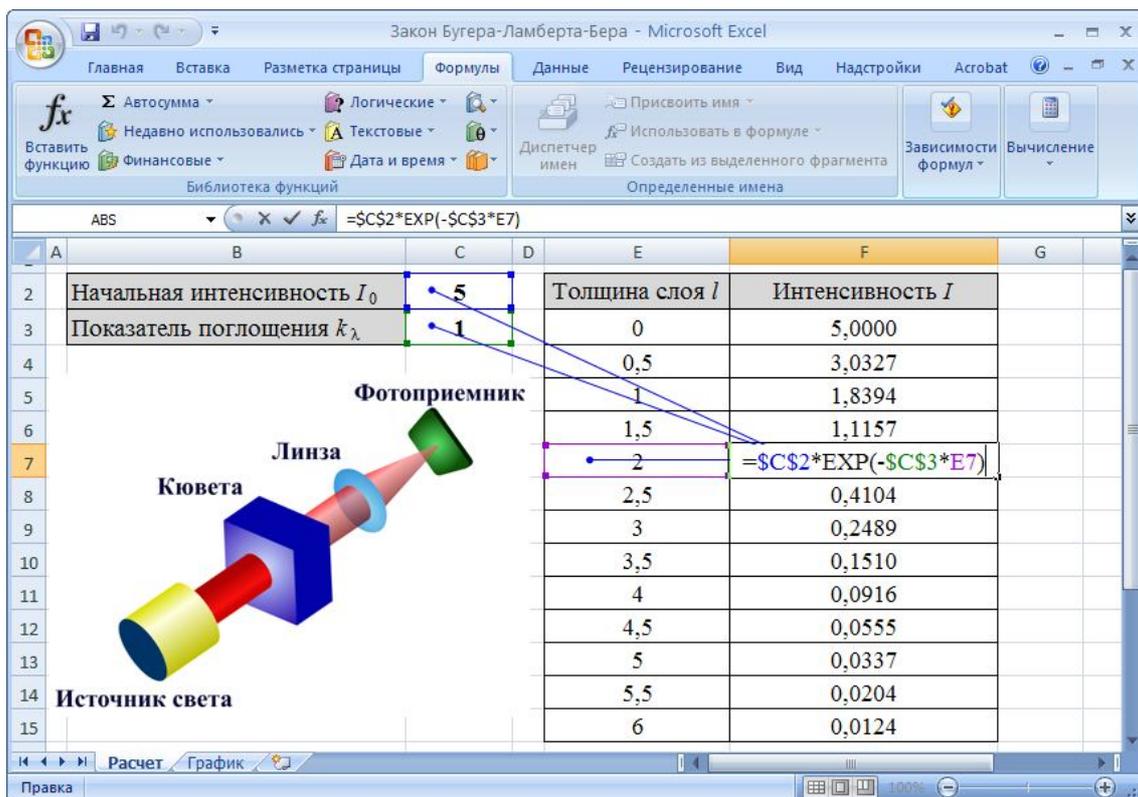


Рисунок 3.20 – Создание формул в Microsoft Office Word 2007

Установим курсор мыши в ячейку F7. В эту ячейку вставим функцию, возвращающую экспоненту заданного числа E7. Для этого в группе «Библиотека функций» запустим элемент «Вставить функцию», после чего в появившемся

списке выберем нужную категорию – математические функции, затем из списка всех математических функций выберем функцию EXP(), в качестве параметра этой функции укажем адрес ячейки E7 и нажмем кнопку «ОК». Однако в ячейке F7 после выполнения этой функции появится лишь значение экспоненты от заданной толщины слоя, чтобы результат соответствует закону Бугера-Ламберта-Бера, необходимо параметр экспоненты умножить на произведение минус единицы и показателя поглощения, значение которого хранится в ячейке C3, а саму экспоненту умножить на начальную интенсивность, значение которой расположено в ячейке C2. При копировании формулы определения интенсивности света, прошедшего через слой, необходимо учитывать, что значение толщины слоя должно изменяться, значение начальной интенсивности и показателя поглощения – нет. Следовательно, окончательно формула примет вид: = \$C\$2*EXP(-\$C\$3*E7). После этого формулу можно скопировать для вычисления интенсивности, соответствующей каждому значению толщины слоя (рисунок 3.20).

Если необходимо определить адреса ячеек, которые влияют на результат вычисления, необходимо выделить анализируемую ячейку, в группе «Зависимости формул» выбрать элемент «Влияющие ячейки».

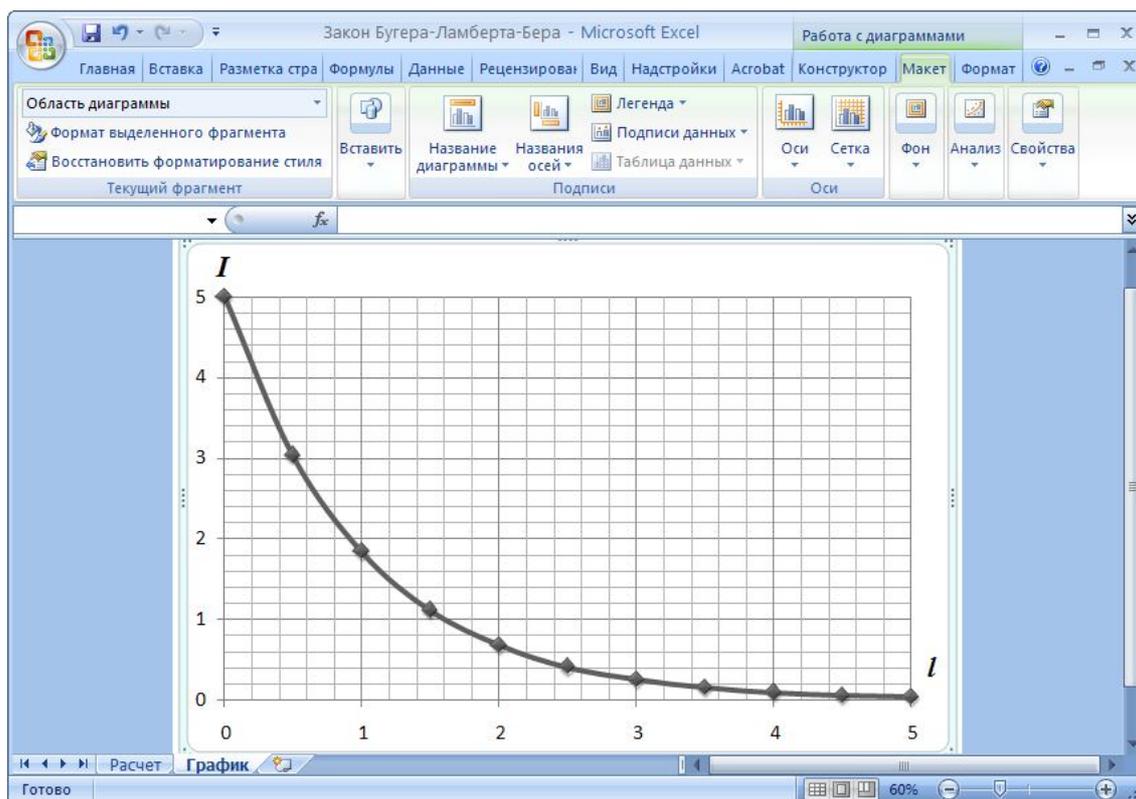


Рисунок 3.21 – Создание формул в Microsoft Office Word 2007

Многие расчеты требуют визуального представления данных с целью более качественного восприятия информации. Для этого в Microsoft Office Excel 2007 используются диаграммы с улучшенным графическим интерфейсом. Что-

бы вставить график зависимости интенсивности света от толщины слоя, необходимо сначала выбрать диапазон данных, затем на ленте Microsoft Office перейти на закладку «Вставка». В группе «Диаграмма» необходимо указать тип диаграммы, в нашем случае выбираем точечную диаграмму с прямыми отрезками, позволяющую сравнить пары значений: толщина слоя и интенсивность света. После этого диаграмма готова для форматирования (рисунок 3.21).

При выделении диаграммы на ленте Microsoft Office появляются дополнительные закладки: «Конструктор», «Макет» и «Формат».

На закладке «Конструктор» расположены элементы изменения типа диаграммы, изменения данных, определения стилей форматирования диаграмм, а также перемещения диаграммы в отдельный лист.

С помощью закладки «Макет» можно вставить в диаграмму рисунок, указать название диаграммы, название осей, добавить легенду, а также настроить вид осей, сетки, фона, вставить линию тренда или планки погрешностей. На закладке «Формат» находятся элементы настройки внешнего вида отдельных элементов диаграммы.

При описании программы Microsoft Office Excel 2007 необходимо упомянуть о закладке «Данные», которая находится на ленте Microsoft Office. Элементы этой закладки ориентированы на всевозможные операции с данными, содержащимися в ячейках таблицы: сортировки, применения фильтра, группировки. Кроме того, на закладке имеются инструменты, позволяющие перенести в таблицу данные из других приложений.

3.5 Создание презентаций с помощью Microsoft Office PowerPoint 2007

Программа **Microsoft Office PowerPoint 2007** представляет собой одно из лучших приложений по созданию электронных презентаций. **Презентацией** называют последовательно показываемые в полноэкранном режиме слайды, которые могут содержать информацию разных типов: текстовую, графическую и видеоинформацию.

Для того чтобы создать новую книгу в приложении Microsoft Office PowerPoint 2007, необходимо в меню кнопки «**Office**» выбрать элемент «Создать». Можно создать пустую новую презентацию или презентацию на основе готовых шаблонов. После этого можно начать работу с презентацией. Создадим электронную презентацию, в которой опишем средство оптического контроля вибрации.

Интерфейс программы Microsoft Office PowerPoint 2007 такой же, как и у остальных приложений Microsoft Office 2007. Главным элементом управления является лента Microsoft Office, рабочая область в обычном режиме разделена на две части. В левой части рабочей области приведен список всех слайдов, из которых состоит презентация. С помощью мыши можно выбрать нужный слайд и перейти на него. Выбранный слайд отобразится в центральной части рабочей области и будет доступен для редактирования.

Во вновь созданной презентации в рабочей области отображается простейший слайд, представляющий собой титульный лист презентации, в котором имеется заголовок слайда, подзаголовок слайда и заметки к слайду.

При начале работы с презентацией необходимо определиться с ее оформлением, чтобы грамотно расположить на форме все элементы. В программе Microsoft Office PowerPoint 2007 имеется большое количество тем дизайна, которые придадут презентации вид профессионального проекта. Для этого необходимо на ленте Microsoft Office выбрать закладку «Дизайн». На данной закладке в группе «Темы» можно выбрать понравившуюся тему, а также подходящую цветовую схему, шрифт и эффекты (рисунок 3.22). На этой закладке в группе «Параметры страницы» устанавливается ориентация слайда, и указываются размеры полей, а в группе «Фон» определяется стиль фона.

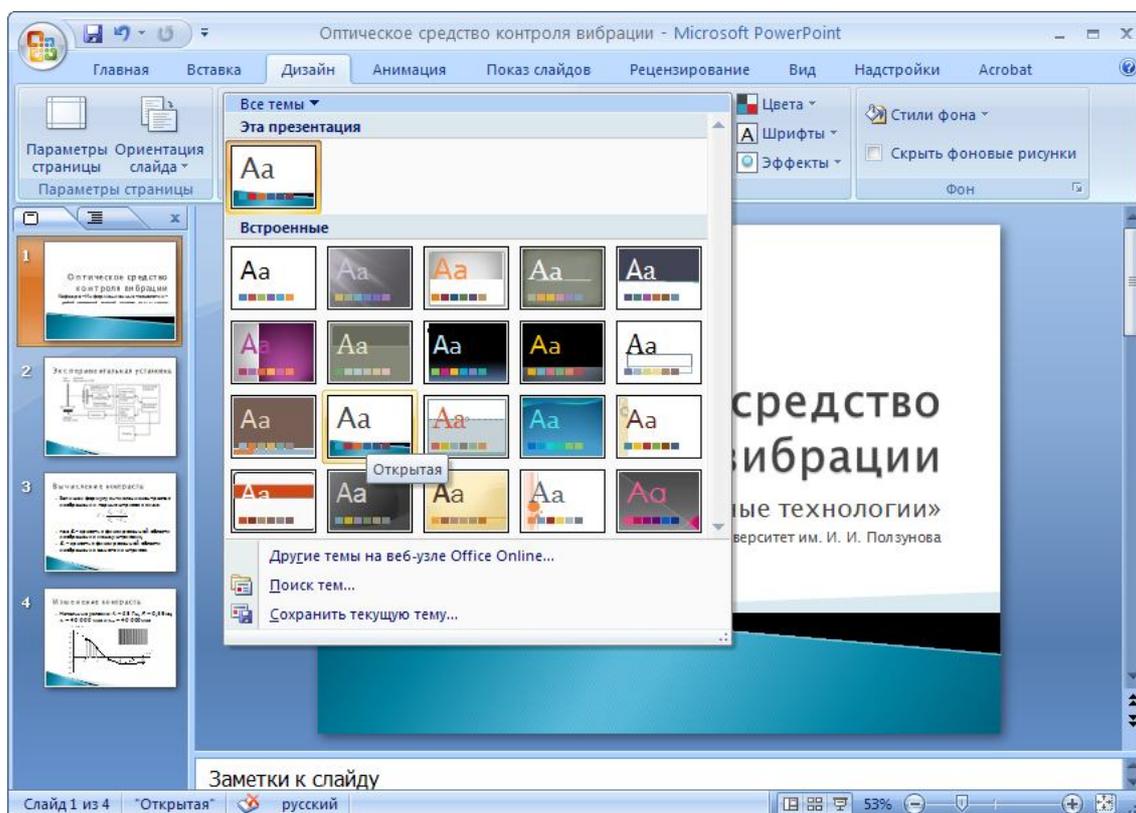


Рисунок 3.22 – Выбор темы в Microsoft Office PowerPoint 2007

После этого можно начать работу по оформлению титульного листа. Форматирование текста осуществляется с помощью групп закладки «Главная». На слайд можно добавить любую стандартную фигуру, в том числе и надпись, используя группу «Рисование». Фигуры можно редактировать, трансформировать, изменять стиль, упорядочивать и выравнивать (рисунок 3.23).

Если необходимо добавить новый слайд, то на закладке «Главная» ленты Microsoft Office в группе «Слайды» выбирается элемент «Создать слайд». В любой момент работы над презентацией можно изменить разметку текущего слайда. Для этого необходимо на ленте Microsoft Office перейти на закладку

«Главная» и в группе «Слайды» запустить элемент «Макеты», после чего в раскрывшейся галерее выбрать нужную компоновку слайда.

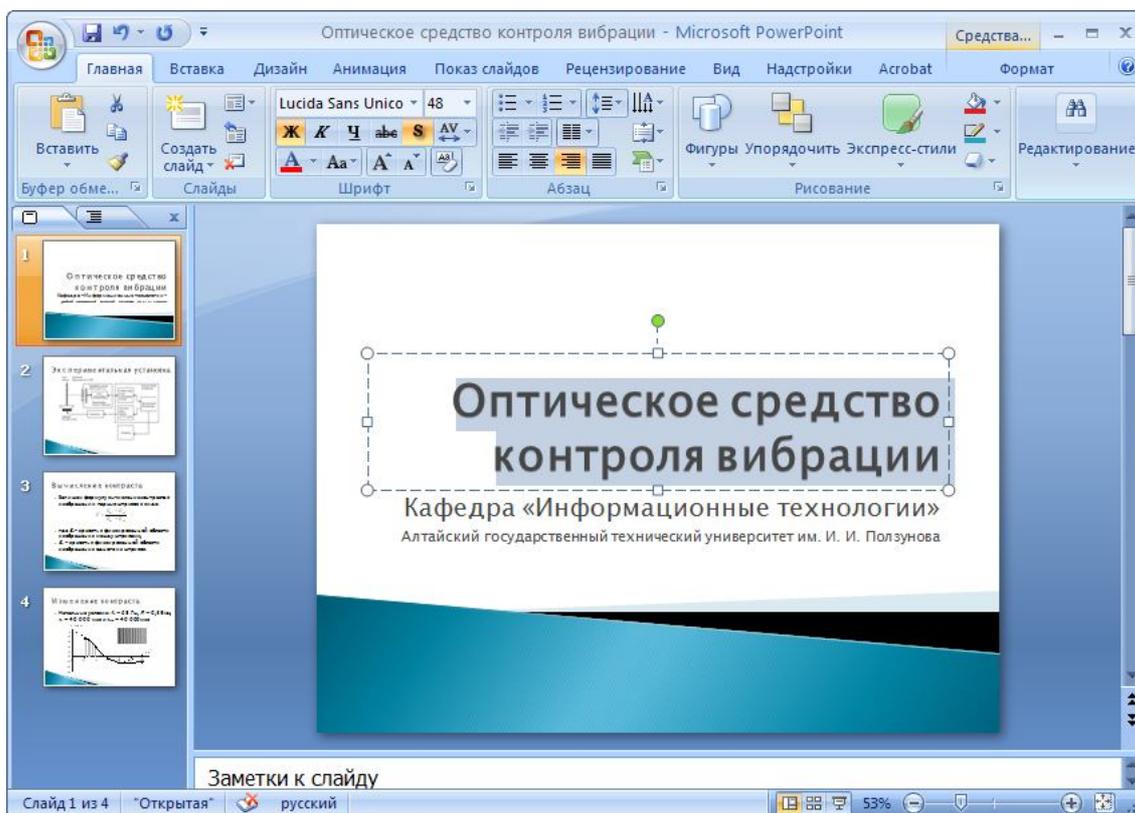


Рисунок 3.23 – Форматирование слайдов в Microsoft Office PowerPoint 2007

Кроме надписей и стандартных фигур в презентацию можно помещать таблицы, рисунки, клипы, фотоальбомы, диаграммы, видеофильмы и звуки.

Для того чтобы вставить в слайд презентации картинку, необходимо на ленте Microsoft Office перейти на закладку «Вставка» и в группе «Иллюстрации» запустить элемент «Рисунок», затем в открывшемся диалоговом окне выбрать графический рисунок на диске и нажать кнопку «Вставить» (рисунок 3.24). Если нужно произвести настройку изображений, то при выделении объекта на ленте Microsoft Office появляется дополнительная закладка «Формат», с помощью элементов которой настраивается стиль картинки, ее расположение и размер, яркость и контрастность. Технология вставки всех других объектов аналогична.

Характерной особенностью программы Microsoft Office PowerPoint 2007 является возможность создания анимации в презентации. Под **анимацией** понимают задание объектам, размещенным на слайдах, различных действий, которые будут с ними происходить при показе слайда. Анимационный эффект можно придать всему слайду или отдельным его элементам. Для задания анимационных эффектов используют закладку «Анимация» ленты Microsoft Office.

На данной закладке в группе «Переход к этому слайду» можно указать анимационный эффект, которым должно сопровождаться появление текущего

слайда. При выборе будет продемонстрирован этот эффект. Чтобы применить анимационный эффект к какому-либо отдельному элементу слайда, нужно сначала его выделить, а затем выбрать элемент «Настройка анимации» группы «Анимация». В правой части окна появится соответствующая панель с кнопкой «Добавить эффект» (рисунок 3.25).

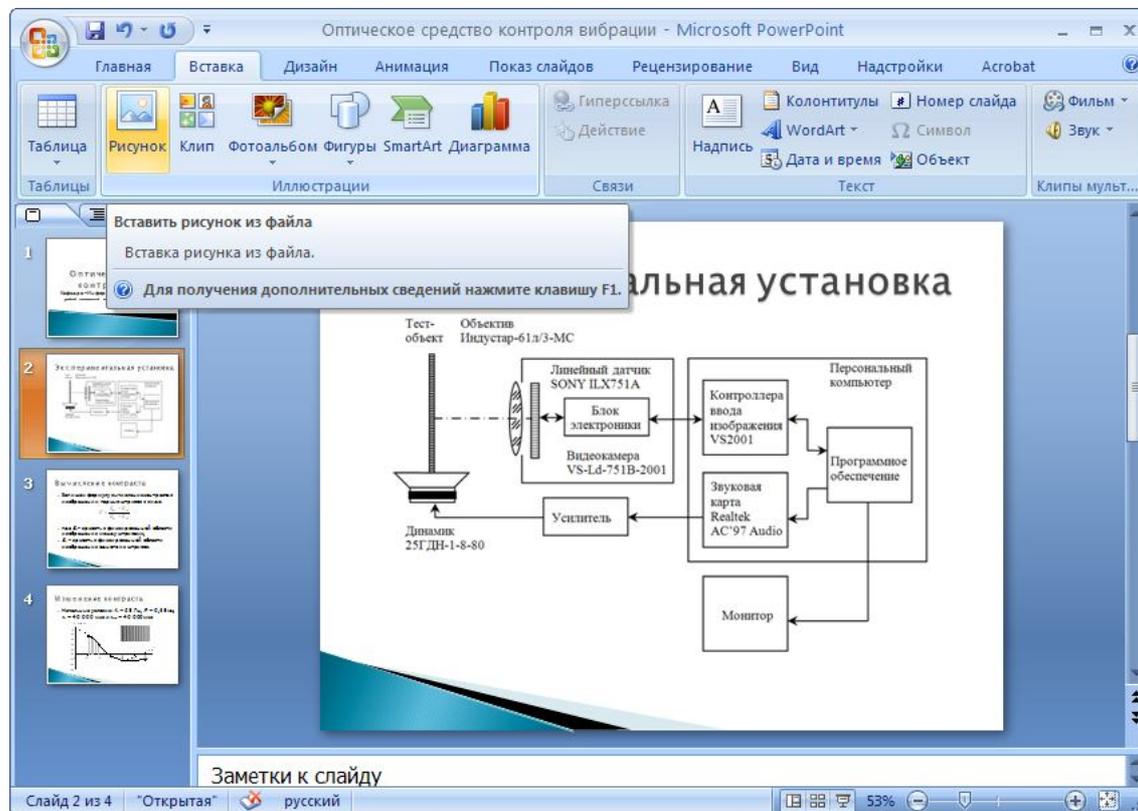


Рисунок 3.24 – Вставка рисунков в Microsoft Office PowerPoint 2007

В приложении Microsoft Office PowerPoint 2007 существует четыре группы возможных эффектов анимации: вход – эффекты при появлении объекта на экране; выделение – эффекты при работе пользователя с объектом; выход – эффекты исчезновения объекта со слайда; пути перемещения – эффекты перемещения объекта по слайду.

После того как выбран нужный эффект, под кнопкой «Добавить эффект» станут доступны несколько списков: начало – указывается действие, по которому будет применен эффект, например, по щелчку мыши или с предыдущим объектом; направление – задается направление применения эффекта, например, появление слева или сверху; скорость – определяется скорость исполнения эффекта.

Если на слайде анимационные эффекты применяются к нескольким объектам, то в режиме разработки рядом с ними будут отображены номера, показывающие очередь их появления. В любой момент разработанную анимацию можно посмотреть, нажав кнопку «Просмотр» панели «Настройка анимации».

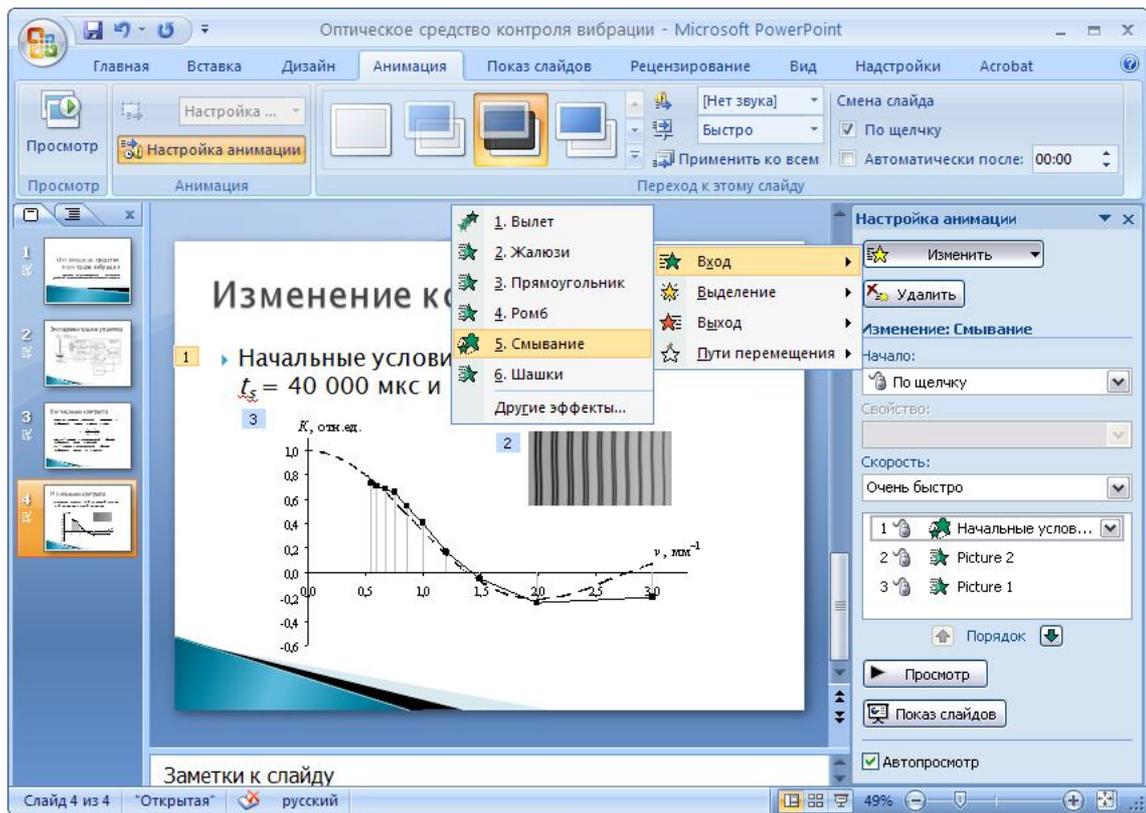


Рисунок 3.25 – Настройка анимации в Microsoft Office PowerPoint 2007

Заключительным этапом работы по созданию презентации является настройка режима ее показа. Для этого на ленте Microsoft Office выбирается закладка «Показ слайдов». С помощью элементов группы «Начать показ слайдов» можно запустить презентацию с начала, с текущего слайда или в произвольном порядке. Элементы группы «Мониторы» позволяют настраивать размер презентации под определенное разрешение монитора. Элементы группы «Настройка» предоставляют возможность записать звуковую дорожку речевого сопровождения, настроить время и установить параметры демонстрации электронной презентации.

3.6 Системы управления базами данных

На сегодняшний день базы данных – это одна из наиболее важных современных компьютерных технологий. Трудно представить программное обеспечение организации любого информационного процесса, которое не использует в своей работе базы данных. Рассмотрим основные термины, используемые в этой области информационных технологий.

База данных – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ.

Язык манипулирования данными – язык, предназначенный для формулирования запросов на поиск, обмен данными между прикладной программой и

базой данных. Стандартным языком манипулирования данными является язык SQL (Structure Query Language), применяемый в большинстве реляционных систем управления базами данных.

Система управления базами данных – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами.

В настоящее время система управления базами данных должна обеспечивать возможность представления внутренней структуры данных, физическую и логическую независимость данных, минимальную избыточность данных, возможность быстрого поиска, эффективные языки запросов к данным, поддерживать безопасность, надежность, конфиденциальность и целостность. Большинство современных систем управления базами данных используют реляционную модель данных. Рассмотрим ее более подробно.

3.6.1 Реляционная модель данных

Реляционная модель данных была предложена сотрудником фирмы IBM Эдгаром Коддом, изложившим основные идеи в статье «Реляционная модель для больших банков совместно используемых данных». Реляционная модель данных позволила снять недостатки существующих моделей, а наглядность и строгая теоретическая проработка принесли большую популярность модели среди разработчиков баз данных.

3.6.1.1 Отношения

Опишем основные понятия, относящиеся к реляционной модели. Реляционная модель основана на математическом понятии **отношения**, физическим представлением которого является таблица.

Реляционная модель предполагает организацию данных только в виде таблиц, каждая из которых состоит из строк и столбцов (рисунок 3.2).

	Столбец			
Таблица		Иванов	ИИТ-51	75
		Петров	ПКМ-42	62
		Сидоров	ПИЭ-71	44
Строка		Федоров	ИИТ-32	87
		Давыдов	МЭ-73	78
				Ячейка

Рисунок 3.26 – Пример табличной организации данных

В реляционной модели данных к таблицам предъявляются определенные требования, выраженные в принципе информационной неделимости: каждое значение в таблице реляционной модели может содержать только одну порцию данных; и принципе информационного кодирования: недопустимо, чтобы значение в таблице реляционной модели содержало более одной порции данных.

3.6.1.2 Ключи

Связи между строками таблиц реализуются в виде ключей. **Ключ** – это определенным образом помеченный столбец таблицы. Ключ может быть составным, то есть состоять из нескольких столбцов. Ключи обычно используют для исключения дубликатов, упорядочения столбцов, ускорения работы со строками таблиц и организации их связи.

Для того чтобы таблицы были связаны, необходимо установить между строками виртуальную связь с помощью первичных и внешних ключей.

Первичный ключ – это ключ, значения которого однозначно идентифицируют каждую из строк заданной таблицы.

Внешний ключ – это ключ, который соответствует первичному ключу другой таблицы.

Когда столбец помечается как ключ, дополнительно создается индекс. **Индекс** хранит список значений ключа и указатели на строки, содержащие эти значения. Наличие индекса позволяет системе управления базами данных быстро находить нужные строки. В целом индексы ускоряют выполнение операций чтения, но замедляют выполнение операций записи, так как при этом необходимо обновлять индекс.

3.6.1.3 Связывание таблиц

Связывание таблиц определяет их подчиненность. Существует три вида связывания таблиц: «один – к одному», «один – ко многим» и «много – ко многим». Рассмотрим более подробно каждый из этих видов.

Связывание таблиц вида «один – к одному» обычно используется при разбиении таблиц с большим числом столбцов на несколько частей. В этом случае в одной таблице остаются столбцы с наиболее важной и часто используемой информацией, а остальные столбцы переносятся в другие таблицы. Пример такого связывания приведен на рисунке 3.27, а, в котором одной строке первой таблицы соответствует одна строка второй таблицы.

Связывание таблиц вида «один – ко многим» означает, что одной строке одной таблицы может соответствовать несколько строк другой таблицы, в частном случае – ни одной. Такой вид связывания встречается наиболее часто. Пример связывания таблиц вида «один – ко многим» приведен на рисунке 3.27, б.

Связывание таблиц вида «много – ко многим» используется тогда, когда одной строке одной таблицы соответствует несколько строк другой таблицы и

наоборот. На практике связывание таблиц вида «много – ко многим» используется достаточно редко, и для реализации такого отношения в реляционной модели преобразуется в связывание несколько таблиц вида «один – ко многим». Пример такого связывания приведен на рисунке 3.27, в.

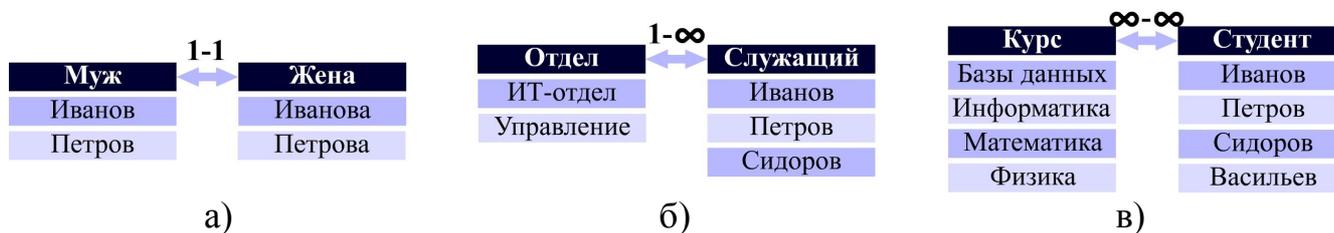


Рисунок 3.27 – Пример связывания таблиц вида:

а) «один – к одному»; б) «один – ко многим»; в) «много – ко многим»

При связывании таблиц необходимо следовать двум фундаментальным правилам целостности: первичный ключ не может быть пустым, и внешний ключ может быть либо пустым, либо равен значению первичного ключа, на который он ссылается. Только в этом случае можно добиться непротиворечивости представления данных.

3.6.2 Нормализация реляционных баз данных

После статьи, посвященной реляционной алгебре, Эдгар Кодд в 1972 г. опубликовал работу под названием «Дальнейшая нормализация реляционной модели баз данных», в которой ввел термин «нормализация» и сформулировал три нормальные формы. Рассмотрим их более подробно.

Нормализация – это метод организации реляционной базы данных с целью сокращения избыточности.

При разработке и проектировании любой реляционной базы данных одной из первых решается задача нормализации ее таблиц, причем этот процесс является итерационным и заключается в последовательном переводе таблиц из первой нормальной формы в нормальные формы более высокого порядка по определенным правилам. Каждая следующая нормальная форма ограничивает функциональную зависимость таблиц при сохранении свойств предшествующих нормальных форм.

Нормализация позволяет в полной мере реализовать преимущества реляционной модели, заставляет разработчика создавать больше отношений, равномерно распределяя в них информацию, что приводит к снижению избыточности и повышению целостности данных.

Первое правило реляционной модели состоит в том, что ни один столбец не должен содержать два или более значений.

Таблица представлена **в первой нормальной форме** тогда и только тогда, когда все ее столбцы являются простыми, то есть содержат только неделимые порции данных.

Приведение таблицы к первой нормальной форме упрощает разработку запросов и создание представлений, однако не исключает дублирования данных, что приводит к потере целостности и увеличению объема занимаемой памяти. Для решения этих проблем необходимо использовать вторую нормальную форму.

Для проведения дальнейшей нормализации таблиц необходимо использовать понятие первичного ключа, так как вторая нормальная форма требует, чтобы все столбцы зависели от первичного ключа.

Таблица представлена **во второй нормальной форме** тогда и только тогда, когда она представлена в первой нормальной форме, и каждый неключевой столбец полностью определяется первичным ключом. Избыточность первой нормальной формы может послужить причиной возникновения аномалий вставки и удаления. Аномалии вставки возникают тогда, когда данные невозможно добавить в таблицу до тех пор, пока они неполные. Применение второй нормальной формы позволяет устранить аномалии вставки и удаления.

Использование первой и второй нормальных форм позволяет избавиться от явного дублирования данных, однако неявное дублирование данных все же имеет место. Решить эту проблему можно с помощью третьей нормальной формы.

Таблица представлена **в третьей нормальной форме** тогда и только тогда, когда она представлена во второй нормальной форме, и все неключевые столбцы таблицы взаимно независимы и полностью зависят от значения первичного ключа.

Приведение отношений в третью нормальную форму является достаточным для решения практических задач, и на этом в большинстве случаев процесс проектирования реляционной базы данных заканчивается. Необходимо отметить, что существует четвертая и пятая нормальные формы, которые в большей степени представляют теоретический интерес, нежели практический.

3.6.3 Денормализация реляционных баз данных

При проектировании базы данных, удовлетворяющей всем нормальным формам, появляется переизбыток таблиц, так как переход к очередной нормальной форме ведет к разделению таблиц. При этом устраняется избыточность данных, однако появляется избыточность внутренних связей между таблицами. Кроме того, возникают дополнительные затраты на выполнение операций объединения, что значительно влияет на производительность.

Устранение избыточности не обязательно означает повышение производительности. Расходы на выполнение операций объединения значительны, поэтому разработчики иногда сознательно идут на нарушение правил нормализации. Процесс нарушения нормальных форм называется **денормализацией**.

3.6.4 Разработка баз данных в Microsoft Office Access 2007

При решении многих прикладных задач порой не требуется привлечения мощных средств разработки баз данных и группы профессиональных программистов. Например, если база данных должна хранить небольшой объем информации, интерфейс клиентских приложений должен быть интуитивно понятен и прост, то в качестве эффективной системы управления базами данных можно использовать программу **Microsoft Office Access 2007**, которая является лидером среди подобного класса программных продуктов. Изучим этапы проектирования баз данных и клиентских приложений в этой среде.

Рассмотрим в качестве примера базу данных «Студенты», которая состоит из трех таблиц: «Студенты», «Дисциплина» и «Результат».

Создать базу данных в Microsoft Office Access 2007 можно с помощью шаблонов, ориентированных на определенную предметную область. В большинстве случаев найти полностью удовлетворяющий целям проекта шаблон не удастся, поэтому приходится разрабатывать базу данных с самого начала. Для создания базы данных необходимо на этапе запуска Microsoft Office Access 2007 выбрать пункт «Новая база данных» и указать ее имя, при этом первая таблица создается автоматически.

Для создания следующих таблиц необходимо перейти на закладку «Создать» ленты Microsoft Office и выбрать пункт «Таблица» в группе «Таблица» (рисунок 3.28).

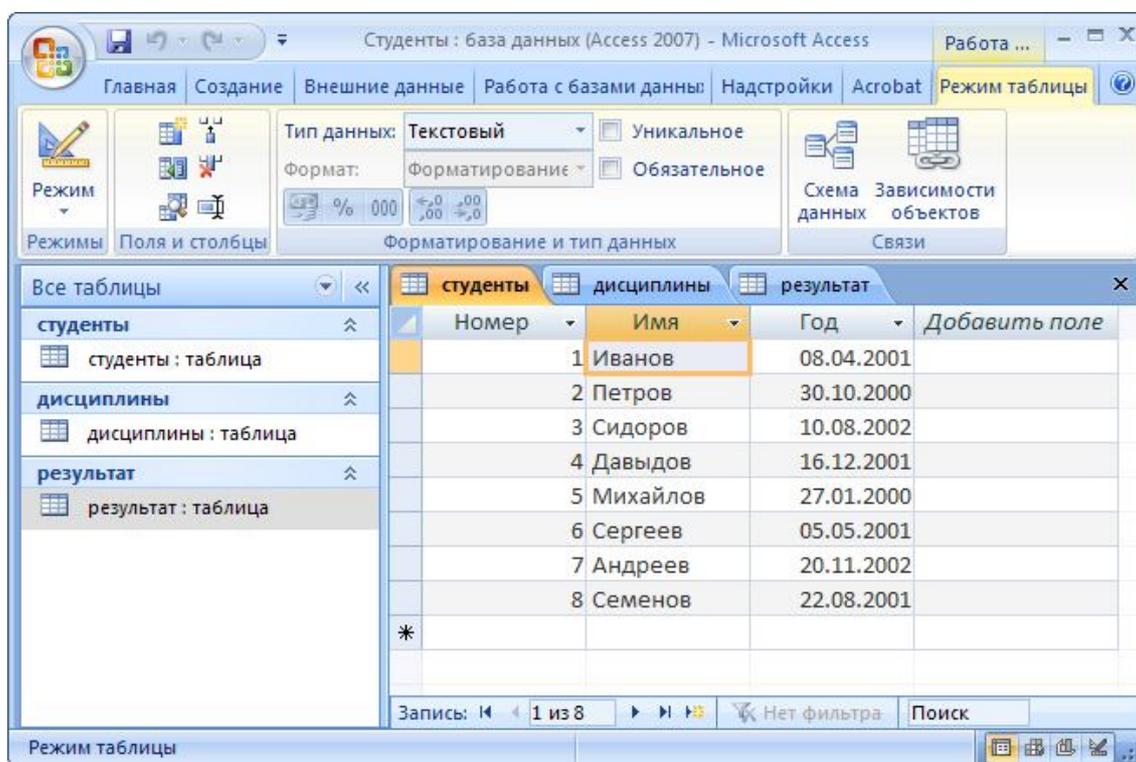


Рисунок 3.28 – Создание таблицы «Студенты»

Для каждой таблицы можно в режиме ввода данных или в режиме конструктора определить количество используемых столбцов, их тип, задать первичный ключ и индексы. В Microsoft Office Access 2007 определено восемь типов данных: текстовый, числовой, дата/время, денежный, логический, поле MEMO, гиперссылка, «вложение». Практически все они стандартные, и работа с этими типами данных не представляет трудностей. Наиболее интересным и новым является тип «вложение», который позволяет хранить внешние данные, например, фотографии, звуковые файлы, видеоданные.

Для каждого столбца помимо типа можно определить формат хранения информации, а также установить контроль ввода уникальных и ненулевых значений. Именно по такому принципу разработаны таблицы «Студенты» и «Дисциплины».

Таблица «Результат» содержит два внешних ключа для связи с разработанными таблицами. Microsoft Office Access 2007 предоставляет эффективный инструмент связи таблиц между собой. Для этого необходимо на ленте Microsoft Office на закладке «Режим таблицы» в группе «Поля и столбцы» выбрать элемент «Столбец подстановок». После этого нужно указать, значения из какого столбца и какой таблицы будут подставляться в исходную таблицу. Такую операцию необходимо проделать для создания всех столбцов, содержащих внешние ключи.

Для того чтобы посмотреть и отредактировать схему базы данных, нужно на ленте Microsoft Office на закладке «Режим таблицы» в группе «Связи» выбрать элемент «Схема данных», после чего представленную схему данных можно отредактировать, настроить связи между таблицами, убрать или добавить представления.

Все строки таблиц можно отсортировать по возрастанию или убыванию либо отфильтровать, причем сложность фильтра может быть различной. Для этого необходимо на ленте Microsoft Office на закладке «Главная» выбрать группу «Сортировка и фильтр». Более профессиональным способом фильтрации и сортировки осуществляются с помощью запросов, которые будут рассмотрены ниже.

Мощным средством визуализации информации, хранящейся в базе данных Microsoft Office Access 2007, являются формы. Формы представляют собой интерфейсный элемент, который позволяет осуществлять взаимодействие между пользователем и базой данных. В формах пользователи осуществляют навигацию, редактирование и удаление данных (рисунок 3.29).

Рассмотрим способы создания форм в Microsoft Office Access 2007. Для этого на ленте Microsoft Office на закладке «Создать» в группе «Форма» необходимо выбрать пункт «Пустая форма». После этого нужно из списка полей переместить столбцы, которые будут отражены на данной форме. Данный способ создания форм осуществляется в режиме формы. На форме можно просматривать и редактировать каждую строку, переходить от одной строки к другой, редактировать, осуществлять поиск и фильтрацию информации.

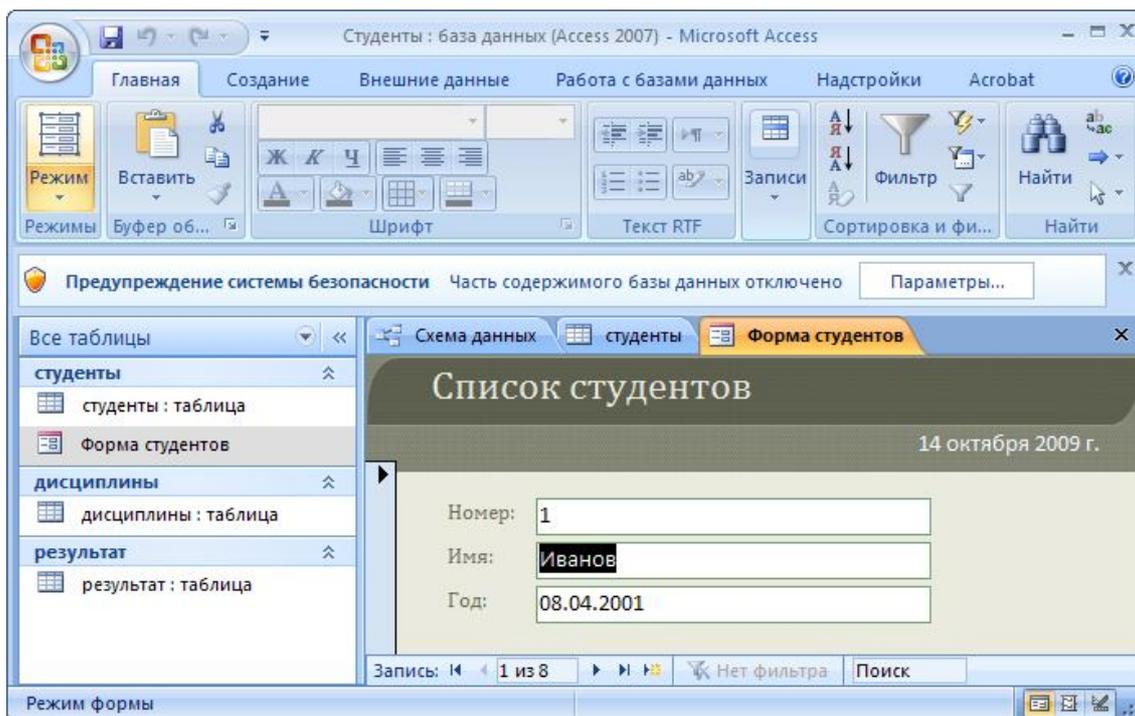


Рисунок 3.29 – Форма «Список студентов»

При необходимости можно использовать режим конструктора, который позволяет более детально проработать форму и настроить все элементы управления, или мастер создания форм, который предоставляет возможность быстрой разработки макета формы.

В некоторых случаях, когда столбцов в таблице большое количество, бывает удобно осуществлять навигацию по таблице, где представлены только основные столбцы, а полную информацию о текущей строке выводить на форму. Такую возможность предоставляет Microsoft Office Access 2007 путем создания разделенных форм. Разработаем разделенную форму для таблицы «Результат». Для этого на ленте Microsoft Office на закладке «Создать» в группе «Форма» необходимо выбрать элемент «Разделенная форма».

Разработка форм и сложных запросов к базе данных направлена лишь на то, чтобы в удобном виде представить информацию пользователю, однако на сегодняшний день ее необходимо еще и сохранить, причем оформление информации также значимо, как и сама информация. Для этих целей в Microsoft Office Access 2007 используются отчеты, которые обеспечивают вывод информации, хранящейся в базе данных, на экран монитора, принтер или внешний носитель информации. Причем существующие мощные средства разработки отчетов позволяют создавать дизайн практически любой сложности.

Приложение Microsoft Office Access 2007 предоставляет возможность создавать макет отчета для выбранной таблицы, использовать мастер отчетов либо разрабатывать отчет с самого начала. Последний способ для опытных разработчиков более приемлем, так как позволяет избежать избыточности и сложных конструкций при проектировании отчета.

Для создания отчета необходимо на ленте Microsoft Office на закладке «Создать» в группе «Отчеты» выбрать элемент «Пустой отчет». После этого на панели «Список полей» нужно выбрать соответствующие столбцы таблиц. Далее необходимо сгруппировать данные для получения среза информации, для этого нужно на ленте Microsoft Office на закладке «Формат» в группе «Группировка и итоги» выбрать элемент «Группировка», после чего выбрать столбец, по которому осуществляется группировка, и столбец, по которому осуществляется сортировка. В нашем случае сгруппируем данные в таблице «Результат» по дисциплинам и отсортируем данные по оценкам.

В каждой группе подведем итоги: выведем средний результат по дисциплине, для этого в свойствах группировки (нижняя панель «Группировка, сортировка и итоги») необходимо выбрать нужную группировку, далее – элемент «С итогами», установить итог на поле «Оценка», тип итогов – «Среднее», причем вывод осуществлять в примечании группы. Для пояснения итога группировки можно в контекстном меню для ячейки с результатом группировки выбрать пункт «Задать подпись». После этого можно вставить в отчет заголовок, служебную информацию, выбрать стиль и отформатировать данные (рисунок 3.30).

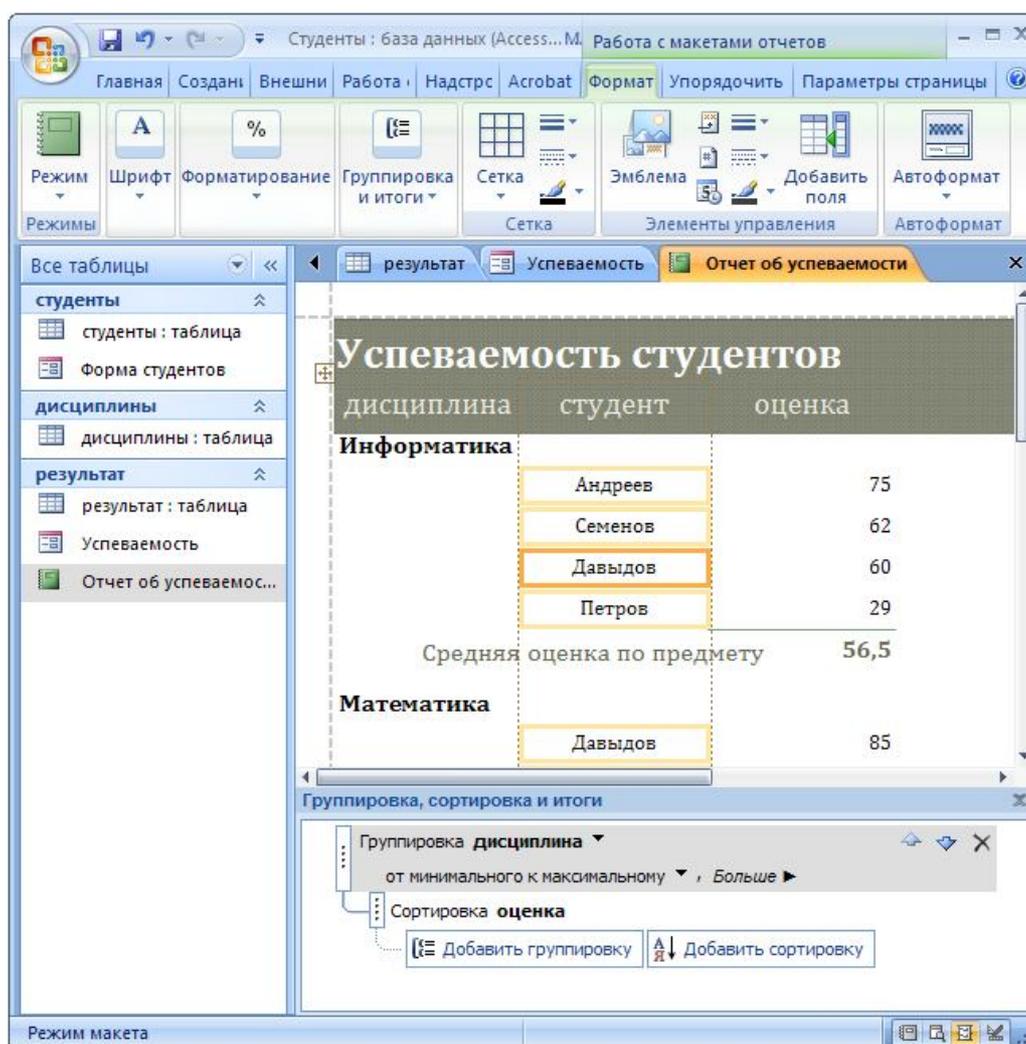


Рисунок 3.30 – Отчет «Успеваемость студентов»

Запросы помогают сформировать выборки данных для представления их пользователю. В Microsoft Office Access 2007 существует отдельный инструмент, который реализует структурированный язык запросов SQL. Создавать запросы можно также несколькими способами: с помощью мастера запросов и с помощью конструктора. Разработаем запрос с помощью конструктора с самого начала.

Для этого необходимо на ленте Microsoft Office на закладке «Создать» в группе «Другие» выбрать элемент «Конструктор запросов». Автоматически система предложит выбрать таблицы, из которых будут выбраны данные. Если в схеме данных до этого были установлены связи между таблицами, то они сохраняются, при этом их можно отредактировать и настроить под решаемую задачу. После этого необходимо выбрать столбцы из таблиц, которые будут включены в выходной запрос. Далее по необходимости устанавливается порядок сортировки. В нашем примере установлена сортировка выходных данных по столбцу «Оценки» по убыванию (рисунок 3.31).

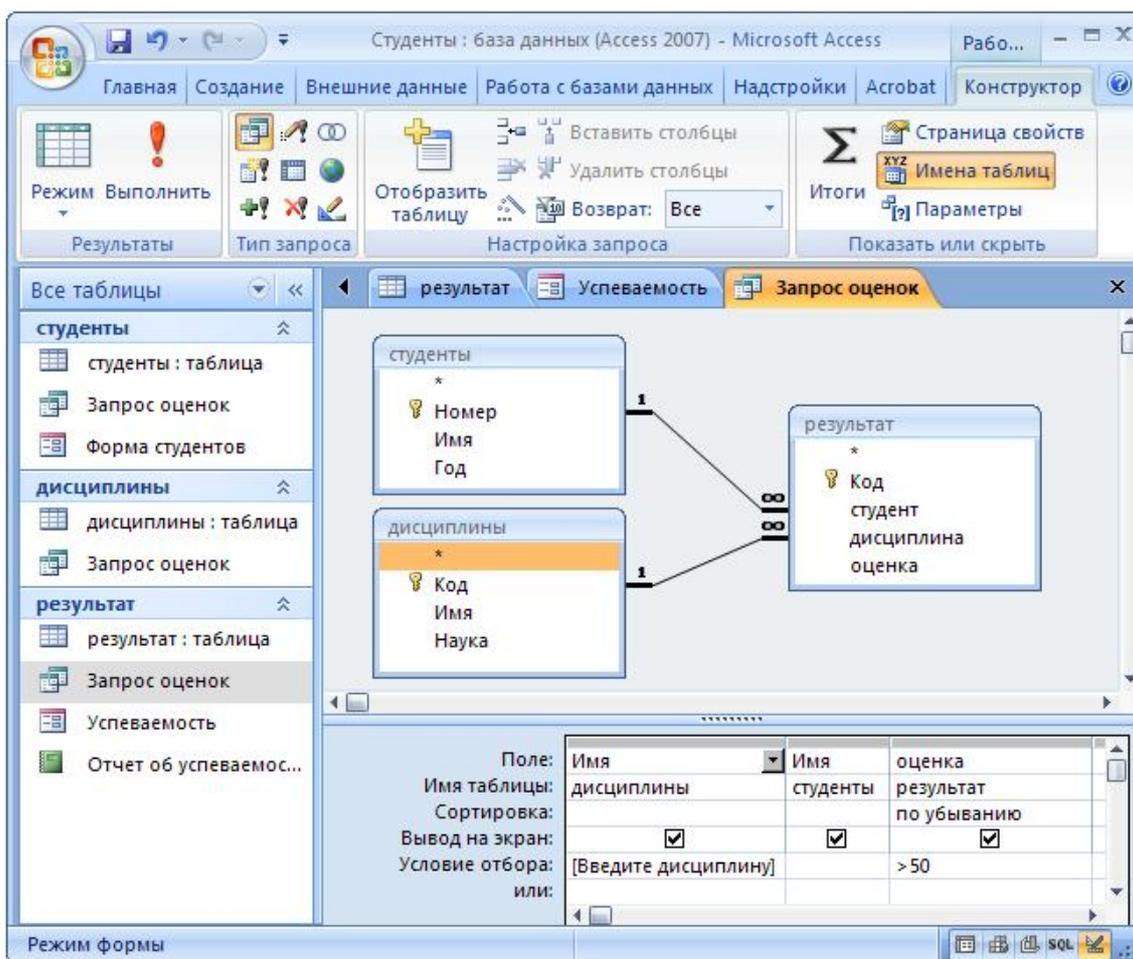


Рисунок 3.31 – Конструирование запроса «Оценки»

Затем определяется условие отбора данных с помощью булевых функций. Интересным представляется случай ввода параметра сравнения, при этом вместо условия устанавливается текст приглашения, заключенный в квадратные

скобки. При запуске запроса первоначально необходимо ввести параметр, при совпадении значения в заданном столбце таблицы с введенным параметром строка включается в результат запроса.

Автоматически после создания запроса формируется SQL-инструкция, по мере необходимости ее можно исправлять или дорабатывать.

Для графического представления данных в Microsoft Office Access 2007 используются диаграммы. Именно использование диаграмм является дополнительным преимуществом Microsoft Office Access 2007, так как графическая интерпретация данных повышает наглядность и эффективность их восприятия.

Диаграммы можно использовать в формах, таблицах, запросах. Рассмотрим пример создания диаграмм на основе разработанного запроса «Оценки».

Для вставки в запрос диаграммы необходимо на ленте Microsoft Office на закладке «Главная» в группе «Режим» выбрать элемент «Сводная диаграмма».

После этого нужно определить тип диаграммы, значения и категории, для чего из «Списка полей диаграммы» переместить необходимые столбцы в соответствующие области. Далее подписать названия осей, отформатировать с помощью стандартных средств текстовую и числовую информацию. Когда отчет создан, можно, выбрав режим «Сводная диаграмма», просмотреть графическую интерпретацию результата запроса «Оценки».

3.7 Технологии обработки графической информации

Проблему обработки графической информации решает раздел информатики, называемый **компьютерная графика**. Компьютерная графика занимается методами создания и редактирования изображений с помощью компьютера. Компьютерную графику используют в своей деятельности люди многих профессий, особенно инженеры, работающие с техническими оптическими системами. Выделяют несколько направлений использования компьютерной графики: изобразительная компьютерная графика, обработка и анализ изображений, распознавание образов и когнитивная компьютерная графика (рисунок 3.32).



Рисунок 3.32 – Направления компьютерной графики

Изобразительная компьютерная графика нашла широкое применение при создании видеофильмов, анимационных роликов, презентаций и компьютерных игр. Обработка и анализ изображений применяются при повышении качества фотографий, выделении на изображении характерных областей. Распознавание образов используется для построения систем технического зрения, в процессе

анализа рентгеновских снимков и расшифровки результатов космической съемки. Когнитивная компьютерная графика нацелена на создание моделей представления данных с помощью изображений, которые способствуют познанию.

Компьютер способен обрабатывать данные, представленные только в цифровом виде. Описание графических данных выполняется в соответствии с определенными правилами, совокупность которых часто называется **форматом**. Принято выделять два основных формата графических данных: растровый и векторный.

3.7.1 Растровая графика

Основным элементом растровой графики является **пиксель**, представляющий собой неделимый элемент, параметры которого описывают соответствующий ему участок реального или синтезированного изображения (рисунок 3.33). К основным форматам хранения растровых изображений относятся BMP, GIF, TIF, JPG, PNG.

Растровая графика имеет два значимых достоинства. Во-первых, каждому пикселю можно задать практически любой оттенок, поэтому при уменьшении размера пикселя растровая графика эффективно представляет изображения фотографического качества. Во-вторых, многие компьютерные устройства ввода и вывода данных используют точки для представления отдельных пикселей, поэтому растровые изображения можно без дополнительных преобразований принимать от этих устройств или отправлять на них графические данные.

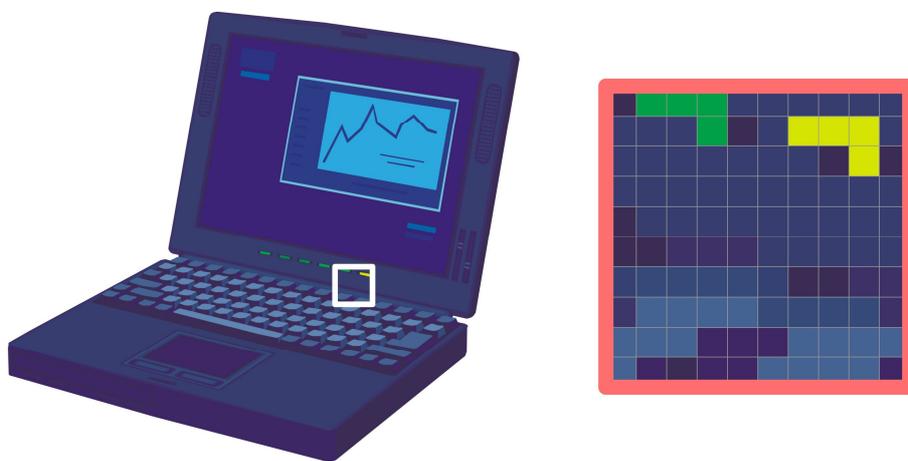


Рисунок 3.33 – Пример растрового изображения

Однако растровые изображения обладают существенными недостатками. Во-первых, при трансформации растровых изображений, например, повороте и масштабировании, может потеряться часть информации. Во-вторых, для хранения растровых изображений требуется большой объем дискового пространства, так для каждого пикселя необходимо запоминать его цвет, чем больше цвето-

вых оттенков способно воспроизводить изображение и чем больше его размер, тем больше под него нужно отводить памяти.

Для решения этой проблемы применяется **сжатие** графических данных, заключающееся в изменении размера графического файла в зависимости от способа организации в нем данных. Все алгоритмы сжатия делятся на две категории: приводят к потере качества, не приводят к потере качества изображений. Существует несколько подходов к сжатию растровых изображений, основными из которых являются метод сжатия **RLE** (Run Length Encoding – кодирование с переменной длиной строки), метод сжатия **LZW** (Lempel, Ziv, Welch – первые буквы фамилий разработчиков) и метод сжатия **JPEG** (Joint Photographic Experts Group – объединенная группа экспертов по фотографии).

При использовании метода сжатия **RLE** осуществляется поиск одинаковых пикселей в одной строке, называемых группами. Группа заменяется на два значения: повторяющийся пиксель и количество его повторений. Этот метод имеет высокие показатели сжатия, если исходное изображение содержит большое количество однотонных областей.

При использовании метода сжатия **LZW** происходит поиск во всем графическом файле и сохранение в словаре одинаковых последовательностей данных, которые называют фразами. Каждой уникальной фразе присваивается более короткий маркер.

Сжатие графических данных с помощью метода **JPEG** основано на использовании алгоритма дискретных косинусных преобразований и кодирования энтропии. Особенность данного алгоритма заключается в том, что он частично идентифицирует и удаляет данные, следовательно, достигается высокий уровень уплотнения записи изображения без заметных потерь качества. Суть этого метода заключается в определении разницы между пикселями с целью поиска плавных цветовых переходов в области 9×9 пикселей. Вместо действительных значений цвета сохраняется скорость его изменения от пикселя к пикселю.

3.7.2 Векторная графика

Основным элементом векторной графики является **примитив**, представляющий собой неделимый элемент, параметры которого описывают с помощью математической модели, задающей их форму, положение, габаритные размеры и оформительские атрибуты: цвет, тип заливки, толщину граничного контура. Объекты сложной формы образуются из комбинации базовых примитивов (рисунки 3.34). К основным форматам хранения векторных изображений относятся WMF, EPS, CDR, AI, FLA.

Векторная графика имеет также два значимых достоинства. Во-первых, векторная графика позволяет задавать достаточно сложные изображения с высокой точностью, причем занимают они небольшое дисковое пространство по сравнению с растровыми изображениями. Во-вторых, объекты, заданные в векторной форме, легко масштабируются, а точность их отображения определяется только параметрами устройства вывода.



Рисунок 3.34 – Пример векторного изображения

К недостаткам векторных изображений относятся ограниченные возможности создания изображений фотографического качества, так как представить распределение цветов фотографии с помощью векторных примитивов сложно. Также векторные изображения, выведенные на печать, могут значительно отличаться от оригинала, так как в принтерах векторный примитив переводится в растровое изображение с потерей качества.

3.7.3 Цветовые модели

Понятия цвета в компьютерной графике является основополагающими. Под **цветом** понимают качественную субъективную характеристику электромагнитного излучения оптического диапазона, определяемую на основании возникающего физиологического зрительного ощущения и зависящую от ряда физических, физиологических и психологических факторов.

Описания цвета в терминах выбранного цветового пространства называют **цветовой моделью**. С помощью модели каждый цветовой оттенок можно представить в виде совокупности чисел – цветовых компонентов, что дает возможность обмениваться цветовой информацией между компьютером и устройствами ввода и вывода графических данных.



Рисунок 3.35 – Классификация цветовых моделей

Все цветовые модели можно разделить на три группы: аппаратно-зависимые модели, используемые в технических средствах ввода и вывода графической информации; аппаратно-независимые модели, описывающие цвет в абстрактных колориметрических терминах; интуитивные модели, построенные на основе субъективного восприятия цвета человеком (рисунок 3.35).

В 1860 г. Максвелл ввел цветовую модель **RGB** (Red – красный, Green – зеленый, Blue – синий), которая является аддитивной моделью, так как требуемый цветовой оттенок формируется за счет смешения трех первичных цветов: красного, зеленого и синего. Эта цветовая модель нашла широкое применение в устройствах самосветящегося типа, таких, как мониторы и телевизоры.

В 1951 г. Э. Мюллер предложил цветовую модель **СМУК** (Cyan – голубой, Magenta – пурпурный, Yellow – желтый, Black – черный), которая является субтрактивной моделью, так как нужный оттенок получается в результате отражения светового потока внешнего источника и равен сумме непоглощенных составляющих. Эта цветовая модель используется в устройствах несамосветящегося типа, таких, как принтеры, светофильтры, бумажные оттиски.

В 1931 г. Международная комиссия по освещению CIE разработала цветовую систему **XYZ**, называемую также «нормальная цветовая система», которая основана на восприятии цвета некоторым стандартным наблюдателем. Эта цветовая модель получена экспериментально, когда множество пользователей с помощью трех монохроматических источников света X, Y и Z пытались воспроизвести эталонный заданный цвет. По результатам проведенных исследований были получены усредненные характеристики X, Y и Z реакции человеческого глаза на свет определенной длины волны. По значениям X, Y и Z компонентов можно получить весь спектр видимых глазом цветов.

В 1905 г. художник А. Мансел дал интуитивное описание трехмерного цветового пространства, с помощью которого можно графически представлять положение видимого цвета в цветовом пространстве. Наиболее используемой интуитивной цветовой моделью является модель **HSB** (Hue – оттенок, Saturation – насыщенность, Brightness – яркость). Оттенок описывает определенный цвет, яркость – интенсивность цвета, насыщенность – степень чистоты цвета. Цветовая модель HSB является субъективной и абстрактной, так как нет устройств, которые синтезируют цвет в этой системе.

3.7.4 Обработка графической информации с помощью Adobe CS4

Пакет приложений Adobe Creative Suite 4 представляет собой набор дизайнерских программ для профессиональной работы с текстом, графикой, анимацией и видеoinформацией, созданный компанией Adobe (www.adobe.com). Наиболее востребованными программами этого пакета являются Adobe Photoshop CS4 для работы с растровой графикой, Adobe Illustrator CS4 для работы с векторной графикой, Adobe Flash professional CS4 для создания анимации. У всех этих программ единый интерфейс и дизайн. Дадим краткую характеристику каждой из этих программ.

3.7.4.1 Программа обработки растровой графики Adobe Photoshop CS4

Программа Adobe Photoshop CS4 является признанным лидером в классе программных продуктов по обработке растровой графики (рисунок 3.36), так как содержит все необходимое для качественного создания и обработки растровых изображений: коррекция и ретушь фотографий, применение фильтров к изображениям, работа со слоями. Представленное программное обеспечение активно применяется графическими дизайнерами в своей работе. Опишем ключевые достоинства программы Adobe Photoshop CS4.

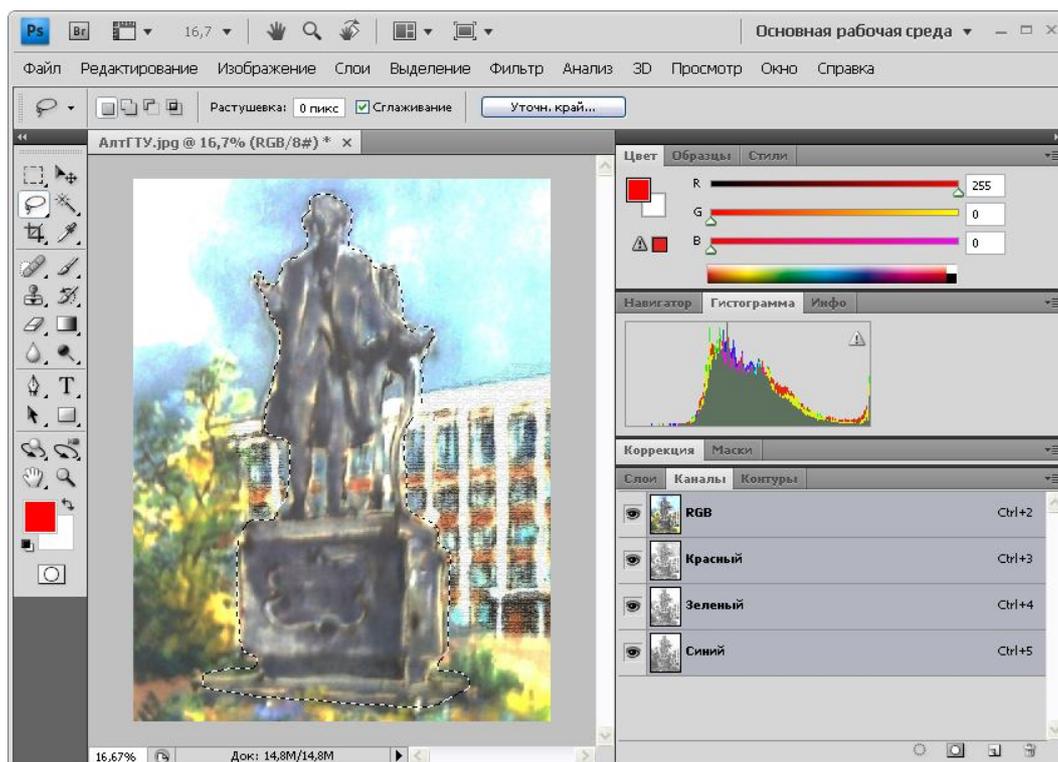


Рисунок 3.36 – Интерфейс программы Adobe Photoshop CS4

Одним из главных преимуществ Adobe Photoshop CS4 является наличие в программе удобного доступа ко всем необходимым инструментам коррекции цвета и тона изображений, а также к средствам управления изображением в целом. Некоторые инструменты осуществляют интеллектуальную обработку цветов и оттенков в изображении. В программе Adobe Photoshop CS4 реализована технология быстрого создания и редактирования растровых и векторных масок, настройки их плотности и растушевки, выделения несмежных объектов.

Существует возможность создания единого изображения из серии различных снимков, масштабирования с учетом содержания изображения, рисования на трехмерных моделях и их обтекания двухмерными картинками.

3.7.4.2 Программа обработки векторной графики Adobe Illustrator CS4

Программа Adobe Illustrator CS4 представляет собой один из самых распространенных векторных редакторов, который содержит огромные возможности для создания и редактирования векторных примитивов (рисунок 3.37).

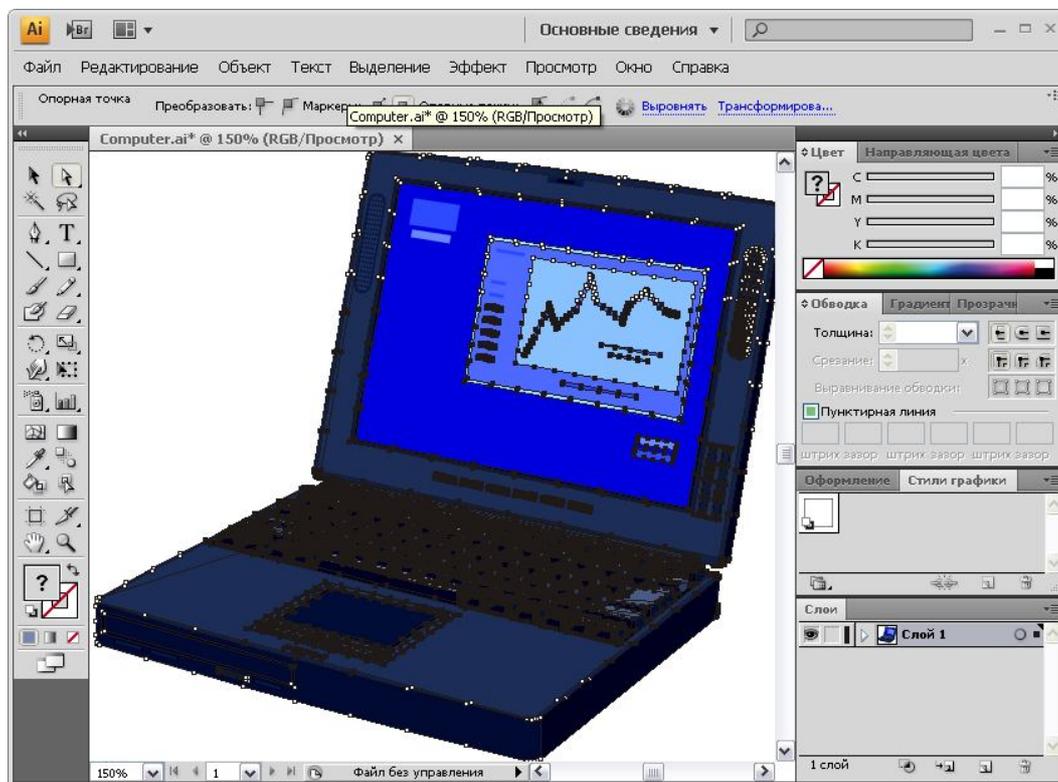


Рисунок 3.37 – Интерфейс программы Adobe Illustrator CS4

Программа Adobe Illustrator CS4 позволяет создавать перспективу для точных фигур и сцен, работать с градиентами непосредственно на объектах, осуществлять контроль над обводками переменной ширины, указателями стрелок и пунктирами, проектировать дизайн с помощью точных инструментов и расширенных элементов управления контурами. В данной программе поддерживается работа с большим количеством монтажных областей разных размеров в одном и том же графическом файле.

3.7.4.3 Программа разработки анимации Adobe Flash Professional CS4

Этот программный продукт Adobe Flash Professional CS4 является одним из наиболее популярных анимационных редакторов. Интерфейс данной программы приведен на рисунке 3.38.

К особенностям программы Adobe Flash Professional CS4, кроме стандартных инструментов создания векторных примитивов, можно отнести поддержку работы со слоями, возможность разработки классической анимации

объектов и анимации их формы, использование встроенного языка программирования Actionscript, с помощью которого можно создавать объектно-ориентированную анимацию, что значительно упрощает процесс проектировки приложения.

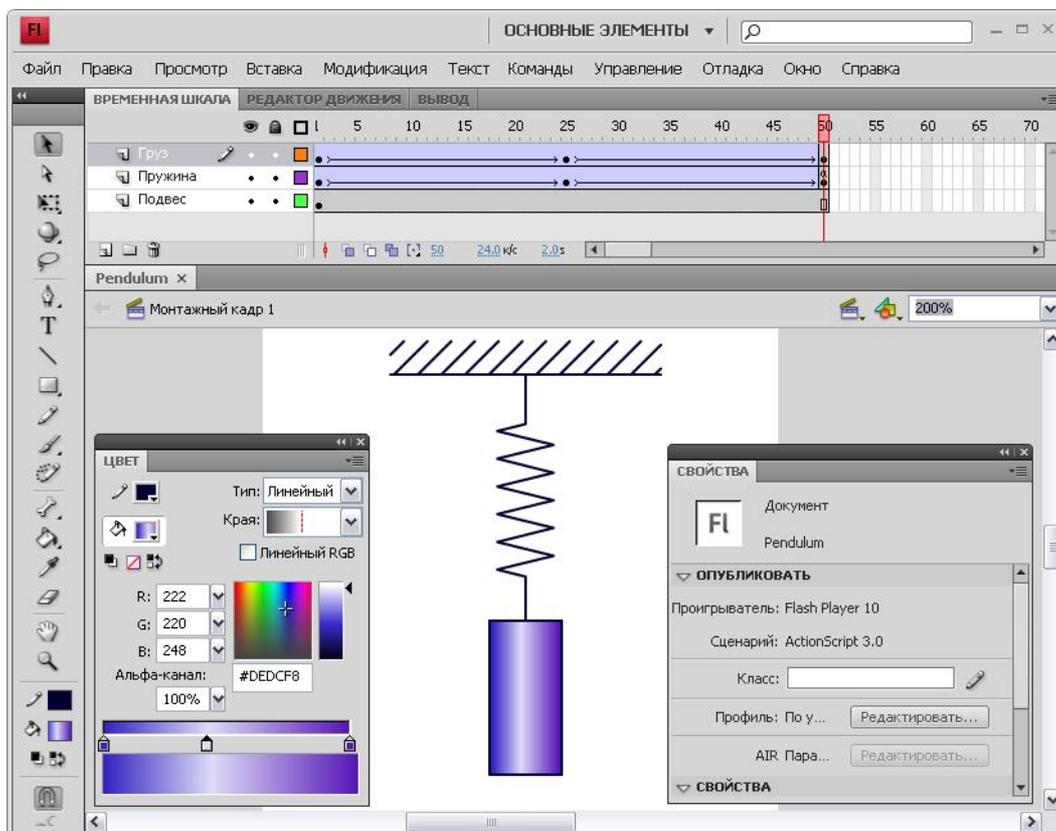


Рисунок 3.38 – Интерфейс программы Adobe Flash Professional CS4

Также в данном программном продукте можно использовать наборы стандартных настроек анимации движения, создавать цепные эффекты из ряда связанных объектов, выполнять анимацию плоских объектов в трехмерном пространстве, внедрять в анимационный проект видеофильмы и звуковые файлы, экспортировать разработанные анимационные ролики в различные форматы. Опубликованные анимационные файлы эффективно используют доступное аппаратное обеспечение для воспроизведения.

Тезаурус

Программное обеспечение, конфигурация, служебное программное обеспечение, операционная система, утилита, драйвер, прикладное программное обеспечение, многозадачность, мультипроцессорная обработка, поток, прерывание, файловая система, каталог, файл, тип файла, кластер, форматирование, стиль, формула, презентация, анимация, база данных, система управления базами данных, реляционная модель данных, первичный ключ, индекс, нормализация, компьютерная графика, растровая графика, векторная графика, пиксель, сжатие, примитив, цвет.

Контрольные вопросы

1. В чем заключается отличительная особенность служебного уровня программного обеспечения?
2. Перечислите основные функции, которые выполняют современные операционные системы.
3. Дайте определение понятию файл.
4. Как организовано хранение информации о файлах и каталогах в файловой системе FAT?
5. В чем заключаются преимущества операционной системы Linux перед другими операционными системами?
6. Какие этапы включает в себя процесс форматирования текста в программе Microsoft Office Word 2007?
7. Каково отличие относительного адреса ячейки от абсолютного адреса в программе Microsoft Office Excel 2007?
8. Как в программе Microsoft Office PowerPoint 2007 создать анимацию объекта?
9. Для чего в реляционной базе данных применяют индексы?
10. Назовите состав, назначение и возможности пакета программ Adobe Creative Suite 4.

Список рекомендуемой литературы

1. Microsoft Office 2007. Все программы пакета : самоучитель [Текст] / А. Н. Тихомиров, А. К. Прокди, П. В. Колосков, И. А. Клеандрова. – СПб . : Наука и Техника, 2008. – 608 с.
2. Божко, А. Н. Компьютерная графика [Текст] / А. Н. Божко, Д. М. Жук, В. Б. Маничев. – М . : Изд-во МГТУ, 2007. – 392 с.
3. Зрюмов, Е. А. Базы данных для инженеров [Текст] / Е. А. Зрюмов, А. Г. Зрюмова. – Барнаул : Изд-во АлтГТУ, 2010. – 132 с.
4. Информатика : базовый курс [Текст] / под ред. С. В. Симоновича. – СПб . : Питер, 2009. – 640 с.
5. Соболев, Б. В. Информатика [Текст] / Б. В. Соболев, А. Б. Галин, Ю. В. Панов. – Ростов н/Д : Феникс, 2007. – 446 с.
6. Таненбаум, Э. Современные операционные системы [Текст] / Э. Таненбаум. – СПб . : Питер, 2010. – 1120 с.
7. Столлингс, В. Операционные системы [Текст] / В. Столлингс. – М . : Вильямс, 2004. – 848 с.

ГЛАВА 4

МОДЕЛИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ И ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

В настоящей главе рассматриваются общие вопросы, связанные с классификацией и формами представления моделей, методами и технологией моделирования, а также описываются примеры математического моделирования оптических задач с помощью программы Mathcad.

4.1 Моделирование как метод познания

В профессиональной деятельности любой инженер постоянно сталкивается с различными задачами. Например, оптотехник занимается проектированием дифракционных решеток, вычислением параметров линзовых систем, разработкой оптических методов измерения физических величин, исследованием свойств фоточувствительных материалов. Все эти задачи можно разделить на два класса: вычислительные и функциональные.

Вычислительные задачи служат для точного расчета каких-либо величин, например, при проектировании репродукционной оптической системы рассчитывается ее линейное увеличение, длина, апертура, размер входного зрачка и его расположение.

Функциональные задачи направлены на создание некоторых аппаратов или систем, предназначенных для выполнения определенных действий. Примером функциональной задачи является создание высокоскоростной видеокамеры, способной регистрировать быстропротекающие процессы с частотой 400 000 кадров в секунду или проектирование тепловизора, позволяющего производить измерения в спектральном диапазоне от 8 до 14 мкм.

Решение любой задачи представляет собой последовательный процесс перехода от реального объекта, на базе которого ставится задача, до ее конечного решения. Одним из этапов данного процесса и является создание модели исследуемого объекта.

Существуют четыре основных вида задач моделирования. Для первого вида задач с практическим содержанием известны математические формулы и физические законы, например, моделирование изображения интерференционной картины от двух источников освещения. Задачи второго вида не имеют точного метода решения или он достаточно громоздок, например, моделирование оптических систем с использованием трассировки лучей. Задачи третьего вида включают в себя актуализацию знаний, например, переход от микроразмерной к наноразмерной фотолитографии при проектировании оптических систем. Для задач четвертого вида характерно информационное моделирование, включающее усложнение алгоритмов функционирования модели, расширение модели, например, повышение точности численных методов решения уравнений Максвелла.

Как правило, при решении задачи от реального объекта исследования переходят к его модели. Это, в первую очередь, связано с тем, что невозможно учесть и рассмотреть все свойства и характеристики исследуемого объекта. Во-вторых, используя модель, а не оригинал, легче исследовать его характеристики. Поэтому при создании модели выделяют наиболее важные свойства объекта, что приводит к упрощению процесса или явления, а значит и всей решаемой задачи.

При использовании модели происходит переход от реального объекта к искусственно созданному объекту, представленному в виде схемы, математических формул, физической конструкции, наборов данных и алгоритмов их обработки. **Исследуемый объект**, по отношению к которому изготавливается модель, называется **оригиналом, образцом, прототипом**.

Таким образом, **модель** (от лат. *modulus* – мера, образец, норма) – материальный или мысленно представляемый объект, который в процессе использования замещает объект-оригинал, сохраняя важные для данного исследования характерные свойства этого оригинала, которые можно в определенных целях использовать для получения информации об исходном объекте.

Процесс исследования объекта с целью создания его модели называется **моделированием**. Исследование объекта включает в себя выделение основных характеристик объекта-оригинала, его строения и свойств. В процессе исследования свойства объекта могут уточняться и дополняться.

Тем самым моделирование может быть определено как метод познания, включающий создание модели (искусственного заменителя) явления, предмета, объекта, ситуации (объекта моделирования) и проведение экспериментов с этой моделью для последующего применения полученных результатов к объекту моделирования. Теория замещения объектов-оригиналов объектом-моделью называется **теорией моделирования**. Если построенная модель дает удовлетворительные результаты, то говорят, что модель **адекватна** рассматриваемому объекту, процессу или явлению.

Современное моделирование трудоемкий процесс, требующий знаний не только в области исследуемого объекта, но и в аналитических компьютерных системах, применяемых для моделирования.

Процесс моделирования будет оправдан только в тех случаях, когда исследуемые события растянуты во времени, и нет смысла дожидаться их наступления; создание объекта очень дорого, исследование объекта приводит к его разрушению.

Теория моделирования содержит много способов создания моделей, все их можно разделить на две группы: аналитическое и имитационное моделирование. Аналитическое моделирование основано на построении моделей в виде аналитических выражений – формул (системы линейных, нелинейных и дифференциальных уравнений). К полученным моделям применяют аналитические или приближенные численные методы.

Имитационное моделирование предполагает построение модели с характеристиками, адекватными оригиналу, на основе его свойств. Модель строится

таким образом, что любое внешнее воздействие, вызывающее изменение свойств модели, будет вызывать такие же изменения у оригинала.

С помощью процесса моделирования можно (рисунок 4.1) вырабатывать понимание устройства и поведения объекта, прогнозировать характеристики и состояние объекта, выбирать воздействия, которые позволят достичь поставленных целей.



Рисунок 4.1 – Задачи моделирования объекта

В реальности эти задачи могут быть связаны и взаимно дополнять друг друга. В зависимости от характера решаемой задачи и свойств моделируемого объекта выбираются свойства модели и используемый математический аппарат.

4.2 Классификация и формы представления моделей

Внешний мир является постоянным источником объектов и различных событий, происходящих с этими объектами. Учитывая различный характер как самих объектов исследования, так и разные по природе воздействия на них, можно выделить несколько основных классификаций способов моделирования, а соответственно, и форм представления моделей. Модели классифицируют по следующим признакам: по области использования, по наличию воздействий, по отношению ко времени, по возможности реализации, по сфере применения (рисунок 4.2).

По области использования можно выделить пять моделей. Учебные модели используются при обучении, к ним относятся тренажеры, наглядные пособия, обучающие программы. Опытные модели представляют собой уменьшенные или увеличенные копии проектируемого объекта, они используются для исследования и прогнозирования его будущих характеристик. Игровые модели создаются для наблюдения за поведением объекта в различных условиях. Имитационные модели имитируют реальность с определенной степенью точности. В случае имитационного моделирования эксперимент либо многократно повторяется, чтобы изучить и оценить последствия каких-либо действий на реальную обстановку, либо проводится одновременно со многими другими похожими объектами, но поставленными в разные условия. Подобный метод поиска правильного решения называется методом проб и ошибок. Научно-технические модели разрабатываются для исследования различных процессов и явлений.



Рисунок 4.2 – Классификация моделей

По наличию воздействий на объект можно использовать две модели. Детерминированные модели позволяют получить однозначно определенный результат, например, какова будет интенсивность светового потока после прохождения через среду. Стохастические модели позволяют предсказать только вероятность каждого возможного результата.

По отношению ко времени имеются два основных класса моделей. Статические модели описывают состояние системы или объекта в определенный момент времени, например, яркость пикселя цифрового изображения объекта. Динамические модели отражают изменение системы или объекта во времени, например, изменение электрического сигнала, снимаемого с фоточувствительных элементов видеокамеры вследствие теневого шума. Динамические модели включают в себя дискретные модели, в которых переменная может принимать только одно из конечного множества значений, и непрерывные модели, в которых для переменных возможны любые значения из определенного интервала.

По возможности реализации (способу представления) модели можно разделить на две группы. Материальные модели представляют собой объекты реального мира или их части; всегда имеют реальное воплощение и воспроизводят геометрические и физические свойства объекта-оригинала. К классу материальных моделей относят три подгруппы моделей. Натуральные модели подразумевают проведение исследования на реальном объекте и обработку результатов эксперимента с применением теории подобия, то есть воспроизведение объекта с другими размерами. Физические модели подразумевают проведение исследования на установках, которые сохраняют природу явления и обладают физическим подобием объекту-оригиналу, то есть объект другой физической природы, но с аналогичным поведением. Абстрактные модели представляют собой мысленные представления, сформированные человеком в результате раздумья и зафиксированные в обозначениях определенной степени условности. К ним относят наглядные модели (моделируемые процессы и явления протекают

наглядно, и модель, как правило, полностью внешне повторяет объект или изображает его схематически, например, фотография или рисунок), знаковые модели (модель системы представляет логический объект, в котором основные свойства и отношения реального объекта выражены системой знаков или символов, являющихся частью естественного или искусственного языков, например, конспект лекций), математические модели (модели, записанные с помощью системы математических объектов, позволяющих получить характеристики реального объекта).

По сфере применения все модели можно разделить на универсальные модели, представляющие собой готовый, ранее разработанный математический аппарат для исследования различных задач, систем, явлений, и специализированные модели, созданные для исследования и описания конкретных задач, явлений и систем.

В некоторых литературных источниках под классификацией по сфере применения подразумевают разделение по отраслям деятельности человека, например, математические, биологические, химические, социальные, экономические, исторические.

4.3 Методы и технологии моделирования

В общем случае технология решения задачи с использованием компьютерного моделирования включает следующие основные этапы (рисунок 4.3).



Рисунок 4.3 – Технология компьютерного моделирования

Начинается моделирование с постановки задачи, при которой происходит описание исходных данных, определение целей моделирования и требований к

результатам. Далее разрабатывается концептуальная модель. На данном этапе происходит формулировка предположений и упрощений, выделение взаимосвязей и существенных элементов объекта исследования. Этот этап также называют структуризация, или качественный анализ. Результаты этапа кратко описывают или представляют в виде схем, диаграмм, чертежей.

После этого проектируется непосредственно математическая модель, этот этап также называют формализацией задачи. Заключается он в описании объекта на искусственном математическом языке с использованием наиболее адекватного математического аппарата. В этом случае происходит описание не самого объекта, а его ранее разработанной концептуальной модели.

Алгоритмизация задачи включает в себя разработку алгоритма расчета представленной математической модели в виде четкой и однозначной последовательности операций, которые могут быть выполнены на компьютере.

Далее осуществляется кодирование алгоритма с помощью выбранных средств программирования, например, языка программирования, электронных таблиц, набора запросов, форм и отчетов.

Тестирование компьютерной системы применяется для выявления и устранения ошибок. Этот этап является обязательным и, зачастую, наиболее длительным, потому что создать сразу программу без ошибок невозможно.

Использование компьютерной системы необходимо для решения поставленных задач, объяснения функционирования моделируемой системы, прогнозирования ее поведения, принятия управленческих решений.

Интерпретация результатов компьютерного моделирования служит для формулирования выводов из анализа явления и следствий полученных прогнозов, выработки мер по реализации выбранного решения.

Процесс решения задачи с использованием компьютерного моделирования является циклическим. Это означает, что на любом из этапов можно осуществить возврат на любой из предыдущих этапов и выполнить внесение изменений. Даже на этапе интерпретации результатов может оказаться, что полученные выводы неприменимы практически, и необходимо вернуться к постановке задачи, пересмотрев состав исходных данных и цели моделирования.

4.4 Информационная модель объекта

В связи с развитием информационных технологий и увеличением мощности вычислительных систем использование информационного моделирования увеличилось в десятки раз. В первую очередь это связано с тем, что большинство задач решается только с помощью компьютера, то есть практически все модели на стадии расчетов переводятся в информационную, а далее и компьютерную модели. Так, например, информационная модель в своем содержании опирается на математический аппарат, а, следовательно, математическую модель, а для решения математической модели используется компьютер, то есть вновь происходит переход к информационной модели. Рассмотрим более подробно основные понятия информационной модели.

Информационный объект представляет собой описание реального объекта, процесса или явления в виде совокупности его характеристик, которые называют **реквизитами**. Структура информационного объекта образует **класс**, которому присваивается собственное уникальное имя.

Конкретные характеристики информационного объекта называют **экземплярами**. Для идентификации экземпляра используются **ключи**, в качестве которых могут выступать реквизиты, причем, при некоторых условиях одни те же реквизиты могут быть как ключевыми, так и описательными. Один реквизит может иметь несколько ключей.

Примеров информационных систем достаточно много как универсальных, так и специализированных. К универсальным информационным моделям относят базы данных, автоматизированные системы управления, базы знаний, экспертные системы, системы автоматизированного проектирования.

Специализированные модели уникальны, предназначены для решения конкретно поставленных задач, являются более дорогостоящими и, как правило, разрабатываются на заказ, например, комплекс программного обеспечения, предназначенный для проектирования, анализа и моделирования оптических систем с произвольно расположенными в пространстве оптическими элементами.



Рисунок 4.4 – Типы информационных моделей

Выделяют три типа информационных моделей: табличные, иерархические и сетевые. В **табличных** информационных моделях объекты и их свойства представлены в виде списка, а их значения размещаются в ячейках прямоугольной формы. В **иерархических** моделях объекты распределены по уровням. Каждый элемент верхнего уровня состоит из элементов нижнего уровня, а элемент нижнего уровня может входить в состав только одного элемента более высокого уровня. **Сетевые** информационные модели применяют для изображения систем, в которых связи между элементами имеют сложную структуру.

4.5 Математическое моделирование с помощью Mathcad

Одним из самых популярных математических пакетов является Mathcad, который способен быстро, просто и эффективно выполнять сложные математические расчеты, включая различные оптические задачи. Рассмотрим более подробно возможности Mathcad версии 13 для математического моделирования в оптике.

После запуска приложения Mathcad 13 появляется стандартное окно, содержащее строку заголовка, панель меню, панель инструментов, панель форматирования, строку состояния и рабочую область с новым документом.

Панель меню содержит девять пунктов: **File**, **Edit**, **View**, **Insert**, **Format**, **Tools**, **Symbolics**, **Window** и **Help**. В каждом из представленных пунктов меню сгруппированы команды Mathcad в зависимости от их назначения.

В меню **File** расположены команды для работы с документом. Команды меню **Edit** предназначены для редактирования документа. Меню **View** содержит команды, позволяющие выполнять настройку внешнего вида рабочего окна документа. Меню **Insert** необходимо для вставки в документ таких объектов, как графики, матрицы, функции, единицы измерения, текст, рисунки, гиперссылки. В меню **Format** представлены команды форматирования объектов и элементов документа. В меню **Tools** содержатся команды настройки параметров документа, опции выполнения вычислений, проверки орфографии. В меню **Symbolics** собраны команды для выполнения символьных вычислений. Команды меню **Window** необходимы для управления окнами, открытыми на момент сеанса работы документов. Меню **Help** позволяют получать справку по различным аспектам работы с Mathcad.

Выполнять основные операции в Mathcad удобнее всего с использованием панелей инструментов. Их изучение в приложении Mathcad 13 наиболее эффективно производить, решая конкретные задачи, которые найдут применение в профессиональной деятельности оптотехника.

4.5.1 Выполнение арифметических операций

Очень часто приходится решать задачи, которые содержат большое количество арифметических операций. Рассмотрим возможности Mathcad на примере аналитического определения положения и величины изображения по заданным значениям фокусного расстояния f_l , размера предмета y , расположенного на расстоянии a (рисунок 4.5).

Сначала задаются значения переменных f_l , y , a . Для этого нужно записать название переменной, отобразить палитру **Evaluation** математической панели инструментов, выбрать оператор **Definition**, соответствующий операции присваивания, после которого указать значение переменной. Далее необходимо определить положение изображения по формуле отрезков, для этого необходимо записать название переменной a_l и выбрать оператор **Definition**. После этого нужно отобразить палитру **Calculator** математической панели инструментов, выбрать оператор **Division**, соответствующий делению двух чисел, и составить дробь, в числителе которой указать единицу, а знаменателе – сумму двух дробей, составленных из выше определенных переменных. Аналогично вычисляется линейное увеличение линзы β и размер изображения y_l . Для вывода значения, хранящегося в переменной, достаточно отобразить палитру **Calculator** математической панели инструментов, выбрать оператор **Evaluate Numerically**.

Аналогично можно выполнять арифметические операции над числами и выражениями.

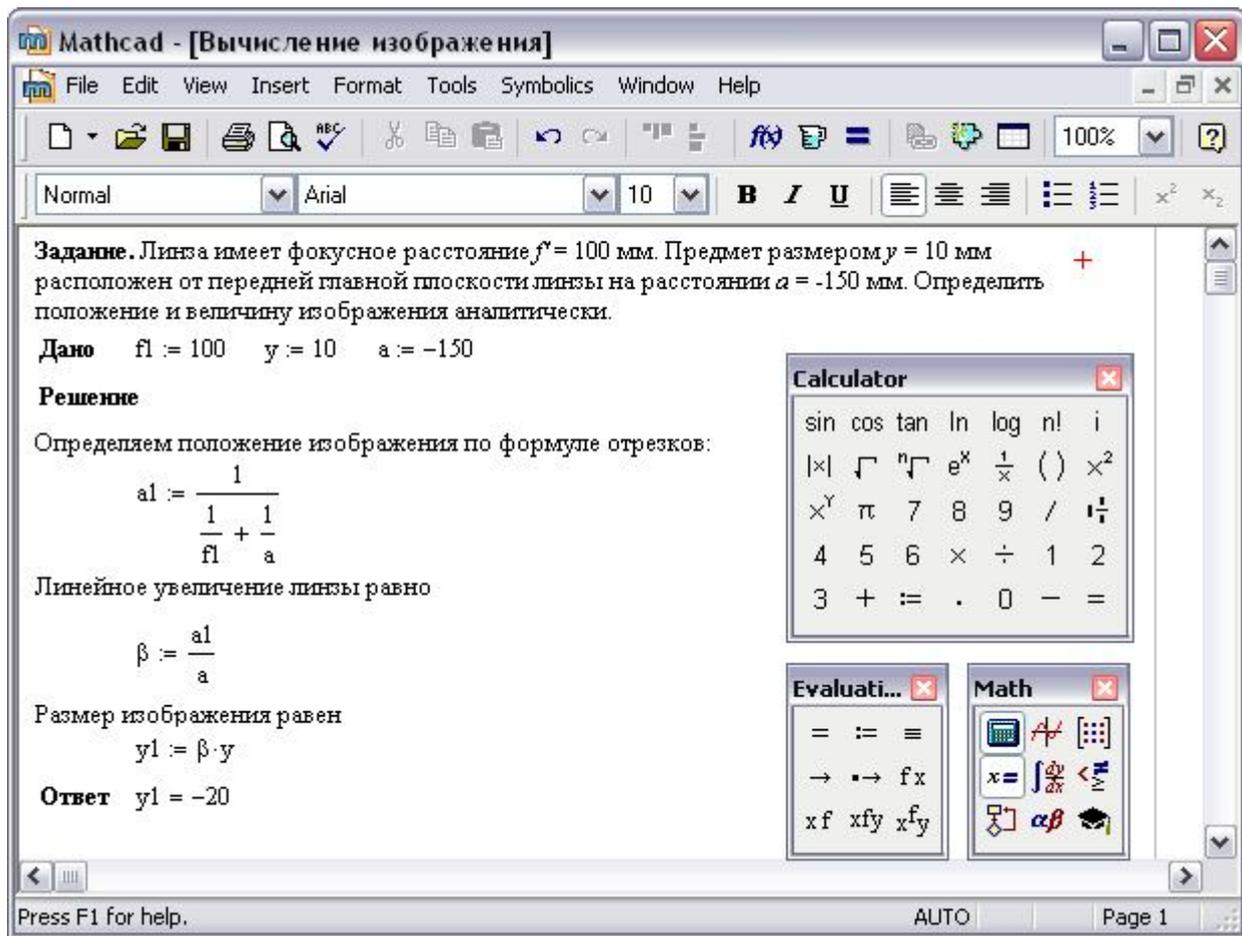


Рисунок 4.5 – Моделирование аналитического определения положения и величины изображения

4.5.2 Решение систем линейных уравнений

Приведем пример использования Mathcad для решения систем нелинейных уравнений, которые очень часто приходится решать при моделировании оптических систем. Для этого решим задачу моделирования задней апертуры репродуктивной оптической системы. Зададимся линейным увеличением β , длиной L , размером входного зрачка $D1$, показателем преломления $n1$ (рисунок 4.6).

Сначала переменным $a1$ и a , относительно которых будет решаться система уравнений, присваиваются начальные значения. Затем вводится ключевое слово **Given**, означающее начало системы уравнений. После этого необходимо ввести два уравнения системы, что соответствует числу переменных. Первое уравнение системы уравнений определяет длину оптической системы, второе уравнение характеризует ее увеличение. Между левой и правой частями уравнений необходимо поставить оператор логического равенства. Для этого нужно

отобразить палитру **Evaluation** математической панели инструментов и выбрать оператор **Evaluate Numerically**, соответствующий операции логического равенства.

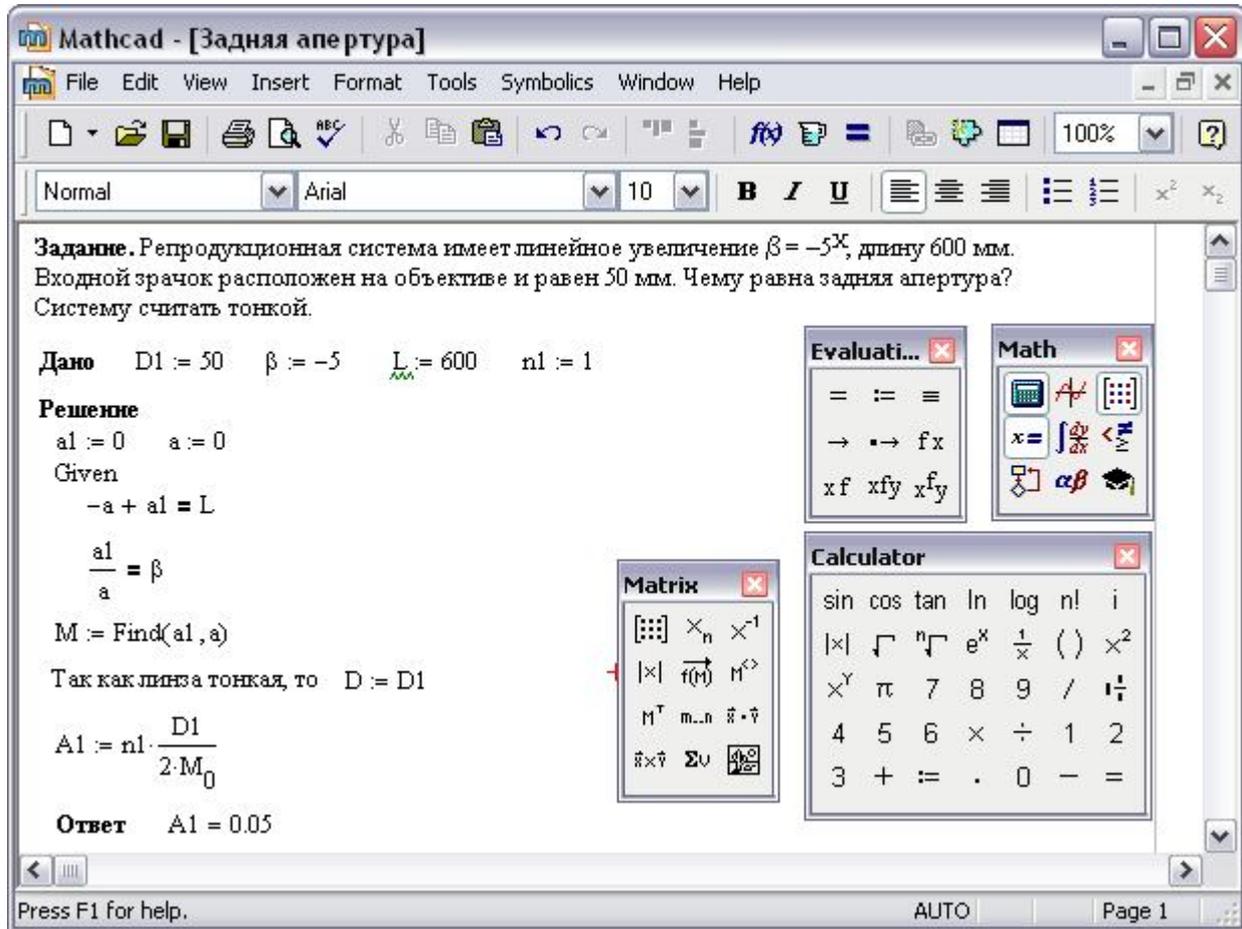


Рисунок 4.6 – Моделирование репродуктивной оптической системы

После того как система задана, переменной M присваивается значение функции **Find**, аргументами которой являются переменные системы уравнений. Функция **Find** возвращает матрицу решений соответствующего размера. Для того чтобы получить решение системы уравнений, необходимо вывести палитру **Matrix** математической панели инструментов и выбрать пункт **Subscript**, после чего ввести имя матрицы M и указать порядковый номер переменной системы уравнений, начиная с нуля. Полученное решение можно использовать при построении арифметических выражений для вычисления задней апертуры репродуктивной оптической системы A_1 .

4.5.3 Построение графиков функций

Очень важным преимуществом Mathcad является наличие эффективных средств визуализации математического моделирования. В большинстве случаев

грамотно построенный график упрощает восприятие математической модели и позволяет более глубоко понять ее физический смысл.

Построим график изменения интенсивности света от толщины поглощающих веществ в соответствии с законом Бугера-Ламберта-Бера (рисунок 4.7). Переменной I_0 присваивается начальное значение интенсивности света. Далее определяется функция изменения интенсивности света I при его распространении через поглощающую среду, имеющая экспоненциальную зависимость относительно двух переменных: k – коэффициента поглощения, x – толщины среды.

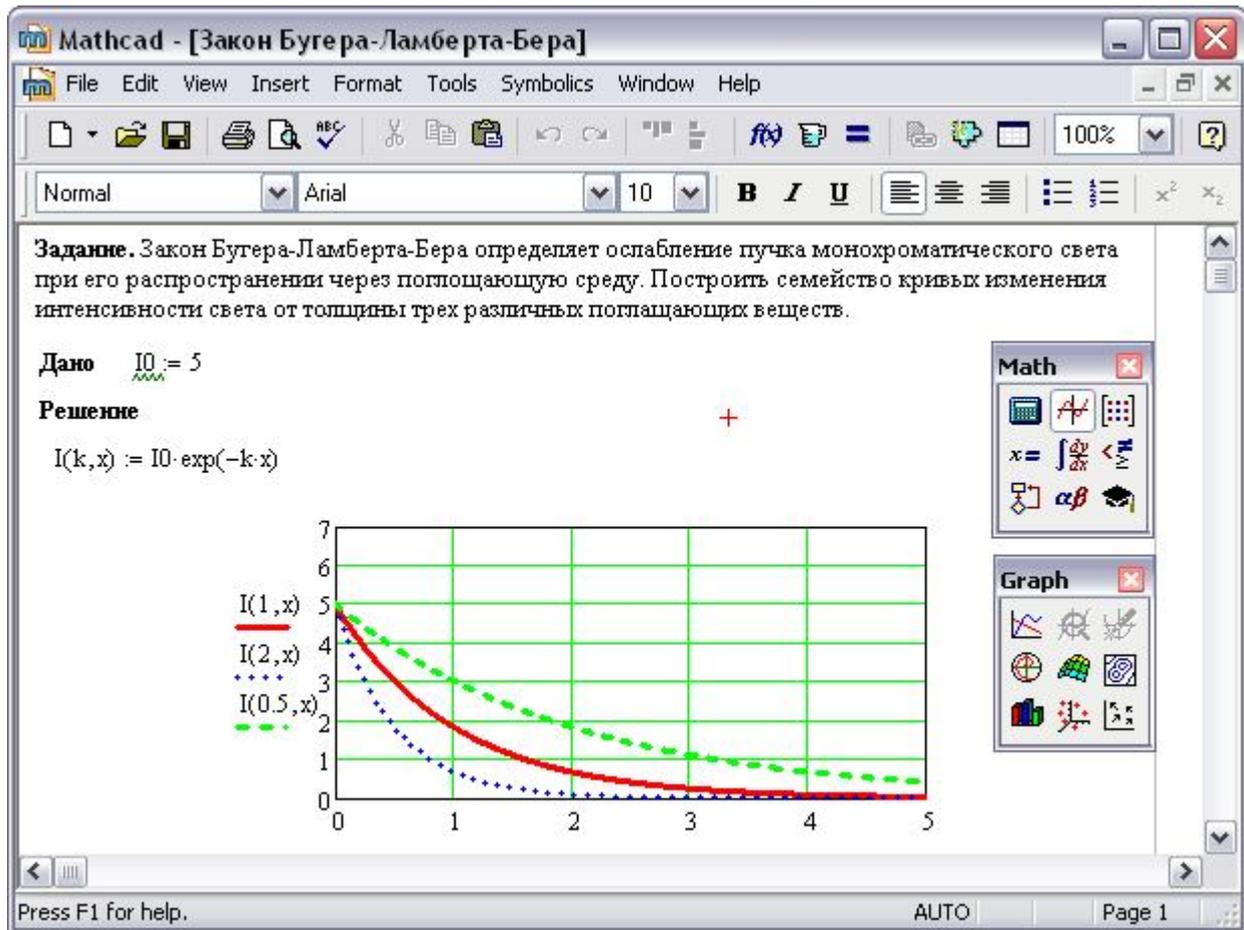


Рисунок 4.7 – Построение семейства кривых изменения интенсивности света

Для построения графика функции $I(k, x)$ необходимо отобразить палитру **Graph** математической панели инструментов, выбрать пункт **X-Y Plot**. В появившейся области графика необходимо заполнить несколько полей. В нижней части графика ввести название переменной, в нашем случае x , а также указать диапазон изменения этой переменной, например, от нуля до пяти. В левой области графика необходимо ввести название заранее определенной функции I , аргумент k которой будет постоянным, а x будет изменяться в заданном диапазоне. Границы изменения функции редактируются аналогично. В Mathcad можно строить в одной области несколько графиков функций. Для этого после оп-

ределения функции необходимо ввести символ «,» и в появившемся поле указать имя следующей функции.

Формат отображения графических данных настраивается с помощью вызова контекстного меню, в котором выбирается пункт **Format**. В появившемся окне **Formatting Currently Selected X-Y Plot** на закладке **X-Y Axes** определяется вид области построения графика функции, а на закладке **Trace** настраивается цвет, толщина, тип линии графика функции.

На рисунке 4.7 построено семейство кривых изменения интенсивности света от толщины трех поглощающих веществ с коэффициентами поглощения 0,5, 1 и 2. Как видно из представленных графиков, интенсивность при увеличении толщины вещества падает по экспоненциальному закону. Следует отметить, чем больше коэффициент поглощения, тем круче наклон графика.

4.5.4 Работа с матрицами

В Mathcad предусмотрены широкие возможности для работы с матрицами. Самым простым способом создания матрицы является вызов пункта **Matrix or Vector** палитры **Matrix** математической панели инструментов. После чего необходимо присвоить элементам матрицы соответствующие значения. Чтобы получить значение элемента матрицы, необходимо выбрать пункт **Subscript** палитры **Matrix** математической панели инструментов, ввести имя матрицы и поставить знак равенства. При этом будут выведены все значения массива. Если необходимо обратиться к определенному значению матрицы, то нужно после указания имени матрицы перейти в режим ввода нижних индексов и указать номера строки и столбца соответствующего элемента.

Более интересным способом заполнения является инициализация матриц с помощью функции их индексов. Рассмотрим возможности этого способа на примере построения интерференционной картины, создаваемой двумя протяженными источниками света.

Для этого первоначально задается интенсивность света I_0 , показатель преломления среды n_1 , расстояние между источниками света d , длина волны источника света λ , расстояние до экрана l (рисунок 4.8).

Затем необходимо инициализировать индексы массива i и j значениями-диапазонами. Для этого указывается имя индекса, устанавливается знак присваивания. Далее на палитре **Matrix** математической панели инструментов выбирается элемент **Range Variable** и задаются начальные и конечные значения диапазона.

Для того чтобы инициализировать матрицу, необходимо на палитре **Matrix** выбрать элемент **Subscript** и указать имя матрицы I , название индексов i и j через запятую. Далее нужно поставить знак присваивания и ввести функцию расчета интенсивности с аргументами i и j .

Следовательно, матрица I будет содержать численные значения интенсивностей интерференционной картины. Чтобы графически представить содержимое матрицы I , необходимо на палитре **Matrix** математической панели инст-

рументов выбирать элемент **Picture** и указать имя матрицы I . В результате на экране отобразится искомая интерференционная картина.

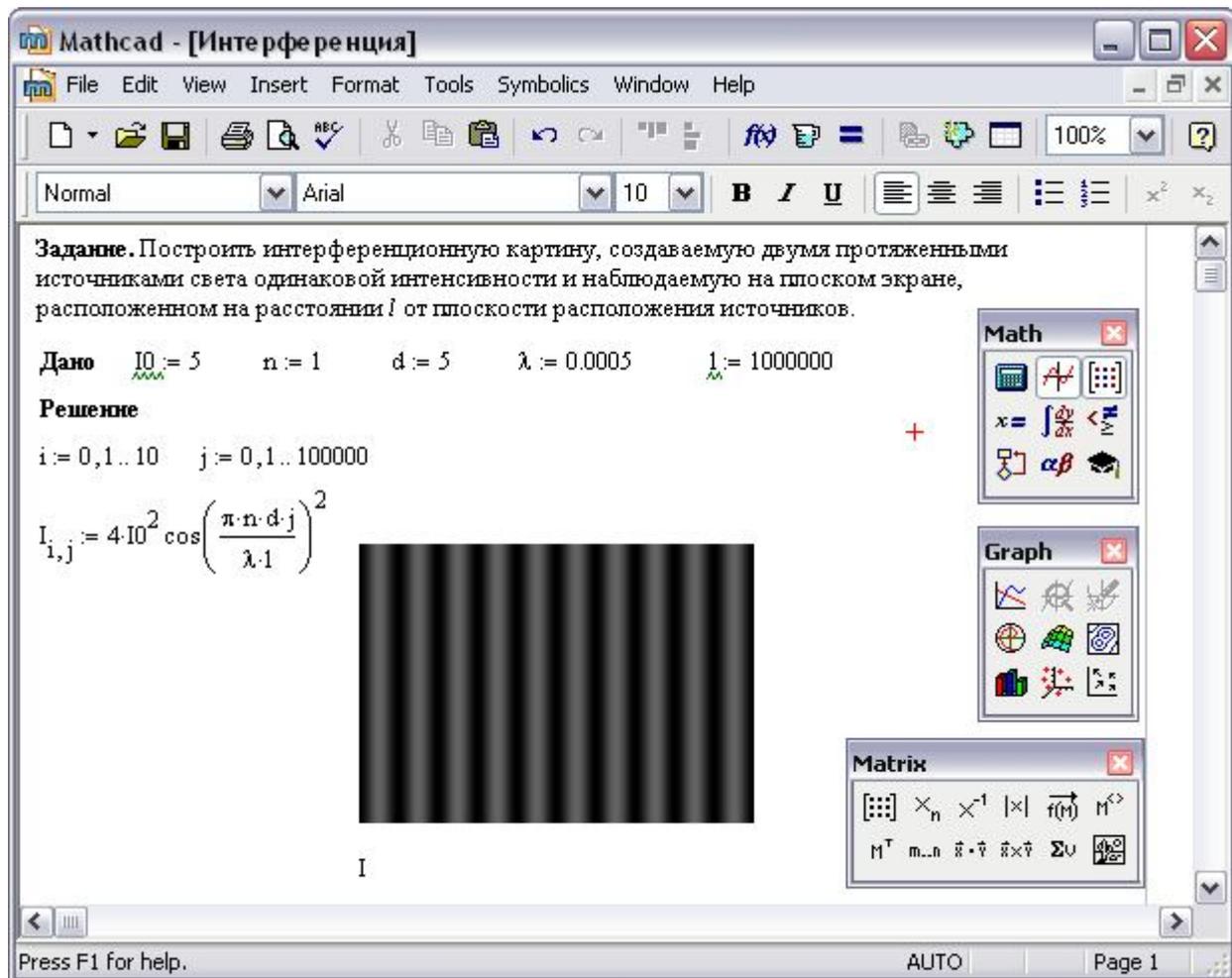


Рисунок 4.8 – Построение интерференционной картины

4.5.5 Обработка изображений

Для оптотехников важным аспектом математического моделирования является умение обрабатывать цифровые изображения. Изображение можно вставить в Mathcad из файла, для этого необходимо на панели меню выбрать пункт **Insert**, после чего выбрать команду **Picture**. В результате выполнения этой команды в рабочую область вставляется объект изображения, в котором необходимо указать имя графического файла.

Вставим в рабочую область два изображения кратеров Луны, указав полный путь и имя соответствующего графического файла. После выделения изображения появляется палитра **Picture Toolbar**. С помощью данной палитры можно производить простейшую обработку изображения. Например, для второго изображения можно применить операцию **Flip Horizontally** и **Flip Vertically**, для того чтобы произвести симметричное отображение относительно горизон-

тальной и вертикальной осей. При этом наблюдается известный оптический обман, когда кратеры превращаются в горы (рисунок 4.9).

Для более сложной обработки изображения необходимо получить информацию о цветном изображении с помощью команды **READRGB**, о черно-белом изображении – с помощью команды **READBMP**. После чего можно производить математический анализ изображения и его изменение, работая с ним как с двумерным массивом или матрицей, элементами которой являются яркости пикселей изображения.

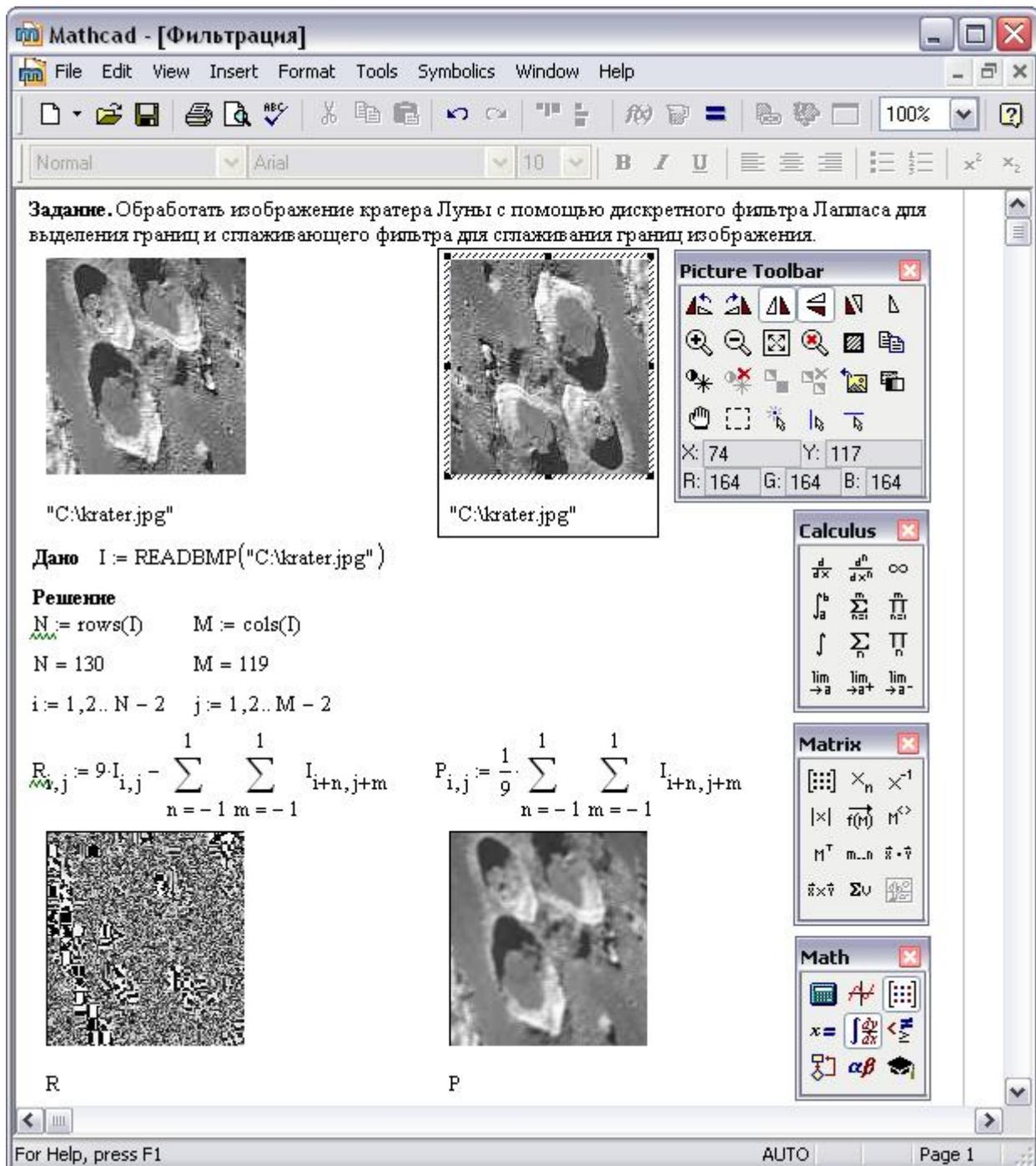


Рисунок 4.9 – Обработка изображения

Поместим яркости черно-белого изображения кратеров Луны с помощью команды **READBMP** в матрицу I , для того чтобы в первом случае произвести выделение границ изображения, во втором случае – сглаживание границ изображения.

Получим размер исходного изображения с помощью функции `rows`, возвращающей количество строк матрицы, и функции `cols`, возвращающей количество столбцов матрицы. Как видно из рисунка 4.9, размер исходного изображения составляет 130×119 пикселей.

Для выделения границ изображения обычно используют дискретный фильтр Лапласа, суть которого заключается в том, что яркость каждого пикселя выходного изображения равна сумме разностей яркости этого пикселя исходного изображения и яркости соседних пикселей.

Для сглаживания границ изображения обычно используют сглаживающий фильтр, суть которого заключается в том, что яркость каждого пикселя выходного изображения равна среднему значению яркостей соседних пикселей исходного изображения.

Чтобы произвести эти операции над исходным изображением, нужно создать индексные переменные i , которая соответствует номеру строки и изменяется от единицы до значения высоты исходного изображения, уменьшенного на два, и j , которая соответствует номеру столбца и изменяется от единицы до значения ширины исходного изображения, уменьшенного на два. В данном случае необходимо учитывать тот факт, что размер выходного изображения, вследствие обработки, уменьшается на два пикселя как по ширине, так и по высоте.

Для этого необходимо после указания имен переменных i и j поставить знак присваивания, на палитре **Matrix** выбрать элемент **Range Variable** и указать начальное и конечное значения диапазона.

Затем соответствующему элементу новых массивов $R_{i,j}$ и $P_{i,j}$ необходимо присвоить результат расчета яркостей пикселей выходных изображений по вышеописанным алгоритмам. При этом необходимо воспользоваться элементом **Summation** на палитре **Calculus** математической панели инструментов.

Чтобы представить матрицы в графическом виде, необходимо на палитре **Matrix** математической панели инструментов выбирать элемент **Picture** и указать имя матрицы, содержащей значения яркостей обработанного изображения, в данном случае R и P .

4.5.6 Интегрирование и дифференцирование функций

Первая попытка выполнения автоматических операций не над числами, а над выражениями, была предпринята Чарльзом Бэббиджем еще в 1836 году при разработке аналитической вычислительной машины, которая, к сожалению, не увенчалась успехом. С тех пор важным преимуществом современного средства вычислений является возможность выполнения аналитических операций.

Пакет для математического моделирования Mathcad позволяет интегрировать и дифференцировать различные функции, выполнять над ними преобра-

зования Лапласа и Фурье, что особенно важно для оптотехников. Рассмотрим применение символьных операций на примере гармонических функций.

Для этого сначала зададим функцию f , например, от одного аргумента x вида $\cos(x) + 2\sin(2x)$ (рисунок 4.10).

Для вычисления неопределенного интеграла выберем пункт **Indefinite Integral** палитры **Calculus** математической панели инструментов, укажем ранее заданную функцию $f(x)$ и запустим команду **Symbolic Evaluation** палитры **Symbolic** математической панели инструментов. В результате выполнения этих действий в рабочей области появится результат интегрирования.

Для вычисления производной выберем пункт **Derivative** палитры **Calculus** математической панели инструментов, укажем ранее заданную функцию $f(x)$ и запустим команду **Symbolic Evaluation** палитры **Symbolic** математической панели инструментов. В результате выполнения этих действий в рабочей области появится результат дифференцирования.

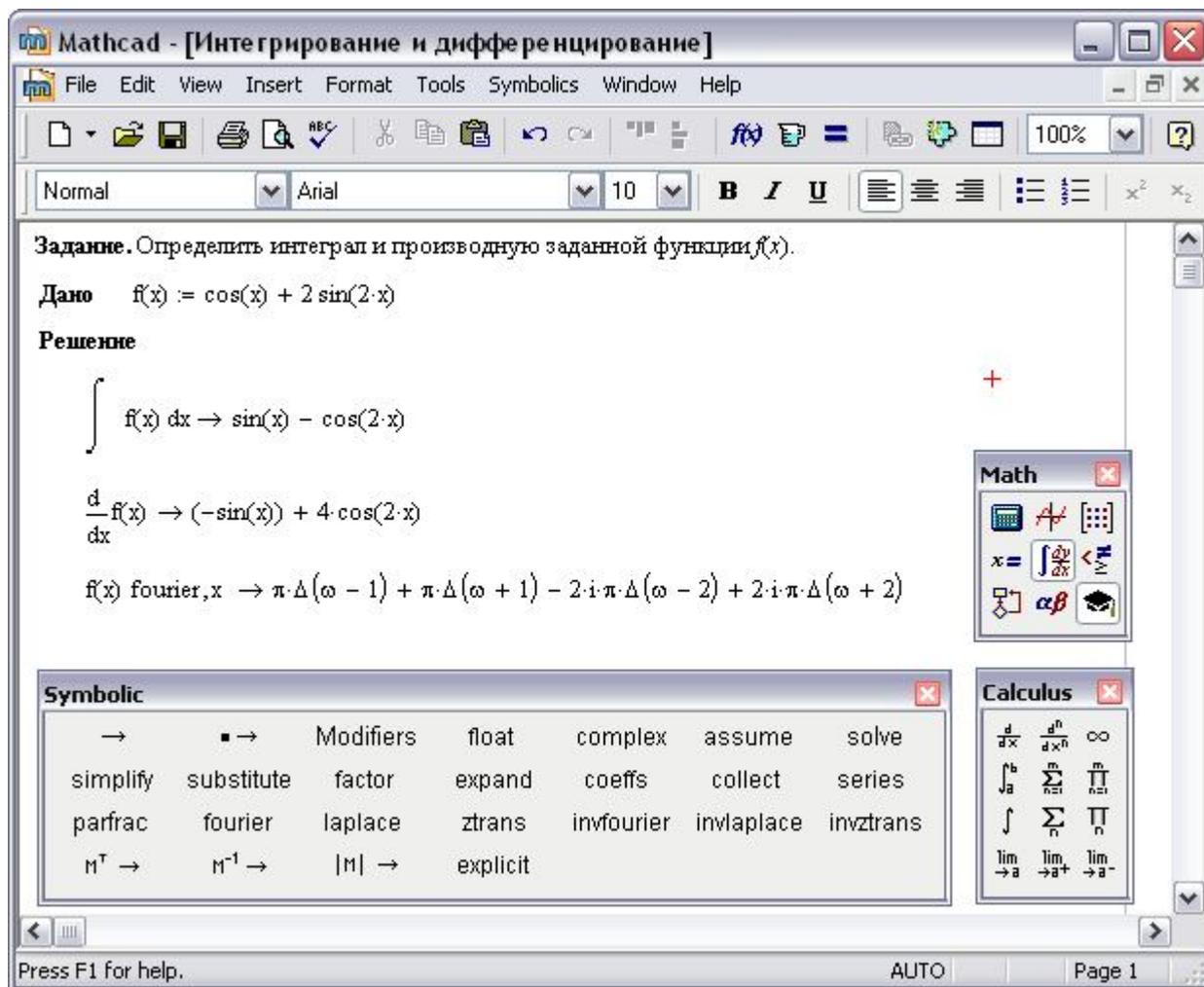


Рисунок 4.10 – Интегрирование и дифференцирование гармонических функций

Очень часто инженерам, в частности оптотехникам, приходится применять преобразование Фурье от различных функций. Для этого необходимо вы-

брать пункт **Fourier Transform** палитры **Symbolic** математической панели инструментов, указать сначала функцию $f(x)$, над которой выполняется преобразование Фурье, затем выделить переменную x , по которой выполняется преобразование, и только после этого запустить команду **Symbolic Evaluation** палитры **Symbolic** математической панели инструментов. В результате выполнения этих действий в рабочей области появится результат преобразования Фурье – набор δ -функций, которые в пакете Mathcad обозначаются символом « Δ ».

Тезаурус

Вычислительные и функциональные задачи, моделирование, прототип, модель, теория моделирования, аналитическое моделирование, имитационное моделирование, информационный объект, класс, реквизит, экземпляр, ключ.

Контрольные вопросы

1. Назовите отличия вычислительных и функциональных задач.
2. Дайте определение термину модель.
3. В каких случаях модель адекватна рассматриваемому объекту?
4. Когда применяется имитационное моделирование?
5. В чем состоит основная особенность стохастических моделей?
6. Приведите пример дискретной динамической модели.
7. Зачем проводится тестирование системы при решении задачи?
8. Для чего в информационной модели объекта применяются ключи?
9. Постройте с помощью Mathcad интерференционную картину, возникающую при отражении света в тонкой воздушной прослойке.
10. Получите с помощью Mathcad спектр функции $f(x) = 2 \cdot \exp(-3 \cdot x^2)$.

Список рекомендуемой литературы

1. Алиев, Т. И. Основы моделирования дискретных систем [Текст] / Т. И. Алиев. – СПб . : СПбГУ ИТМО, 2009. – 363 с.
2. Васильев, А. Н. Mathcad 13 на примерах [Текст] / А. Н. Васильев. – СПб . : БХВ-Петербург, 2006. – 528 с.
3. Могилев, А. В. Информатика [Текст] / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. – М . : Академия, 2004. – 848 с.
4. Половко, А. М. Mathcad для студентов [Текст] / А. М. Половко, И. В. Ганичев. – СПб . : БХВ-Петербург, 2006. – 336 с.
5. Смирнов, А. П. Компьютерное моделирование измерительных процессов : практикум в среде MathCAD на примерах из механики и оптики [Текст] / А. П. Смирнов. – СПб . : СПбГУ ИТМО, 2006. – 101 с.
6. Советов, Б. Я. Моделирование систем [Текст] / Б. Я. Советов, С. А. Яковлев. – М . : Высшая школа, 2001. – 343 с.
7. Соболев, Б. В. Информатика [Текст] / Б. В. Соболев, А. Б. Галин, Ю. В. Панов. – Ростов н/Д : Феникс, 2007. – 446 с.

ГЛАВА 5 КОМПЬЮТЕРНЫЕ СЕТИ

В настоящей главе рассматриваются общие вопросы, связанные с техническими средствами компьютерной коммуникации, защитой информации в локальных и глобальных компьютерных сетях, сетевыми технологиями обработки данных, сетевыми сервисами и стандартами.

5.1 Сетевые технологии обработки данных

Значение компьютерных сетей для современного общества сложно переоценить. Компьютерные сети сегодня являются не просто средством связи технических систем, но и средством коммуникации огромного числа пользователей. В общем смысле компьютерная сеть представляет собой совокупность компьютеров, физически связанных друг с другом, которые передают данные между участниками сети. Компьютер, образующий сеть, называют **хостом** (host – хозяин). Хостом может являться как полноценный компьютер, так и, например, сетевой принтер.

Одним из основных понятий в теории компьютерных сетей является термин **ресурс**. Компьютерные сети предоставляют доступ к ресурсам. Ресурсы сети бывают трех видов: аппаратные, программные и информационные (рисунок 5.1).



Рисунок 5.1 – Виды ресурсов компьютерных сетей

К аппаратным ресурсам относится, например, жесткий диск, выполняющий функции файлового сервера. В качестве программных сетевых ресурсов может выступать программное обеспечение, установленное на сервере и позволяющее ускорить процесс обработки информации другим пользователям сети. Информационный ресурс представляет собой данные, хранящиеся на удаленных компьютерах. Сегодня такими ресурсами в наиболее полном объеме располагает глобальная сеть Интернет.

Все компьютерные сети можно разделить на две категории: локальные компьютерные сети и глобальные компьютерные сети. Примером локальной компьютерной сети является сеть Ethernet, а глобальной – сеть Интернет. Такое деление сетей, в первую очередь, связано с использованием специализированных протоколов. В данном случае под **протоколом** понимаются специализированные стандарты, которые обеспечивают совместимость всех уровней архи-

тектуры компьютерных сетей. Протоколы, определяющие характер взаимодействия компонентов сети, называют аппаратными, а протоколы, обеспечивающие взаимодействие программ и данных, называют программными.



Рисунок 5.2 – Категории компьютерных сетей

Единый комплект протоколов для всех участников используется в локальных сетях. **Локальные сети**, как правило, достаточно компактны, длина такой сети может составлять от 2 метров до 3 километров, может объединять компьютеры одного этажа, корпуса или группы рядом находящихся зданий. **Глобальные сети** имеют большие территориальные размеры и могут объединять в себе регионы, страны и континенты, а также включать отдельные локальные сети. Глобальные компьютерные сети могут использовать различные сетевые протоколы.

В сетевых технологиях принято различать участников сети. В первую очередь это связано с распределением ресурсов между участниками сети и прав доступа к этим ресурсам. У участников сети права доступа к общим сетевым ресурсам могут значительно различаться. Совокупность приемов разделения и ограничения прав участников компьютерной сети называют **политикой сети**. Управление сетевой политикой называется **администрированием** компьютерных сетей.

5.2 Топологии компьютерных сетей

Существует множество вариантов соединения компьютеров в сеть. Самое простое соединение состоит из двух компьютеров, в этом случае в качестве аппаратных средств используются стандартные порты ввода-вывода, а в качестве программного обеспечения могут выступать встроенные в операционную систему приложения. Но, как правило, сеть включает в себя более двух компьютеров, в этом случае установка прямого соединения невозможна, тогда компьютерная сеть строится на основе определенной топологии. **Топология** представляет собой способ организации физических связей компьютеров и прочих сетевых компонентов. В зависимости от вида топологии выбирается необходимое оборудование для монтажа сети, просчитывается возможность расширения сети и выбирается способ ее управления, которое базируется на разделении обязанностей. Существует четыре основные сетевые топологии: шина, звезда, кольцо и ячеистая топология.

5.2.1 Топология «шина»

Топология «шина» является наиболее простой из всех топологий и довольно распространенной. В топологии «шина» все компьютеры подключаются к одному общему кабелю (рисунок 5.3). Данные в топологии «шина» передаются всем компьютерам, но принимает их только тот компьютер, чей адрес совпадает с адресом получателя, который зашифрован в сигнале. В каждый момент времени передачу может вести только один компьютер, все остальные находятся в состоянии приема. Поэтому производительность сети при топологии «шина» очень сильно зависит от количества компьютеров. Чем больше компьютеров в сети, тем больше время ожидания, следовательно, медленнее работает вся сеть.

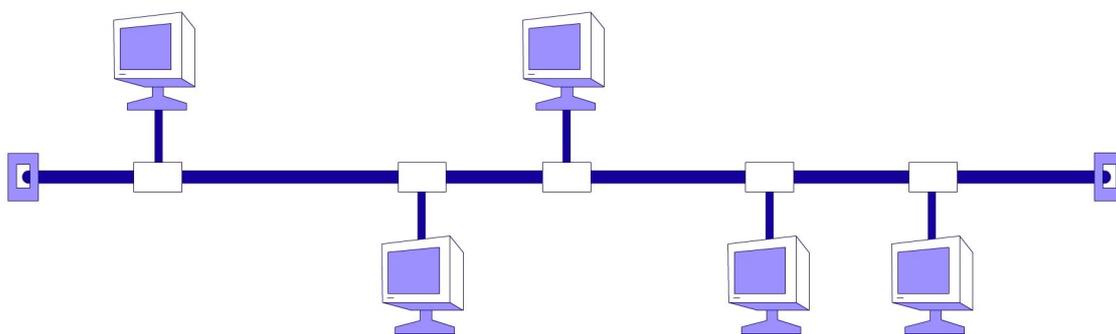


Рисунок 5.3 – Компьютерная сеть с топологией «шина»

К параметрам, влияющим на быстродействие сети с топологией «шина», относятся тип аппаратного обеспечения сетевых компьютеров, частота, с которой компьютеры передают данные, тип работающих сетевых приложений, тип сетевого кабеля, расстояние между компьютерами в сети.

Электрические сигналы в топологии «шина» распространяются по всему кабелю. Сигналы, достигшие концов кабеля, отражаются от них, в результате возникает наложение сигналов, что приводит к искажению и ослаблению основного сигнала. Для того чтобы сигнал не отражался от концов шины, используют специальное устройство, называемое терминатором.

Нарушение общего кабеля или терминаторов приводит к выходу из строя участка сети или всей сети в целом. Отключение какого-либо компьютера на работу компьютерной сети не влияет.

5.2.2 Топология «звезда»

В топологии «звезда» все компьютеры с помощью сегментов кабеля подключаются к центральному устройству, в качестве которого может выступать отдельный компьютер или концентратор, по этому признаку различают топологии с активной или пассивной звездой соответственно (рисунок 5.4). Сигналы

от передающего компьютера поступают через центральное устройство ко всем остальным компьютерам сети.

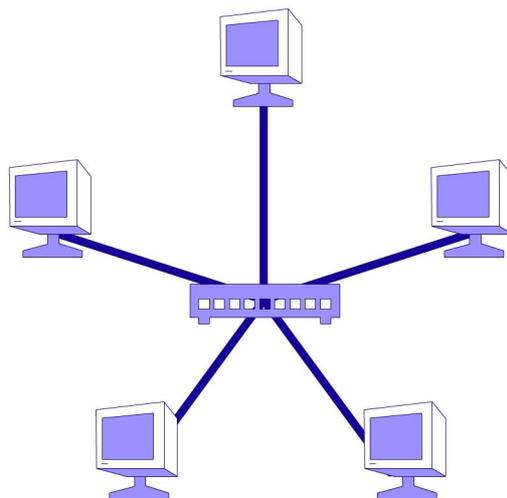


Рисунок 5.4 – Компьютерная сеть с топологией «звезда»

Топология «звезда» имеет четко выделенный центр, к которому подключены все остальные компьютеры сети. Обмен всей информацией происходит через центральный компьютер, который, как правило, имеет более высокие технические характеристики и сложное сетевое оборудование. Центральный компьютер занимается только обслуживанием сети. Такое устройство топологии позволяет повысить отказоустойчивость сети, полный отказ сети возможен только в случае выхода центрального компьютера, отказ одного из периферийных компьютеров на работу остальной сети не влияет.

Сегодня в мире наиболее популярна топология с пассивной звездой. Концентратор в пассивной звезде принимает сигналы от периферийных компьютеров, восстанавливает их и пересылает другим компьютерам. Схема построения пассивной и активной звезды идентичны.

Существенным недостатком топологии «звезда» является поддержка подключения к центральному компьютеру ограниченного числа периферийных компьютеров, а также большой расход кабеля при ее проектировании.

5.2.3 Топология «кольцо»

Топология «кольцо» предполагает последовательную передачу информации в сети от одного компьютера к другому. Данные передаются только в одну сторону. В топологии «кольцо» каждый компьютер соединяется линиями связи с двумя другими компьютерами (рисунок 5.5). Причем от одного компьютера он только получает информацию, а другому – только передает. Выделенного центрального устройства в топологии «кольцо» не существует, все компьютеры равноправны, но может быть выделена специальная рабочая станция, которая управляет сетью и контролирует обмен информацией.

Обмен информацией в топологии «кольцо» происходит с помощью **маркеров**, которые представляют собой специальную последовательность битов, передающихся по сети. В каждой сети существует только один маркер. Маркер передается по кольцу последовательно от одного компьютера к другому до тех пор, пока его не захватит тот компьютер, который хочет передать данные.

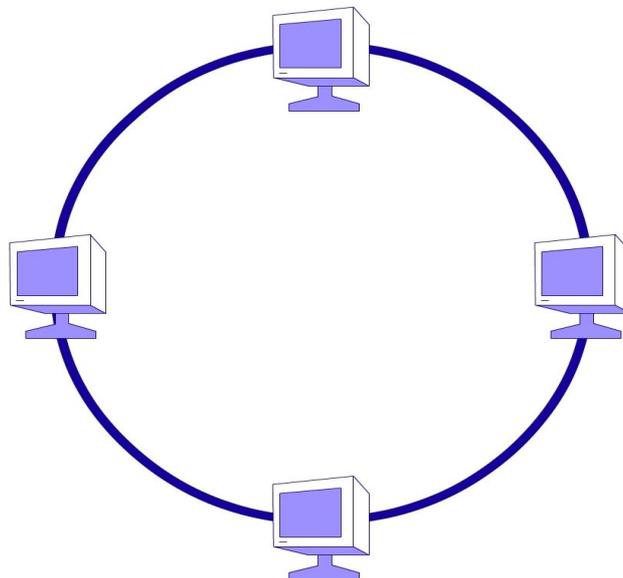


Рисунок 5.5 – Компьютерная сеть с топологией «кольцо»

В топологии «кольцо» каждый компьютер не только принимает сигнал и передает его дальше, но и восстанавливает его, поэтому проблем с затуханием сигнала в кольце не существует, главное, чтобы затухание сигнала не происходило между двумя соседними компьютерами. Благодаря такой ретрансляции можно строить компьютерные сети на основе топологии «кольцо» протяженностью до нескольких десятков километров. Топология «кольцо» может поддерживать подключение большого количества компьютеров, устойчива к перегрузкам и обеспечивает работу с очень большими потоками данных. При выходе из строя одного компьютера или повреждении кабеля вся сеть выходит из строя, поэтому к одному компьютеру протягивают два кабеля.

5.2.4 Ячеистая топология

Ячеистая топология предполагает схему соединения компьютеров в сеть, при которой каждый компьютер соединен со всеми рядом стоящими компьютерами (рисунок 5.6). Эта топология является самой устойчивой к отказам и перегрузкам из всех рассмотренных ранее. Существенным недостатком ячеистой топологии являются большие затраты на прокладку кабеля.

В ячеистой топологии сигнал от компьютера отправителя до компьютера получателя может пройти по разным маршрутам, поэтому разрыв какого-либо

участка сети на работе всей остальной сети не сказывается. Такая топология используется при построении глобальной сети Интернет.

Кроме базовых топологий существуют их комбинации. Чаще всего используются две комбинированные топологии: «звезда-шина» и «звезда-кольцо».

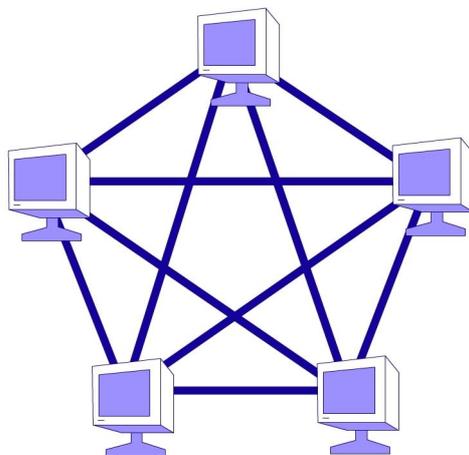


Рисунок 5.6 – Компьютерная сеть с ячеистой топологией

5.3 Линии передачи данных в компьютерных сетях

В большинстве случаев компьютеры в сетях физически соединены между собой с помощью линий связи. В локальных компьютерных сетях информация передается последовательно, бит за битом. Параллельная передача данных также возможна, но физически реализовать ее намного сложнее. Во-первых, при параллельной передаче информации компьютеры соединяются не менее, чем восемью проводами одновременно, что очень дорого даже в случае небольших расстояний между участниками сети. Кроме того, длина прокладываемых линий связи должна быть абсолютно одинакова, что также достаточно тяжело выполнить при увеличении расстояния между компьютерами. Поэтому, несмотря на то, что последовательная передача данных по сети медленнее и сложнее, чем параллельная, используется она намного чаще.

К основным характеристикам линий связи относят следующие параметры: частотный диапазон сигналов, характеризующий затухание сигнала в линии связи, скорость распространения сигнала, помехозащищенность линии связи, волновое сопротивление кабеля, которое важно для электрических кабелей и зависит от формы и взаиморасположения электрических проводников внутри кабеля, материала диэлектрика и технологии его изготовления.

5.3.1 Проводные линии передачи данных

Существует большое количество различных линий связи, выпускаемых промышленно. Все их виды строго стандартизированы и имеют достаточно точные характеристики. Наибольшую популярность получили проводные ли-

нии связи из-за простоты монтажа и низкой стоимости кабеля. Выделяют три основных вида проводных линий связи: кабели на основе витой пары, коаксиальные и оптоволоконные кабели.

5.3.1.1 Витая пара

Витая пара является сегодня самым дешевым видом линий связи. Кабель на основе витой пары состоит из нескольких пар скрученных изолированных медных проводов, объединенных в одной пластиковой оболочке. Структура кабеля на основе витой пары представлена рисунке 5.7.

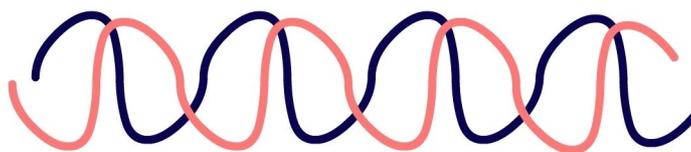


Рисунок 5.7 – Структура кабеля на основе витой пары

Как правило, данный вид кабеля состоит из двух или четырех пар витых проводов, является очень гибким и его легко прокладывать в помещениях. Существуют два вида кабелей на основе витой пары: экранированный и неэкранированный. Неэкранированные витые пары обладают слабой защищенностью, подвержены влиянию внешних электромагнитных полей и легко прослушиваются. Для увеличения помехозащищенности используют экранированные кабели на основе витой пары. В этом случае каждая пара помещается в металлическую оплетку, которая является экраном и значительно снижает воздействие помех на медные провода внутри кабеля и взаимные наводки между парами. Несмотря на все недостатки, наиболее распространенным кабелем является неэкранированная витая пара, так как является более дешевым и легко монтируемым видом кабеля. На основе витой пары, в основном, строятся сети с топологией «звезда» или «кольцо». Скорость передачи данных в таких линиях связи не превышает 100 Мбит/с.

5.3.1.2 Коаксиальный кабель

Основу коаксиального кабеля составляет электрический кабель, состоящий из центрального медного провода и металлической оплетки, между которыми находится диэлектрик. Диэлектрик является внутренней изоляцией кабеля. Все составляющие коаксиального кабеля помещены в общую внешнюю оболочку из пластика. На рисунке 5.8 представлена структура коаксиального кабеля, по которому передается электрический сигнал.

По сравнению с витой парой коаксиальный кабель имеет более высокую помехоустойчивость и допустимые скорости передачи данных. В основном ко-

аксиальный кабель применяют при монтаже сетей с топологией «шина». На концах коаксиального кабеля всегда устанавливаются терминаторы для предотвращения внутреннего отражения сигнала, причем волновое сопротивление терминатора должно совпадать с волновым сопротивлением кабеля, а один из терминаторов должен быть заземлен.

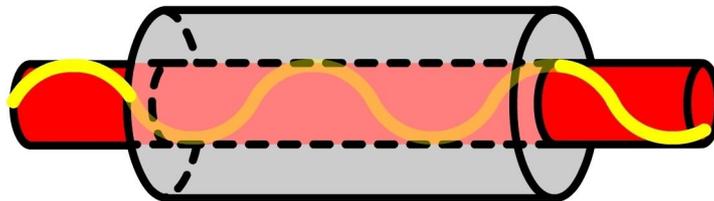


Рисунок 5.8 – Структура коаксиального кабеля

Существует два вида коаксиальных кабелей: тонкий и толстый. Тонкий коаксиальный кабель имеет диаметр 5 мм и является более гибким. Он может прокладываться с любыми изгибами и не требует дополнительной фиксации. Сигнал в нем затухает быстрее, чем в толстом кабеле, поэтому он используется в небольших по протяженности сетях.

Толстый коаксиальный кабель имеет диаметр 10 мм. Он очень жесткий, при прокладке требует жесткой фиксации. При монтаже толстого коаксиального кабеля используют специальное оборудование для прокалывания его оболочки и установления контакта с жилами. Толстый коаксиальный кабель имеет более высокий уровень помехозащищенности по сравнению с тонким коаксиальным кабелем. На сегодняшний день коаксиальный кабель используется очень редко из-за его стоимости, сложности в прокладке и низких технических характеристик. Допустимое значение скорости передачи данных в компьютерных сетях, элементы которых связаны с помощью коаксиальных линий связи, равно 500 Мбит/с.

5.3.1.3 Оптоволоконный кабель

В оптоволоконном кабеле информация распространяется по оптическим волокнам в виде модулированных световых импульсов. Основу оптоволоконного кабеля составляет прозрачное стекловолокно. Стекловолокно передает свет на расстояния до десятков километров с незначительным ослаблением. Структура оптоволоконного кабеля представлена на рисунке 5.9.

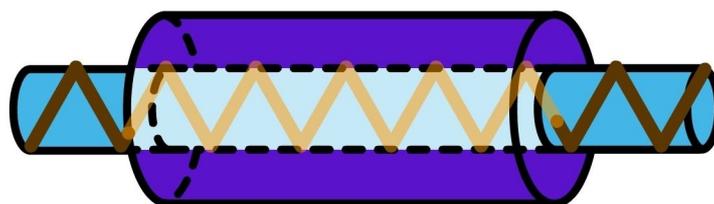


Рисунок 5.9 – Структура оптоволоконного кабеля

В центре оптоволоконного кабеля находится стекловолокно диаметром от 1 до 10 мкм. В качестве внутренней изоляции используется стеклянная или пластиковая оболочка, которая не позволяет свету выходить за пределы стекловолокна. За счет использования изоляции достигается эффект полного внутреннего отражения света от границы двух веществ с разными коэффициентами преломления, в частности, коэффициент преломления света у стеклянной оболочки значительно ниже, чем у центрального оптоволокна. В качестве защиты от внешних механических воздействий кабель помещают в металлическую оплетку, причем в одной оплетке может содержаться сразу несколько оптоволоконных кабелей.

Оптоволокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон, одно из которых используется для передачи данных, а другое – для приема данных. Скорость передачи в оптоволоконной сети составляет от 100 Мбит/с, теоретически данные по оптоволоконным линиям связи можно передавать со скоростью до 200 Гбит/с.

По уровню помехозащищенности и секретности передаваемой информации оптоволоконный кабель имеет самые высокие характеристики среди всех линий связи. Световой сигнал не искажается при действии внешних электромагнитных полей, а сам оптоволоконный кабель не испускает внешних электромагнитных излучений. Прослушать оптоволоконный кабель практически невозможно, так как при несанкционированном доступе оптоволокно будет повреждено, что приведет к потере работоспособности сети.

Пропускная способность оптоволоконного кабеля очень высокая и достигает 1 кГц. Величина затухания светового сигнала при увеличении частоты практически не увеличивается, что является основным недостатком электрических кабелей. Стоимость самого простого оптоволоконного кабеля сегодня сопоставима со стоимостью тонкого коаксиального кабеля, однако цена всей оптоволоконной сети значительно увеличивается за счет использования дополнительного оборудования, которое преобразует электрические сигналы в световые импульсы.

Оптоволоконный кабель применяется в сетях с топологией «звезда» или «кольцо». Кабель обеспечивает очень хорошую гальваническую развязку компьютеров в сети и не требует согласования и заземления.

Оптоволоконные кабели имеют несколько недостатков. Монтаж сети на основе оптоволокна является самым сложным и трудоемким. Это связано с тем, что при установке разъемов на оптоволокне срезы кабеля выполняются с микронной точностью, чтобы на стекловолокне не осталось сколов. Если центральное стекловолокно будет иметь сколы и низкую полировку срезов, то световой сигнал будет сильно затухать при передаче. Для установки разъемов применяют специальный гель, которым склеивают стекловолокно. Гель имеет такой же коэффициент преломления света, что и само стекловолокно. Вышеописанный процесс является дорогостоящим. Поэтому на практике используют готовые нарезанные куски оптоволокна разной длины с установленными разъемами.

Оптоволоконный кабель имеет ограниченное количество разветвлений кабеля. При большом количестве разветвлений световой сигнал будет сильно ослабляться и до конца сети может вообще не доходить. По гибкости оптоволоконный кабель уступает электрическим кабелям, так как радиус сгиба оптоволоконного кабеля составляет от 10 до 20 см. На затухание сигнала в оптоволоконном кабеле сильно влияют ионизирующие излучения, они вызывают помутнение стекловолокна. Сегодня выпускают специальные кабели из стекла, стойкого к радиации, но его цена намного дороже простого стекловолокна. Этот тип кабеля также не любит перепадов температуры, в результате которых стекловолокно может потрескаться.

Прочность оптоволоконного кабеля низкая. Он чувствителен к таким механическим действиям, как удары или действие ультразвука, которые вызывают в оптоволокне микрофонный эффект, для устранения которого используют специальные мягкие звукопоглощающие оболочки.

Существует два вида оптоволоконных кабелей: одномодовый и многомодовый кабель. Диаметр центрального оптоволокна в одномодовом кабеле составляет 1,3 мкм, и световой сигнал, передаваемый по данному кабелю, имеет такую же длину волны. За счет того, что в одномодовом кабеле все световые лучи проходят одинаковое расстояние и достигают фотоприемников одновременно, дисперсия и искажения сигнала малы даже при больших сетевых расстояниях. В качестве передающих и принимающих устройств в одномодовом кабеле используют лазер. Одномодовый оптоволоконный кабель является наиболее перспективной технологией связи компьютеров.

В многомодовом оптоволоконном кабеле форма сигнала на принимающем устройстве может значительно искажаться. Это связано с тем, что световые лучи внутри оптоволокна могут иметь разные траектории движения. Диаметр внешней оболочки многомодового кабеля составляет 125 мкм, а центрального оптоволокна 62,5 мкм. Длина световой волны в многомодовом кабеле равна 850 нм, а допустимая длина кабеля не более 5 км. Для передачи и приема данных используются светодиоды, которые имеют большой срок службы и стоят значительно дешевле лазерных устройств.

5.3.2 Беспроводные линии передачи данных

В последнее время большое распространение получили беспроводные линии связи, основным преимуществом которых является мобильность компьютеров в сети, что достигается отсутствием проводов. Беспроводные сети могут быть гибридными, то есть использовать несколько видов соединения компьютеров в рамках одной сети.

В беспроводных компьютерных сетях применяется способ передачи данных «точка-точка». Это означает, что обмен данными происходит только между компьютерами. Для организации такой связи необходимо использовать дополнительные устройства, которые называются трансиверы. **Трансивер** представляет собой устройство для подключения компьютера к сети, осуществляющее

прием и передачу сигналов. Термин образован от двух английских слов передатчик-приемник (TRANSMitter-reCEIVER).

Существуют одиночные трансиверы и хост-трансиверы, которые могут работать с автономными компьютерами или с компьютерами, уже подключенными к какой-либо сети. В кабельных сетях трансивер встроен в сетевой адаптер, в беспроводных сетях он представляет собой отдельное устройство. В беспроводных сетях трансивер также называют **точкой доступа**.

В большинстве случаев данные в беспроводных сетях передаются с помощью инфракрасного излучения и с помощью радиопередачи в узком диапазоне или рассеянном спектре. Рассмотрим каждый из этих видов подробнее.

5.3.2.1 Инфракрасные линии передачи данных

В инфракрасных беспроводных сетях используют для передачи данных инфракрасные лучи. В подобных системах необходимо генерировать очень мощный сигнал, чтобы сгладить влияние на него других источников света. Скорость передачи в таких сетях высока и составляет 10 Мбит/с (рисунок 5.10).

Различают 4 типа инфракрасных сетей: сети прямой видимости между приемником и передатчиком, сети на рассеянном излучении, когда сигнал от передатчика достигает приемника, отражаясь от стен и потолка, сети на отраженном излучении, широкополосные оптические сети.

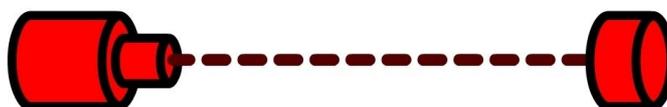


Рисунок 5.10 – Схема передачи данных в инфракрасных беспроводных сетях

Среди основных достоинств инфракрасных сетей можно отметить высокую скорость передачи данных и отсутствие чувствительности к электромагнитным полям. К недостаткам использования этого класса сетей можно отнести трудности при передаче сигналов на большие расстояния, а также подверженность помехам со стороны сильных источников света. В последнее время интенсивно развиваются технологии построения лазерной беспроводной сети, которая строится на основе применения в качестве передатчика лазерного источника. Необходимым условием функционирования такой сети является прямая видимость между компьютерами.

5.3.2.2 Радиопередача данных

Радиоканал использует в качестве носителя информации радиоволны. Скорость передачи данных в радиосетях может достигать 10 Мбит/с и зависит от выбора длины волны и способа кодирования информации. Радиопередача данных применяется как в персональных сетях, так и в глобальных компьютер-

ных сетях. Существует два способа передачи данных по радиосетям в зависимости от ширины полосы передачи данных: узкополосные и широкополосные.

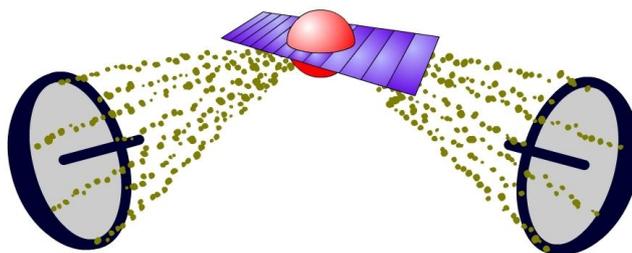


Рисунок 5.11 – Схема передачи данных в беспроводных радиосетях

При узкополосной радиопередаче пользователи настраивают передатчики и приемники на определенную частоту. Соединение компьютеров может быть выполнено без ретрансляторов или с их применением, например, при использовании сотовой или спутниковой связи (рисунок 5.11). Выполнение условия прямой видимости между компьютерами в данном случае не обязательно.

Одной из главных проблем построения широкополосных радиосетей является решение задачи доступа многих пользователей к ресурсам сети. Добиться этого можно с использованием разделения между компьютерами пространства, времени, частоты или кода.

Метод разделения пространства основан на том, что беспроводное устройство может вести передачу данных только в пределах одной определенной территории, на которой любому другому устройству запрещено передавать свои сообщения. Однако на одной территории, как правило, функционируют несколько устройств.

Метод разделения частоты базируется на том, что каждое устройство работает на строго определенной частоте, благодаря чему несколько устройств могут вести передачу данных на одной территории. Однако этот подход требует выделения отдельной частоты каждому беспроводному устройству.

Метод разделения времени основан на том, что каждый передатчик транслирует сигнал на одной и той же частоте, но в различные промежутки времени, при этом процесс передачи данных в сети синхронизирован. В данном случае, временные интервалы могут динамично перераспределяться между устройствами сети: устройствам с большим трафиком назначаются более длительные интервалы, чем устройствам с меньшим объемом трафика.

Метод разделения кодов базируется на том, что каждый передатчик транслирует сигнал на одной частоте, в одно время, но с разными кодами. Каждому передатчику сети присваивается собственный уникальный код, с помощью которого шифруются все биты исходных данных. Приемник, зная уникальный код нужного передатчика, дешифрует все сообщения от всех передатчиков сети, если сигнал на выходе дешифратора превышает установленное значение, значит, это сообщение от нужного приемника. Сообщения других передатчиков сети воспринимаются приемником как шум. Достоинством такого подхода является высокая помехозащищенность и скрытность передачи данных.

В отдельную категорию беспроводной передачи данных выделяют мобильные сети, которые в качестве передающей среды используют телефонные сотовые сети. Отличительная особенность сотовых сетей состоит в том, что передача информации происходит с помощью равномерно распределенных по всей территории ретрансляторов. **Ретранслятор** представляет собой оборудование связи, которое соединяет два или более радиопередатчиков, удаленных друг от друга на большое расстояние.

Существует также радиосеть на основе пакетного соединения. В этом случае данные разбиваются на пакеты, в которых содержится адрес источника, адрес приемника и информация для коррекции ошибок. Все пакеты передаются на спутник, который их транслирует в широкоэвещательном режиме. Затем устройства с соответствующим адресом принимают эти пакеты.

Одной из самых популярных технологий беспроводной передачи данных является Bluetooth, которая обеспечивает беспроводную связь между компьютерами, сотовыми телефонами, приборами, датчиками на расстоянии до 100 м. Спецификация Bluetooth описывает пакетный способ передачи данных с разделением информации во времени. Радиообмен происходит в полосе частот от 2400 до 2483,5 МГц, которая применяется во многих странах для безлицензионного доступа.

5.4 Сетевое оборудование

Сетевым оборудованием называется набор устройств, которые используются для создания и обеспечения правильной работы сети. Сетевое оборудование работает вместе с сетевой операционной системой, в которой определены права пользователей и разграничен доступ к ресурсам сети. Сетевое оборудование бывает двух типов: пассивное и активное.

Пассивным оборудованием называют устройства, предназначенные только для передачи сигнала в сети без его анализа. К пассивному сетевому оборудованию относят проводники и приспособления для их организации и защиты.

Активным оборудованием называют устройства, которые определяют свои дальнейшие действия в зависимости от значения принятого сигнала. Для активного оборудования, как правило, необходим отдельный источник питания. Рассмотрим основные виды сетевого оборудования.

Сетевая карта предназначена для сопряжения компьютера с сетью. Сетевая карта обеспечивает обмен данными между компьютером и линией связи в соответствии с принятыми правилами обмена информацией. Этот вид оборудования выпускается во многих вариантах для различных интерфейсов компьютера. К основным параметрам сетевых карт относятся скорость передачи данных, поддержка возможности загружать компьютер без жесткого диска по сети, поддержка режима одновременного приема и передачи данных.

Концентратор представляет собой устройство для соединения линий связи для сетей с любой логической топологией. Различают активные и пассив-

ные концентраторы. Активный концентратор способен принимать данные, регенерировать их и передавать дальше в сеть. Пассивный концентратор принимает сигнал и пересылает его в сеть без усиления или восстановления. Основными характеристиками концентраторов являются количество портов, определяющее максимальное число соединяемых линий связи, скорость передачи данных и тип сетевого носителя.

Повторитель предназначен для увеличения расстояния сетевого соединения путем повторения электрического сигнала. Повторитель принимает сигнал, заново формирует его аналоговый вид, усиливает и передает сигнал другому сегменту сети. Повторитель действует на электрическом уровне для соединения двух сегментов.

Коммутатор в большинстве случаев применяется для увеличения размеров сети за счет разбиения на подсети, а также обеспечивает функции по управлению трафиком в сети. Коммутатор является центральным устройством сети, так как к нему подключаются все остальные сетевые устройства и компьютеры. Управление коммутатором осуществляется с помощью сетевой операционной системы.

Маршрутизатор представляет собой устройство, позволяющее переадресовывать и маршрутизировать пакеты из одной сети в другую, а также фильтровать широковещательные сообщения. Маршрутизатор использует адрес получателя, указанный в пакетах данных, и определяет по специальной таблице маршрутизации путь, по которому следует передать данные. Таблица маршрутизации состоит из некоторого числа записей, называемых маршрутом. Маршрут содержит такую информацию, как адрес сети получателя, адрес следующего узла, которому следует передавать пакеты, и некоторый вес записи, который называют метрикой. Метрики записей в таблице помогают вычислять кратчайшие маршруты к различным получателям. Если в таблице маршрутизации для адреса нет описанного маршрута, пакет отбрасывается.

Мостом является устройство, соединяющее два отдельных ограниченных по длине сетевых сегмента и обеспечивающее передачу трафика между ними. Мост также усиливает и преобразует сигналы между различными топологиями и линиями связи разных типов, что позволяет расширить размер сети и количество подключенных устройств. Мост аналогичен по своим функциям коммутатору, но внутреннее устройство у них абсолютно разное. Мост имеет в своем составе процессор, который обрабатывает весь трафик.

Шлюз предназначен для преобразования данных из одной формы в другую. Шлюз принимает из одного сегмента сети пакет данных, сформатированный под определенный протокол, и перед отправкой в другой сегмент сети преобразует его в пакет другого протокола. Сетевые шлюзы могут быть реализованы аппаратным или программным способом.

Терминатор представляет собой резистор, который позволяет гасить сигнал на концах сегмента сети. Сопротивление терминатора составляет 50 Ом, но возможны и другие варианты. Терминаторы применяются в топологии «шина».

Каждому сетевому устройству для его идентификации в компьютерной сети присваивается уникальный **MAC-адрес** (Media Access Control – управление доступом к среде), который устанавливается во время изготовления. В широковещательных компьютерных сетях MAC-адрес позволяет доставлять пакеты с данными только нужному сетевому устройству.

5.5 Эталонная модель OSI

Международный институт стандартизации предложил эталонную модель взаимодействия открытых систем OSI (Model of Open System Interconnections), которая позволяет передавать различные данные и работать с разным сетевым оборудованием стандартными способами. Эталонная модель OSI состоит из семи уровней, каждому из которых соответствуют различные сетевые операции, оборудование и протоколы (рисунок 5.12). Рассмотрим каждый из этих уровней более подробно.



Рисунок 5.12 – Модель OSI

Согласно модели OSI архитектуру компьютерных сетей следует рассматривать на разных уровнях, самым нижним из которых является **физический уровень**, отвечающий за передачу двоичных сигналов по линиям передачи данных.

Канальный уровень структурирует данные и обеспечивает их доставку от одного сетевого устройства к другому, при этом данные разбиваются на логические порции, в которых содержится информация об источнике данных и их получателе.

Сетевой уровень отвечает за адресацию сообщений, при этом происходит перевод логических адресов сетевых устройств, которые могут находиться в различных сегментах сети с разными способами адресации, в физические адреса.

Транспортный уровень обеспечивает формирование пакетов данных из передаваемых сообщений и их эффективную доставку от источника до получателя. На данном уровне также решаются задачи поиска ошибок в сообщениях.

Сеансовый уровень позволяет двум приложениям разных компьютеров устанавливать, использовать и завершать соединение, называемое сеансом, а также обеспечивать нужный уровень безопасности этого сеанса.

Уровень представления отвечает за преобразование форматов, используемых для обмена данными между приложениями компьютеров сети, при этом происходит шифровка и расшифровка сообщений, их сжатие и распаковка.

Самым верхним является **прикладной уровень**, который предоставляет доступ прикладным процессам к сетевым возможностям, поддерживает выполнение приложений, управляет потоком данных и обработкой ошибок.

Все сетевые функции, выполняемые на каждом уровне, взаимодействуют только с функциями вышестоящего и нижележащего уровней. Обмен данными в компьютерных сетях происходит путем их перемещения с верхнего на нижний уровень на компьютере отправителя данных. Затем данные по линиям связи передаются на компьютер получателя, на котором происходит обратное воспроизведение путем перемещения данных с нижнего на верхний уровень.

5.6 Сетевые архитектуры

Сетевая архитектура представляет собой комбинацию стандартов, топологий и протоколов, необходимых для создания работоспособной сети. В настоящее время существует две часто используемые архитектуры компьютерных сетей: архитектура Ethernet и архитектура Token Ring. Рассмотрим каждую из них более подробно.

5.6.1 Сетевая архитектура Ethernet

Сетевая архитектура **Ethernet** является архитектурой построения больших локальных компьютерных сетей с количеством компьютеров до 1024.

Передача данных в сетях этого типа возможна со скоростью 10 Мбит/с по коаксиальному кабелю – стандарты 10Base-5 и 10Base-2, по витой паре – стандарт 10Base-T и по волоконно-оптическому кабелю – стандарт 10Base-F. Сегодня также активно применяется данная архитектура для построения беспроводных сетей, это стандарты радио-Ethernet и Fast Ethernet. В настоящее время дополнительно предложены две технологии для передачи данных по сети Ethernet со скоростью 100 Мбит/с – стандарты 100Base-T и 100VG-AnyLAN.

В компьютерных сетях Ethernet для доступа к данным используется протокол CSMA/CD (множественный доступ с контролем несущей и обнаружением конфликтов). В соответствии с этим протоколом устройства начинают передачу данных только после обнаружения свободного канала связи для сокращения между ними количества **коллизий**, когда два сетевых устройства одновременно пытаются передать пакеты данных в компьютерную сеть.

5.6.2 Сетевая архитектура Token Ring

Сетевая архитектура **Token Ring** представляет собой архитектуру построения локальных компьютерных сетей, в которых компьютеры могут производить передачу данных только тогда, когда они владеют **маркером**, непрерывно циркулирующим по кольцу.

Передача данных в сетях этого типа возможна со скоростью 4 или 16 Мбит/с по экранированной, неэкранированной витой паре или оптоволоконному кабелю – стандарт IEEE 802.5. В сетевой архитектуре Token Ring предусмотрена возможность объединения соединительными мостами до 8 колец, в каждом из которых может находиться до 260 сетевых устройств.

5.7 Принцип работы глобальной сети Интернет

Интернет представляет собой глобальное сообщество различных компьютерных сетей. Слово Интернет является аббревиатурой английского выражения «Interconnected Networks», которое переводится как связанные сети. В широком понимании Интернет представляет собой глобальное информационное пространство, организуемое с помощью миллионов компьютеров, связанных друг с другом различными линиями связи. Основным отличием глобальной сети Интернет от других сетей является его децентрализация, так как здесь нет четко выделенного центра, поэтому отказ какого-либо участка сети на работу глобальной сети Интернет не влияет.

Первая сеть (ARPANet), базирующаяся на принципе децентрализации, была разработана 29 октября 1969 года, она имела гибкую маршрутизацию и пакетный способ передачи данных. Тогда предприняли первую попытку дистанционного подключения к компьютеру. Из Калифорнийского университета пытались авторизоваться на компьютере Стэнфордского института. Попытка закончилась крахом системы в момент ввода буквы «G» в слове «LOGIN». Именно этот день и считается днем рождения Интернета.

В результате усовершенствования данной сети и внедрения протоколов передачи данных, таких как TCP/IP, к ней стали подключаться крупные коммерческие предприятия, и ARPANet перешла в Интернет. Глобальное распространение Интернет получил после появления языка гипертекстовой разметки HTML и разработки первого веб-браузера для отображения HTML-документов на экране монитора.

Между компьютерами в Интернет прямое соединение, как правило, отсутствует. Данные, которые они посылают друг другу, разбиваются на пакеты. Разные пакеты даже при одном сеансе связи могут пройти от одного компьютера к другому разными маршрутами, причем пакеты, переданные позже, могут прийти раньше пакетов, посланных первыми. В конечном итоге на компьютере получателя вся переданная информация в полном объеме соберется в нужный документ абсолютно правильно.

5.7.1 Передача данных в глобальной сети Интернет

Для передачи данных в Интернете в большинстве случаев используется протокол ТСП/IP. Этот протокол состоит из двух сетевых протоколов, относящихся к разным сетевым уровням. Протокол ТСП (Transmission Control Protocol) – протокол транспортного уровня. Он управляет тем, как происходит передача информации. Протокол IP (Internet Protocol) – адресный протокол. Он принадлежит сетевому уровню и определяет, куда происходит передача.

С помощью протокола ТСП отправляемые данные делятся на небольшие пакеты, как правило, размер пакета не превышает 1500 байт. Каждый пакет маркируется таким образом, чтобы в нем были данные, необходимые для правильной сборки документа на компьютере получателя. После этого адресный протокол IP доставляет пакеты данных адресату.

Протокол IP работает следующим образом. У каждого участника Интернета есть уникальный IP-адрес, который позволяет организовать точную доставку ТСП-пакетов. Этот адрес записывается с помощью четырех байтов, например, 195.38.46.11. По этим четырем байтам каждый компьютер в сети определяет, кому адресованы ТСП-пакеты, и, если необходимо, пересылает их ближайшему компьютеру. При этом ближайшим будет тот компьютер, который работает с более хорошими условиями связи и линиями с большей пропускной способностью. Выбором маршрута и определением качества связи занимаются маршрутизаторы. Например, маршрутизатор при анализе качества связи может признать ближайшим компьютер, находящийся на другом континенте, но связанный с текущим компьютером высокопроизводительной линией спутниковой связи, чем компьютер, находящийся на соседней улице, но связанный с текущим компьютером с помощью телефонной линии. В результате некоторого числа пересылок ТСП-пакет достигнет своего адресата.

IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети выбирается администратором произвольно. Если необходимо, чтобы сегмент сети работал как составная часть Интернета, то номер сети назначается по рекомендации специального подразделения Интернета InterNIC (Internet Network Information Center). Как правило, поставщики услуг Интернета получают диапазоны адресов у подразделений InterNIC, а затем распределяют их между своими абонентами.

Таким образом, протокол IP осуществляет перемещение данных в сети, а протокол ТСП обеспечивает надежную доставку данных.

5.7.2 Службы глобальной сети Интернет

При работе в Интернете пользователь обычно обращается к одной из его служб. Как правило, работа любой службы Интернета основана на взаимодействии аппаратного и программного обеспечения сервера с клиентским оборудованием и программами.

Служба представляет собой пару программ, взаимодействующих между

собой согласно определенным правилам, которые называют **протоколами**. Одна из программ этой пары называется **сервером**, а вторая – **клиентом**.

Каждая служба для работы использует свой протокол. Протоколы служб называют прикладными протоколами. Для того чтобы воспользоваться какой-либо службой Интернета, пользователю необходимо на своем компьютере установить специальную программу, поддерживающую работу с прикладным протоколом службы.

Электронная почта (E-MAIL) является одной из самых популярных служб Интернета. Обеспечивают работу электронной почты специальные почтовые серверы, которые получают сообщения от клиентов и пересылают их по цепочке к почтовым серверам адресатов, где эти сообщения накапливаются. Основу почтовой службы составляют два протокола: SMTP и POP3. Протокол SMTP обеспечивает отправку почтовой корреспонденции с компьютера на сервер, а протокол POP3 пересылает почту с сервера на компьютер клиента. Эти протоколы всегда работают в паре.

Служба World Wide Web (WWW) является самой распространенной службой Интернета, которая представляет собой единое информационное пространство, состоящее из взаимосвязанных электронных документов, хранящихся на веб-серверах. Отдельные электронные документы называют веб-страницами, тематически объединенные группы веб-страниц образуют веб-сайты. Для просмотра веб-страниц используются специальные программы, которые называются **браузерами**. Браузер выполняет отображение документа на экране в соответствии с командами, внедренными в документ.

Основу службы WWW составляют язык гипертекстовой разметки документов HTML, универсальный способ адресации ресурсов в сети URL и протокол обмена гипертекстовой информацией HTTP (рисунок 5.13).



Рисунок 5.13 – Основа службы WWW

Для создания веб-страниц используется язык гипертекстовой разметки **HTML** (HyperText Markup Language). При этом обычный текстовый документ размечается командами HTML, которые называются **тегами**. Теги отличаются от обычного текста тем, что они заключены в угловые скобки. Большинство тегов состоят из двух частей: открывающего и закрывающего тегов. Сложные теги могут иметь дополнительные атрибуты и параметры, расширяющие возможности форматирования текста.

Адресация документов определяется специальным унифицированным указателем ресурса – **URL** (Uniform Resource Locator), при помощи которого

организуется связь между ресурсом и гиперссылками в Интернете.

Адрес URL состоит из трех частей. Во-первых, определяется служба, которая осуществляет доступ к данному ресурсу (например, для службы WWW прикладным является протокол передачи гипертекста **HTTP** (HyperText Transfer Protocol), после имени протокола ставится последовательность символов «://»). Во-вторых, прописывается доменное имя сервера, на котором хранится нужный ресурс. В-третьих, указывается полный путь доступа к документу на данном сервере. В качестве разделителя используется символ «/» (рисунок 5.14).



Рисунок 5.14 – Адресация документов

Служба имен доменов DNS (Domain Name System) представляет собой распределенную базу данных, которая ставит в соответствие IP-адресам компьютеров, подключенных к глобальной сети Интернет, дружественные для пользователей адреса. Компьютер представлен в сети IP-адресом, состоящим из четырех байтов. Но пользователю с таким представлением адреса ресурса работать неудобно, поэтому было введено уникальное доменное имя, заданное с помощью URL. Перевод доменных имен в связанные с ними IP-адресами выполняют серверы службы имен доменов DNS.

Служба передачи файлов FTP (File Transfer Protocol) используется для передачи информации в Интернете со специализированных серверов, на которых хранятся большие архивы данных. Служба FTP использует в своей работе два TCP-соединения. С помощью одного TCP-соединения обеспечивается управление, а другое используется для пересылки данных.

Для обмена мгновенными сообщениями в Интернете используется **служба ICQ**. Название этой службы является акронимом выражения «I seek you» («Я тебя ищу»). Суть данной службы заключается в том, что если два человека подключены к Интернету, то они могут общаться друг с другом напрямую. Для подключения пользователю необходимо зарегистрироваться на центральном сервере ICQ и получить персональный идентификационный номер. Данный номер сообщается необходимым пользователям сети и между ними устанавливается соединение.

5.8 Информационная безопасность

В информатике понятие безопасности является весьма широким и включает такие направления, как повышение надежности работы компьютера, сохранение ценных данных, защита информации от несанкционированного доступа. Однако ключевым положением в информационной безопасности является

понятие вируса. Большинство программных продуктов, обеспечивающих информационную безопасность компьютера, отслеживают работу вирусов.

5.8.1 Классификация вредоносных программ

Компьютерным вирусом называют программный код, встроенный в программу или документ, который предназначен для несанкционированного пользователем уничтожения, блокирования, модификации или копирования информации, нарушения работы компьютера или компьютерных сетей.

Вирусы можно классифицировать по следующим основным признакам: среда обитания, операционная система, особенности алгоритма работы, деструктивные возможности (рисунок 5.15).



Рисунок 5.15 – Классификация вирусов

По среде обитания вирусы делятся на файловые, загрузочные, макровирусы и сетевые вирусы. Одними из самых распространенных типов компьютерных вирусов являются файловые вирусы, которые внедряются в выполняемые файлы, создают дубликаты файлов или используют особенности организации файловой системы для выполнения несанкционированных действий. Загрузочные вирусы записываются в загрузочный сектор диска и запускаются при загрузке операционной системы, становясь ее частью. Макровирусы поражают документы, выполненные в некоторых прикладных программах, которые имеют средства для исполнения макрокоманд. Например, к таким документам относятся файлы, созданные с помощью пакета программ Microsoft Office, который поддерживает создание макросов на языке программирования Visual Basic for Application. Сетевые вирусы, которые называют сетевыми червями, используют для своего распространения электронную почту, системы обмена мгновенными сообщениями, сети обмена данными, а также недочеты в конфигурации сети и ошибки в работе сетевых протоколов. Существует большое количество сочетаний, например, файловые загрузочные вирусы или файловые сетевые черви.

Еще одним классификационным признаком является вид операционной системы, так как любой вирус ориентирован на заражение файлов или выполнение несанкционированных действий в определенной операционной системе.

По алгоритмам работы выделяют резидентные вирусы и вирусы, использующие стелс-алгоритмы или полиморфичность. Резидентные вирусы при заражении компьютера постоянно остаются в оперативной памяти, перехватывая обращения операционной системы к объектам заражения, чтобы выполнить несанкционированные действия. Такие вирусы являются активными до полного выключения компьютера. Применение стелс-алгоритмов базируется на перехвате запросов операционной системы на чтение или запись зараженных объектов, при этом происходит временное лечение этих объектов, что позволяет вирусам скрыть себя в системе. Вирусы, основанные на применении алгоритмов полиморфичности, не содержат ни одного постоянного участка кода, что достигается за счет шифрования кода вируса и модификации программы-расшифровщика. Как правило, два образца одного и того же вируса не будут иметь ни одного совпадения в коде. Это чрезвычайно усложняет их обнаружение в системе.

По деструктивным возможностям можно выделить неопасные и опасные вирусы. Неопасные вирусы практически не влияют на работоспособность компьютера и не понижают эффективность работы операционной системы, кроме увеличения дискового пространства, которое они занимают. Опасные же вирусы выводят из строя операционную систему и оборудование, портят информацию, хранящуюся на диске.

Разработчики антивирусных программ используют собственные классификации детектируемых вирусов. Наиболее удачная классификация создана в «Лаборатории Касперского» (www.kaspersky.com), которую используют множество других антивирусных компаний. Эта классификация основана на разделении вирусов по типу совершаемых ими на компьютере пользователя действий (рисунок 5.16).

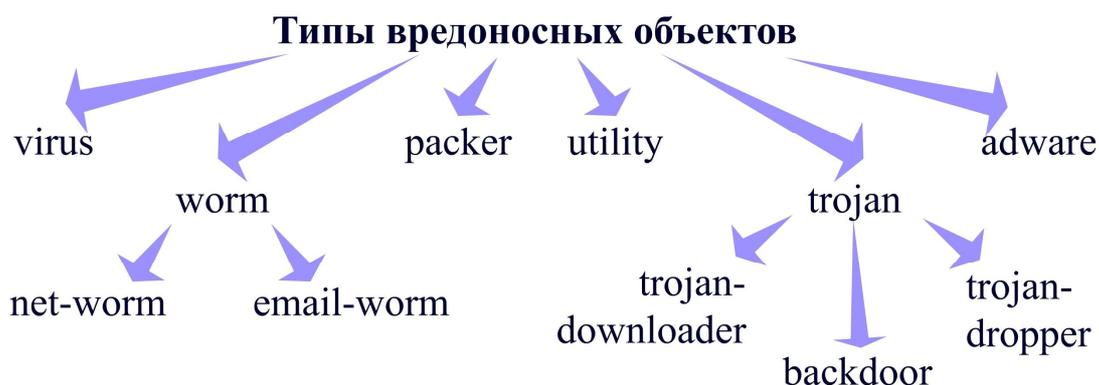


Рисунок 5.16 – Основные типы вредоносных объектов

Среди сетевых вирусов (worm), в основном, выделяют вредоносные программы, которые используют для своего распространения электронную почту (email-worm) и сети обмена данными (net-worm).

Упаковщики (racker) архивируют различными способами содержимое файла, в том числе с помощью шифрования, для того, чтобы исключить корректное разархивирование информации.

Вредоносные утилиты (utility) разрабатываются для автоматизации создания других вирусов, червей или троянских программ, в большинстве случаев они не представляют угрозы компьютеру, на котором исполняются.

В группе вредоносных троянских программ (trojan), проникающих на компьютер под видом безвредного, в основном, свободно распространяемого программного обеспечения, выделяют программы, предназначенные для скрытого удаленного управления пораженным компьютером (backdoor); программы, предназначенные для несанкционированной установки на компьютер различных вирусов, содержащихся в этой программе (trojan-dropper); программы, предназначенные для несанкционированной загрузки на компьютер новых версий вирусов из глобальной сети Интернет (trojan-downloader).

Существуют программы, которые не являются вредоносными, но обладают функциональными возможностями для совершения несанкционированных вредоносных действий, такие программы относятся к категории adware.

Самой большой категорией вредоносных программ являются вирусы, не вошедшие ни в один из вышепредставленных классов, которые предназначены для несанкционированного нарушения работы компьютера.

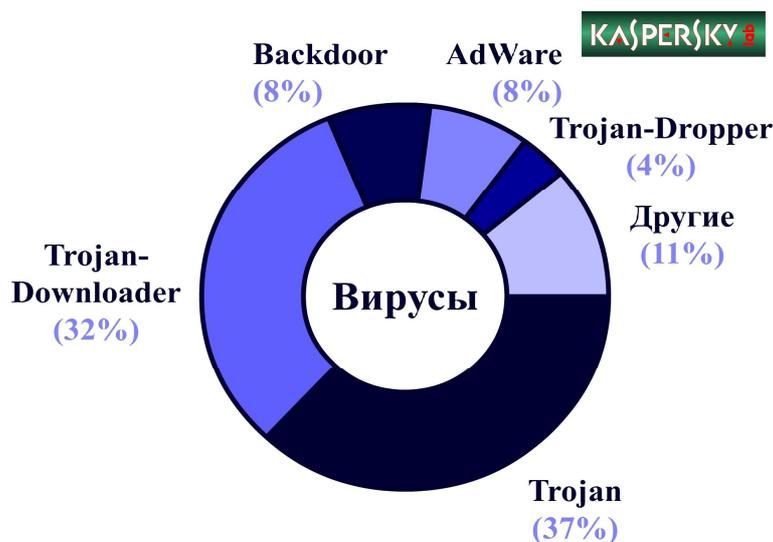


Рисунок 5.17 – Вирусная активность

По данным «Лаборатории Касперского», которая ведет мониторинг вирусной активности (www.kaspersky.com/viruswatch3), за ноябрь 2010 года больше всего было зарегистрировано троянских вредоносных программ trojan (37%), на втором месте – trojan-downloader (32%), на третьем месте – backdoor и adware (по 8%), далее trojan-dropper (4%). На долю всех остальных вирусов приходится 10% всех новых вредоносных программ (рисунок 5.17). Нужно отметить, что эта информация постоянно меняется, так как ежедневно появляются новые вирусы различных классов, которые по-разному себя проявляют.

Необходимо упомянуть еще о нескольких информационных процессах, которые сами по себе не являются вирусами, однако могут иметь вредоносные последствия не только для компьютера, но и для финансового состояния его пользователя, это спам и фишинг.

Спамом называют массовую рассылку электронной почты для получения выгоды на адреса пользователей, которые не выражали желания ее получать. С одной стороны, спам мешает работе пользователей, так как заставляет тратить время на обработку бесполезной информации, с другой стороны, спам нагружает каналы связи и сетевое оборудование провайдеров, что увеличивает трафик и снижает пропускную способность передачи полезной информации. Многие почтовые серверы и антивирусные программы поддерживают автоматическую фильтрацию большинства подобных электронных писем.

Фишингом называют вид мошенничества в глобальной сети Интернет с целью получения персональных данных пользователей. В данном случае пользователь получает сообщение о том, что ему необходимо обновить конфиденциальную информацию, перейдя по предложенной ссылке. Зайдя на поддельный сайт, пользователь оставляет там свои персональные данные, что может привести к краже его паролей, номеров кредитных карт, банковских счетов. Современные антивирусы собирают базу данных о таких угрозах и, при попытке пользователя перейти по фишинговой ссылке, предупреждают его об опасности.

5.8.2 Методы обеспечения информационной безопасности

Существует три основных метода защиты от вирусов, которые заключаются в предотвращении поступления компьютерных вирусов, в предотвращении вирусной атаки, если вирус проник в компьютер, и в предотвращении разрушительных последствий, если атака уже произошла (рисунок 5.18).



Рисунок 5.18 – Методы защиты от вредоносных программ

Реализовать эти методы можно с помощью использования программных, аппаратных или организационных средств. Для этого устанавливаются антивирусные программы, сканируются системные области операционной системы и полностью жесткий диск, создаются резервные копии дисков, разграничиваются права доступа к информации, отслеживается поступление информации по

различным портам ввода и вывода информации, производится обучение пользователей работе на компьютере.

Для обнаружения вирусов современные антивирусные программы используют целый набор различных технологий, которые можно разделить на несколько групп: поиск вирусов по сигнатурам, проведение проактивной защиты, применение эвристических методов поиска вредоносных программ (рисунок 5.19).



Рисунок 5.19 – Технологии работы антивирусных программ

Поиск по маске основан на сравнении кода исследуемой программы с **сигнатурами**, которые представляют собой часть вирусного кода. Если программа содержит этот код, то она инфицирована. В случае неполного совпадения сигнатуры с анализируемым кодом антивирус может отметить файлы как подозрительные. Для того чтобы обеспечить надежную работу компьютера, необходимо постоянно пополнять антивирусную программу новыми сигнатурами.

Обеспечение проактивной защиты основано на изучении поведения программы после ее запуска, при этом отслеживаются подозрительные действия, которые выполняет эта программа, например, изменение системных файлов или подключение к глобальной сети Интернет.

Эвристические методы нацелены на исследование файла, который не опознается сигнатурным сканером в качестве подозрительного или вредоносного. Их задача состоит в обнаружении еще не известных вредоносных программ. Принцип их работы может быть основан на поиске характерных для вредоносной программы фрагментов кода. Однако наиболее перспективной является технология применения эмуляторов при поиске вирусов. При этом происходит моделирование работы выполнения исследуемой программы в среде операционной системы. На практике эти два метода применяются одновременно для мониторинга появления вредоносных программ, в этом случае возможности эмулятора дополняются поиском по характерным сигнатурам.

Антивирусные программы могут работать в режиме монитора, когда происходит постоянный контроль над работой компьютера, или в режиме сканера, когда осуществляется проверка файлов на диске по определенной команде.

5.8.3 Антивирусная программа Kaspersky Internet Security 2011

Антивирусная программа Kaspersky Internet Security 2011 является одним из самых мощных инструментов поиска и нейтрализации вредоносных программ и угроз глобальной сети Интернет (рисунок 5.20). Антивирус Kaspersky Internet Security 2011 производит проверку файлов, сообщений, электронной почты и сайтов, обеспечивает защиту персональных данных, позволяет настроить доступ к объектам глобальной сети Интернет.

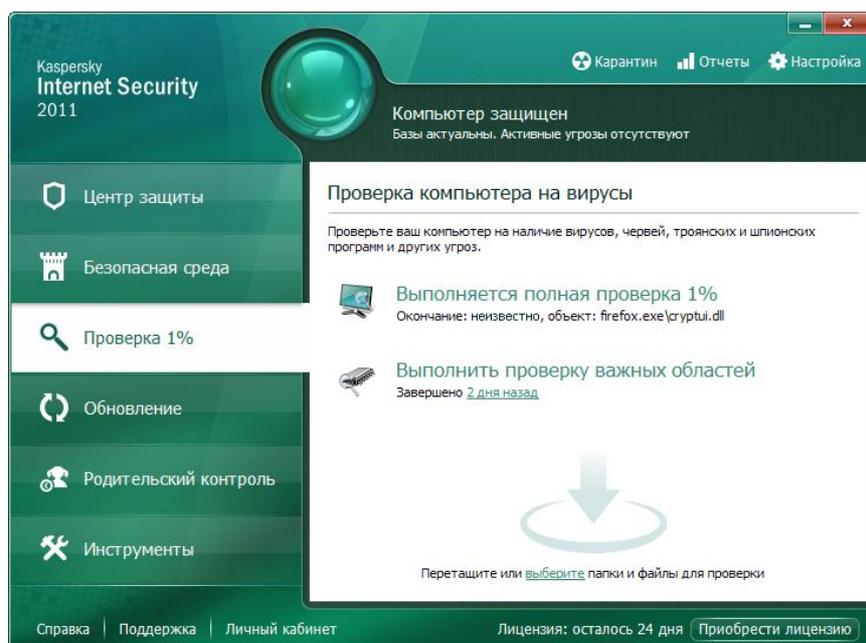


Рисунок 5.20 – Интерфейс программы Kaspersky Internet Security 2011

Программа Kaspersky Internet Security 2011 производит проактивную защиту от вирусов в режиме реального времени, отфильтровывает содержание сайтов, тестирует операционную систему на наличие уязвимостей, регулярно обновляет базу и программные модули. В данном антивирусе реализована виртуальная клавиатура для надежной защиты личных данных, сетевой экран для защиты от хакерских атак, интеллектуальная система защиты от спама и баннеров, родительский контроль для ограничения доступа к ресурсам компьютера и глобальной сети Интернет различным пользователям.

Тезаурус

Хост, ресурс, протокол, локальная сеть, глобальная сеть, политика сети, администрирование, топология, линия передачи данных, витая пара, коаксиальный кабель, оптоволокно, трансивер, точка доступа, ретранслятор, сетевая карта, концентратор, повторитель, коммутатор, маршрутизатор, мост, шлюз, терминатор, сетевая архитектура, коллизия, маркер, домен, служба, браузер, вирус, троянская программа, спам, фишинг, антивирус, сигнатура.

Контрольные вопросы

1. В чем состоит отличие топологии с пассивной звездой от топологии с активной звездой?
2. Укажите основные характеристики толстого коаксиального кабеля и его преимущества перед витой парой.
3. В каких случаях в качестве линии передачи данных используют одномодовый оптоволоконный кабель?
4. В чем заключается функциональная особенность работы в компьютерной сети маршрутизатора?
5. Какие функции выполняются в сети на канальном уровне эталонной модели OSI?
6. Перечислите основные достоинства и недостатки сетевой архитектуры Ethernet.
7. Назовите принципы работы протокола TCP/IP, используемого для передачи данных в глобальной сети Интернет.
8. Почему сложно обнаружить вирусы, основанные на применении алгоритмов полиморфичности?
9. Какие существуют разновидности и как распространяются вредоносные троянские программы?
10. На каких принципах основаны эвристические методы поиска вредоносных программ?

Список рекомендуемой литературы

1. Информатика : базовый курс [Текст] / под ред. С. В. Симоновича. – СПб . : Питер, 2009. – 640 с.
2. Куроуз, Дж. Компьютерные сети [Текст] / Дж. Куроуз, К. Росс. – СПб. : Питер, 2004. – 765 с.
3. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы [Текст] / В. Г. Олифер, Н. А. Олифер. – СПб . : Питер, 2010. – 944 с.
4. Соболев, Б. В. Информатика [Текст] / Б. В. Соболев, А. Б. Галин, Ю. В. Панов. – Ростов н/Д : Феникс, 2007. – 446 с.
5. Таненбаум, Э. Компьютерные сети [Текст] / Э. Таненбаум. – СПб. : Питер, 2010. – 992 с.
6. Шаньгин, В. Ф. Информационная безопасность компьютерных систем и сетей [Текст] / В. Ф. Шаньгин. – М . : ИНФРА-М, 2008. – 416 с.
7. Широкополосные беспроводные сети передачи информации [Текст] / В. М. Вишневецкий, А. И. Ляхов, С. Л. Портной, И. В. Шахнович. – М. : Техносфера, 2005. – 592 с.

ГЛАВА 6 АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

В настоящей главе рассматриваются базовые вопросы алгоритмизации и программирования, описаны основные понятия структурного и объектно-ориентированного программирования, приведена классификация современных языков программирования и их описание, рассмотрены алгоритмические конструкции и операторы на примере языка программирования Python.

6.1 Понятие программы и алгоритма

Одним из первых осознал последствия массового применения компьютеров в жизнедеятельности человека выдающийся ученый, создатель советской школы программирования А. П. Ершов. Именно он высказал мысль, что «программирование – вторая грамотность», и предложил изучать информатику и программирование в школах и университетах.

Сегодня нельзя представить современного оптотехника, энергетика, строителя, химика или экономиста, не знающего основ программирования, потому что высококвалифицированный специалист постоянно сталкивается с задачами автоматизации расчетов, оценки моделей или оптимизации процессов, которые необходимо решать с помощью языков программирования.

Одними из основополагающих понятий теории программирования являются термины «программа» и «алгоритм».

Программой называют логически упорядоченную последовательность команд, необходимых для управления компьютером. Однако в процессе управления компьютером необходимо решать самые разнообразные задачи, решение которых строится по определенному алгоритму.



Рисунок 6.1 – Свойства алгоритма

Алгоритмом называют формальное описание способа решения задачи в виде конечной последовательности действий. При решении любой задачи сначала строится алгоритм и только потом на его основе создается программа с помощью языка программирования.

Любой алгоритм должен обладать такими свойствами, как дискретность, массовость, детерминированность, результативность и формальность (рисунок 6.1). Рассмотрим каждое из этих свойств более подробно.

Свойство **дискретности** алгоритма говорит о том, что он состоит из неделимых, атомарных действий, причем в рамках решения задачи алгоритм переходит от одного устойчивого состояния в другое.

Свойство **массовости** алгоритма означает его применимость к решению любой задачи данного класса при любых начальных условиях.

Свойство **детерминированности** алгоритма заключается в том, что переход алгоритма от одного устойчивого состояния в другое должен быть строго определен.

Свойство **формальности** алгоритма определяет получение конечного результата при выполнении алгоритма в независимости от внешних условий.

Свойство **результативности** алгоритма означает то, что алгоритм должен получить результат за конечное число действий.

6.2 Блок-схема алгоритма

Описать алгоритм можно несколькими способами, например, раскрыть каждый этап решения задачи на естественном языке. Такой способ имеет несомненное достоинство: любой человек, прочитав этот алгоритм, поймет его смысл. Однако существенным недостатком словесного описания алгоритма является отсутствие строгих правил построения алгоритмов, в данном случае нет единого подхода в его оформлении.

Для формализации этой задачи используются блок-схемы. **Блок-схема** представляет собой описание алгоритма с помощью символов, имеющих заданное значение, соединяющих их линий и пояснительного текста. Этот способ универсальный, наглядный и явно отображает порядок выполнения отдельных действий. Для обозначения каждого действия используется свой символ. Рассмотрим значение этих символов более подробно.

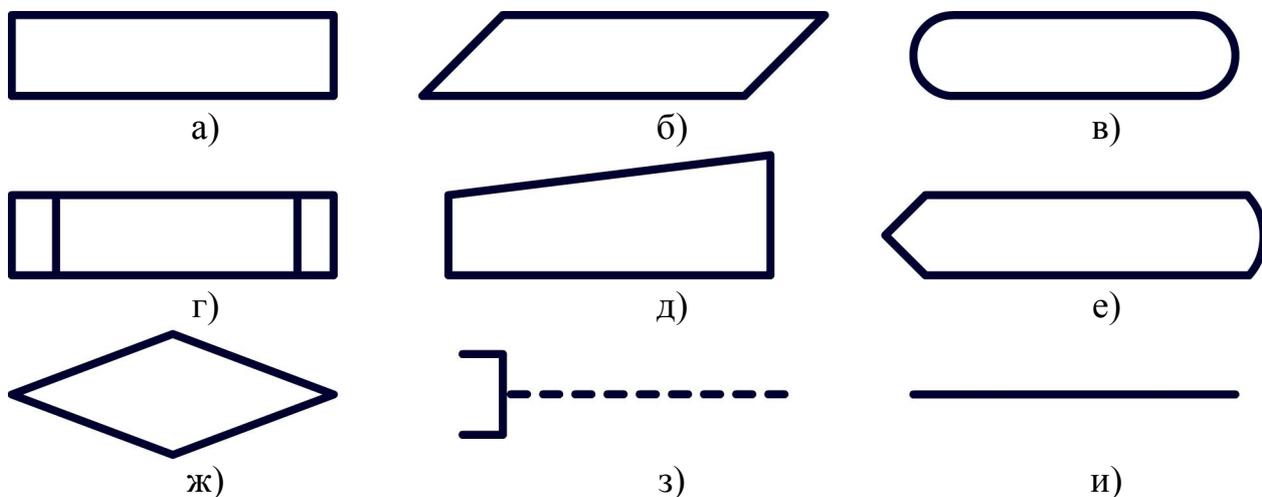


Рисунок 6.2 – Описание символов блок-схем

Для обозначения обработки данных любого вида, в том числе выполнение операций, приводящих к изменению значения, формы или размещения данных, используется символ, приведенный на рисунке 6.2, а.

Для обозначения ввода или вывода данных на любой носитель используется символ, приведенный на рисунке 6.2, б.

Для обозначения выхода во внешнюю среду или входа из внешней среды, в том числе начало или конец блок-схемы алгоритма, используется символ, приведенный на рисунке 6.2, в.

Для обозначения предопределенного процесса, состоящего из одной или нескольких операций, которые определены в другом месте, например, в подпрограмме или модуле, используется символ, приведенный на рисунке 6.2, г.

Для обозначения данных, ввод которых осуществляется вручную с устройств любого типа, например, клавиатуры или светового пера, используется символ, приведенный на рисунке 6.2, д.

Для обозначения данных, представленных на экране монитора, используется символ, приведенный на рисунке 6.2, е.

Для обозначения решения переключательного типа, имеющего один вход и несколько альтернативных выходов, только один из которых может быть активизирован после вычисления заданных условий, используется символ, приведенный на рисунке 6.2, ж.

Для обозначения описательных комментариев или пояснительных записей используется символ, приведенный на рисунке 6.2, з.

Для обозначения потока данных используется символ, приведенный на рисунке 6.2, и. При необходимости для повышения удобочитаемости могут быть добавлены указатели в виде стрелок.

Все представленные символы используются для построения основных алгоритмических конструкций.

6.3 Основные алгоритмические конструкции

Различают четыре основные алгоритмические конструкции: линейный, ветвящийся, циклический и рекурсивный алгоритм (рисунок 6.3). Эти конструкции по отдельности и в комбинации применяются для построения сложных алгоритмов. Рассмотрим каждую из этих конструкций более подробно.



Рисунок 6.3 – Виды основных алгоритмических конструкций

Линейный алгоритм представляет собой последовательность действий, которые выполняются строго друг за другом и только один раз. Такая алгоритмическая конструкция довольно проста, очень надежна и стабильна.

Ветвящийся алгоритм является разновидностью линейной алгоритмической конструкции, которая обеспечивает выбор между двумя альтернативными вариантами в зависимости от значений входных данных. Эта конструкция позволяет алгоритму реагировать на изменяющиеся данные и адаптироваться к ним.

Циклический алгоритм представляет собой последовательность действий, которые могут выполняться несколько раз в зависимости от значений входных данных. Эта конструкция применяется очень часто при обработке набора однотипных данных.

Рекурсивный алгоритм включает в себя последовательность действий, в процессе выполнения которых он обращается сам к себе. На практике встречается не так много задач, требующих применения этой алгоритмической конструкции, однако, если это необходимо, то рекурсивная алгоритмическая конструкция делает алгоритм более компактным и гибким.

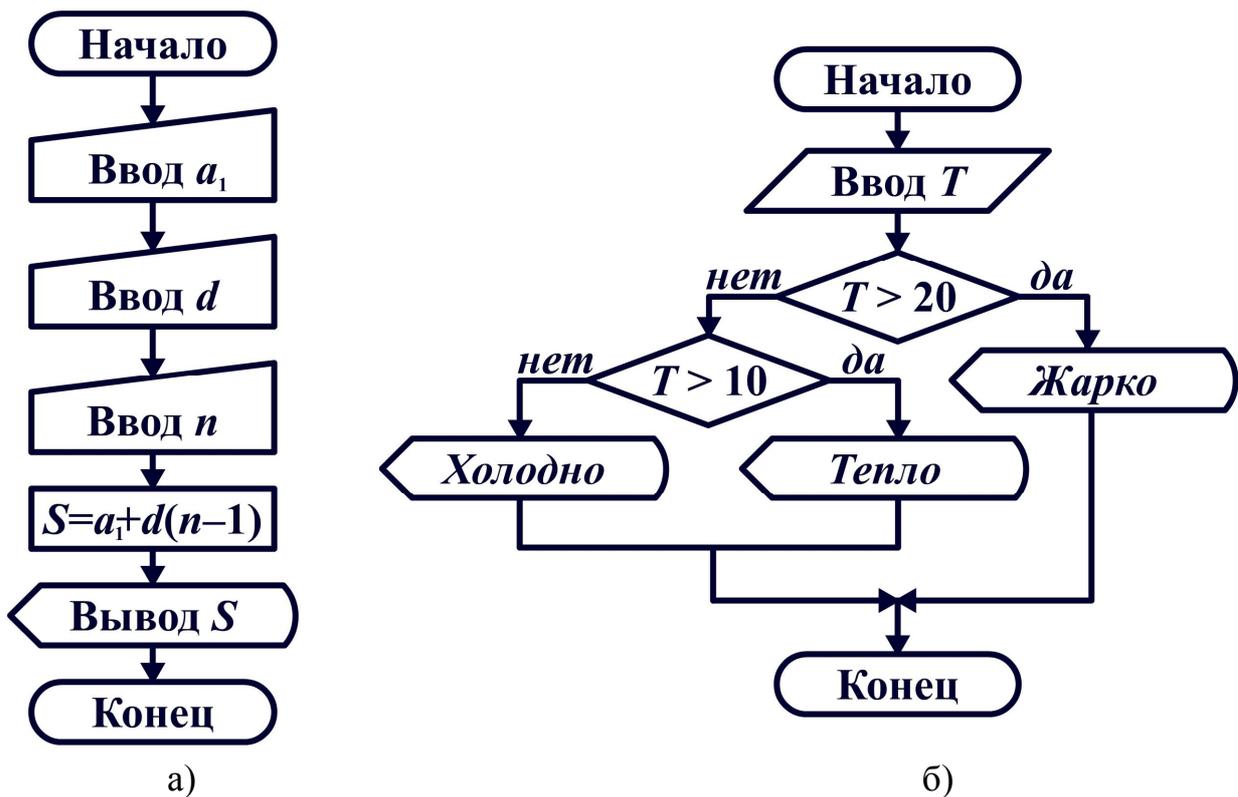


Рисунок 6.4 – Основные алгоритмические конструкции:
а) линейный алгоритм, б) ветвящийся алгоритм

Рассмотрим применение наиболее простых линейных и ветвящихся алгоритмических конструкций на конкретных примерах.

При использовании линейной алгоритмической конструкции (рисунок 6.4, а) сначала происходит ввод с клавиатуры первого члена арифметической прогрессии a_1 , ее разности d и количества членов арифметической прогрессии n . На основании введенных данных по формуле вычисляется сумма арифметической прогрессии S , значение которой выводится на экран монитора.

При использовании ветвящейся алгоритмической конструкции (рисунок 6.4, б) осуществляется ввод значения температуры T . Если значение температуры T больше 20, то на экран монитора выводится текст «Жарко», иначе – выполняется еще одно условие. Если значение температуры T больше 10, то на экран монитора выводится текст «Тепло», иначе – выводится текст «Холодно».

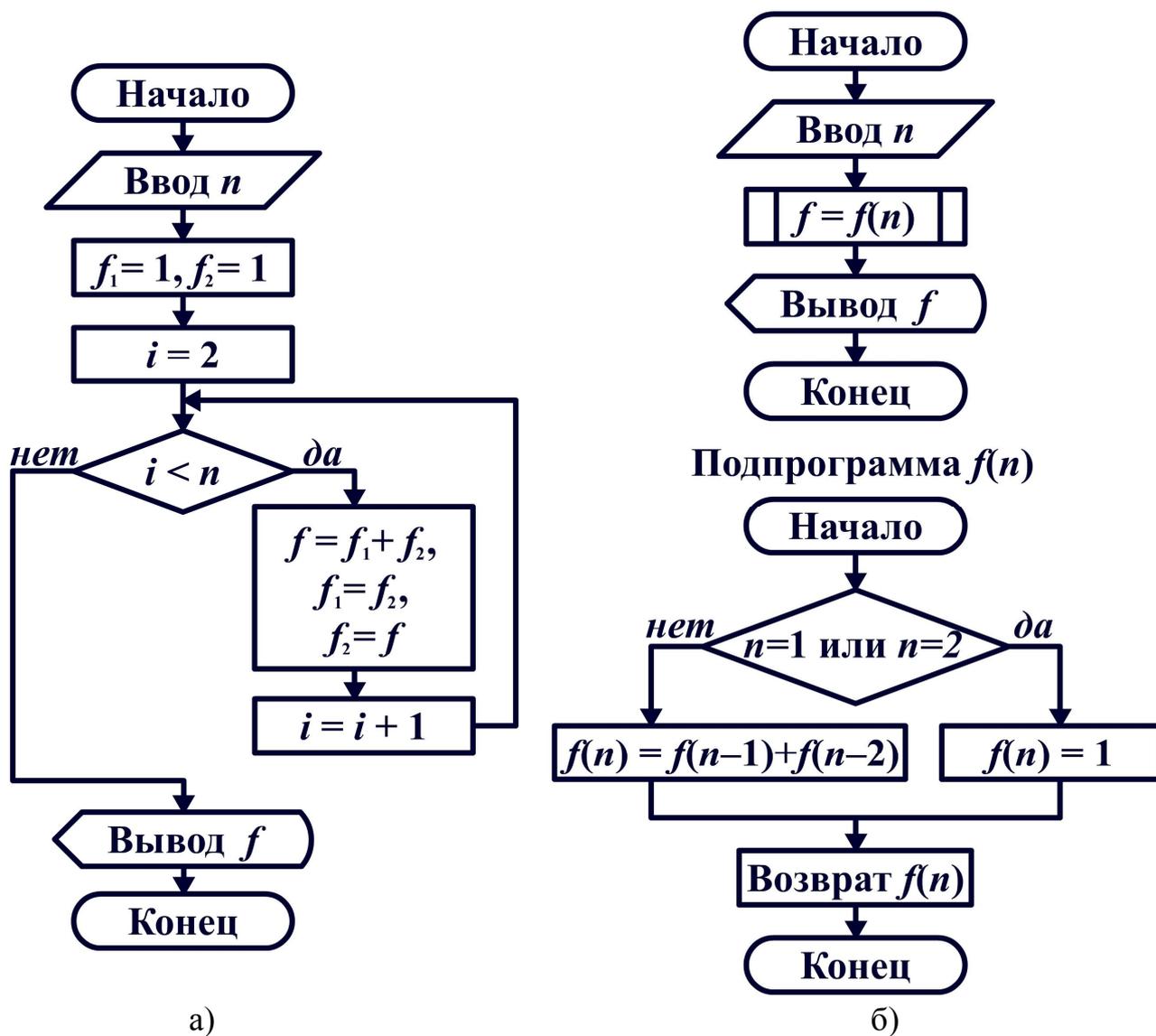


Рисунок 6.5 – Основные алгоритмические конструкции:
 а) циклический алгоритм, б) рекурсивный алгоритм

Рассмотрим принципы использования циклического и рекурсивного алгоритмов на примере вычисления чисел Фибоначчи, представляющих собой

ряд чисел, в котором каждое последующее число равно сумме двух предыдущих: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 и т. д.

При использовании циклической алгоритмической конструкции (рисунок 6.5, а) сначала происходит ввод номера определяемого элемента n , затем двум переменным f_1 и f_2 присваиваются единичные значения. Далее выполняется циклическое выполнение действий $(n - 2)$ раз, при которых переменной f присваивается сумма значений переменных f_1 и f_2 . После чего происходит изменение значений переменных: значению переменной f_1 присваивается значение f_2 , значению переменной f_2 , присваивается значение f . После работы цикла переменная f содержит значение определяемого числа Фибоначчи, которое выводится на экран монитора.

При использовании рекурсивной алгоритмической конструкции (рисунок 6.5, б) количество действий алгоритма значительно сократится. В основной программе также происходит ввод номера определяемого элемента n , но затем вызывается подпрограмма, возвращающая значение определяемого числа Фибоначчи, которое выводится на экран монитора. В подпрограмму передается лишь значение n , при этом происходит проверка значения n , с помощью ветвящейся алгоритмической конструкции. Если значение n равно единице или двойке, то подпрограмма возвращает значение, равное единице, иначе происходит вызов этой же подпрограммы, но уже с аргументами $(n - 1)$ и $(n - 2)$, результат выполнения этих подпрограмм складывается и возвращается в вызывающую ветку алгоритма. Такая вложенная структура позволяет получить значение определяемого числа Фибоначчи, равного сумме двух предыдущих элементов.

6.4 Классификация языков программирования

После разработки алгоритма необходимо создать на его основе программу с помощью языка программирования.

Язык программирования представляет собой искусственный формализованный язык, предназначенный для описания программ и алгоритмов и решения задач на компьютере.

У каждого языка программирования имеется синтаксис и семантика. **Синтаксисом** языка программирования называют совокупность требований, предъявляемых к языку, **семантикой** – смысл каждой команды и других конструкций языка. Так как языки программирования являются искусственными, то в них синтаксис и семантика строго определены и не допускают свободного толкования выражения, что характерно для естественного языка.



Рисунок 6.6 – Деление языков программирования

6.4.1 Алгоритмическое программирование

При использовании **алгоритмического программирования** программа непосредственно состоит из последовательности команд, выполняющих действия, определенные в алгоритме. При построении программ с помощью такого подхода используются простейшие операторы: присваивание и условный оператор. Этот подход в настоящее время в чистом виде практически не применяется, так как при увеличении текста программы разобраться в правильности работы невозможно.

6.4.2 Процедурное программирование

Для решения вышеописанной проблемы используется **процедурное программирование**. Целью этого подхода является создание программы, состоящей из последовательности подпрограмм, каждая из которых выполняет одно из действий, предусмотренных алгоритмом. Комбинируя эти подпрограммы, можно формировать итоговый алгоритм уже не из простых операторов, а из законченных блоков кода, имеющих определенную смысловую нагрузку, причем можно многократно обращаться к таким подпрограммам по их названиям.

Выделяют два типа подпрограмм. **Процедура** содержит группу операторов, выполняющих одно из действий алгоритма в зависимости от входных данных. **Функция** дополнительно вычисляет некоторое значение и передает его обратно в главную программу.

Со временем сложность задач возрастала, что приводило к увеличению размера программного кода. Использование процедурного программирования не только не позволяло повысить производительность программного обеспечения, но и значительно усложняло процесс его разработки.

6.4.3 Объектно-ориентированное программирование

Обойти эти недостатки стало возможным с использованием объектно-ориентированного подхода. **Объектно-ориентированное программирование** основано на понятии **объекта**, представляющего собой совокупность атрибутов, методов их обработки и событий, на которые объект может реагировать.

Объекты могут иметь идентичную структуру и отличаться только значениями атрибутов. В таком случае в программе создается новый тип, который называют **классом**, а каждый конкретный объект, имеющий структуру этого класса, называют **экземпляром класса**. Выделяют следующие основные понятия объектно-ориентированного программирования (рисунок 6.8).

Полиморфизм означает свойство переопределять методы наследуемого класса и корректно их использовать в различных контекстах.

Инкапсуляция означает свойство скрывать внутреннюю структуру класса, в которой описываются его методы и атрибуты, значения которых в промежутках между вызовами методов класса могут меняться.

Наследование означает возможность создавать новые классы на основе существующих с наследованием всех их свойств и методов и добавлением собственных. Класс, не имеющий предшественника, называют базовым.



Рисунок 6.8 – Основные понятия объектно-ориентированного программирования

Во многих средах объектно-ориентированного программирования используется визуальный подход к созданию программ. При этом многие стандартные операции выполняются не путем ручного программирования, а с помощью готовых визуальных компонентов. Их свойства и поведение настраиваются с помощью редакторов, визуально показывающих характеристики соответствующих элементов. При этом исходный текст программы генерируется автоматически.

6.5 Обзор современных языков программирования

Компания TIOBE Software (<http://www.tiobe.com>) ежемесячно публикует рейтинг языков программирования, который строится по данным популярных поисковых систем о количестве разработчиков и компаний, использующих язык, а также о количестве обучающих курсов по нему.

Несмотря на бурные дискуссии относительно качества и достоверности этого рейтинга, его значимость велика, так как позволяет прогнозировать тенденции развития языков программирования и грамотно выбирать средства реализации долгосрочных программных проектов.

Как видно из рисунка 6.9, на первом месте в данном рейтинге находится язык программирования Java, однако видно, что с каждым годом он неуклонно теряет свои позиции и в скором времени будет потеснен языком C, имеющим более стабильное положение. Также имеет тенденции к снижению популярности язык программирования C++, рейтинг которого сравним с рейтингом PHP, активно применяемого в программировании приложений для глобальной сети Интернет. Язык программирования Visual Basic тоже за последние годы неуклонно сдает свои позиции, хотя и находится в пятерке самых популярных языков. Наиболее динамично развиваются языки программирования C# и Python, которые за последний год обошли в данном рейтинге язык Perl. Язык Delphi имеет довольно стабильные позиции, так как требуется обслуживать ранее разработанное программное обеспечение, хотя никаких предпосылок для увеличе-

ния рейтинга нет, так как малое количество современных программ разрабатываются с его помощью. Стремительными темпами развивается язык программирования Objective-C, что в основном связано с ростом популярности мобильных платформ, приложения для которых разрабатываются именно на этом языке. Дадим краткую характеристику каждому из этих популярных языков программирования.

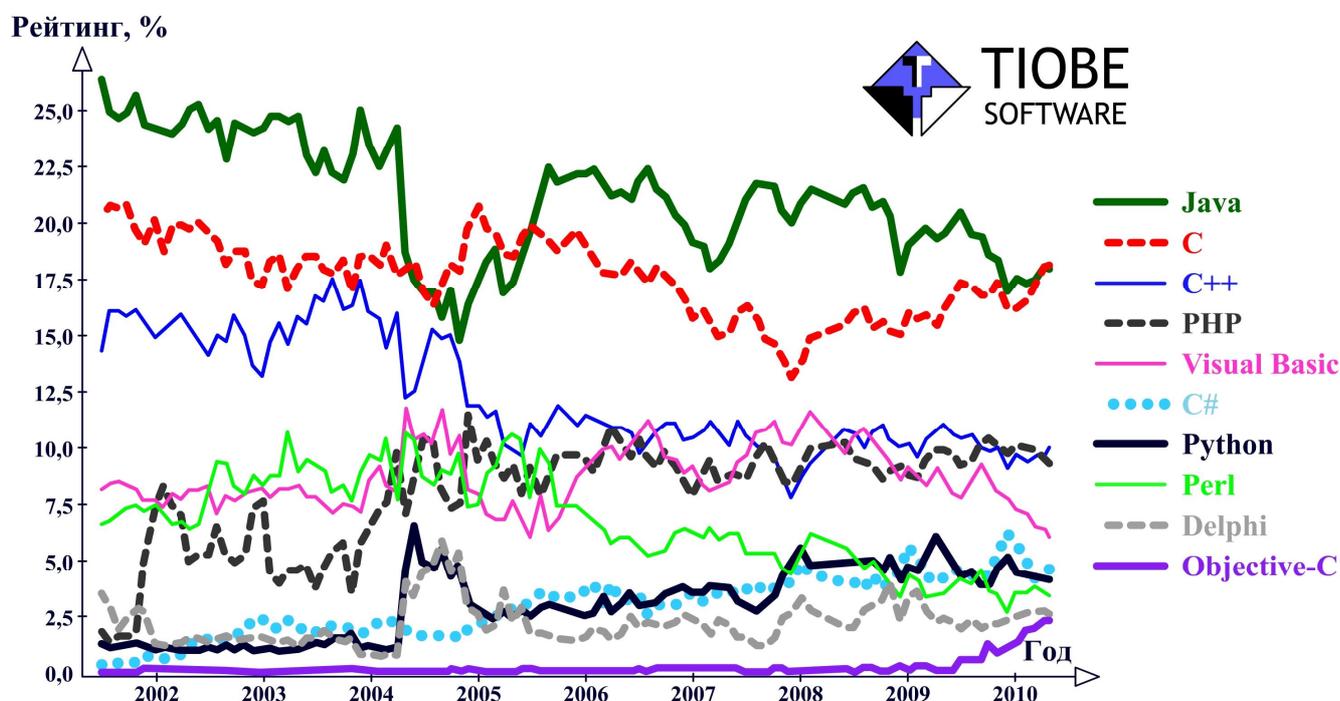


Рисунок 6.9 – Рейтинг языков программирования

Язык программирования C является процедурным языком, который был создан для высокоэффективной работы с операционной системой UNIX. Хорошо зарекомендовав себя, этот язык был перенесен и на другие платформы и стал одним из самых популярных языков программирования в мире. С помощью этого языка создается большинство операционных систем, системного программного обеспечения и прикладных программ. Для языка программирования C характерны лаконичность, современный набор конструкций управления потоком выполнения, структур данных и большой набор операций. Многие языки взяли за основу синтаксис языка C.

Язык программирования C++ является объектно-ориентированным расширением языка C и сочетает в себе свойства как высокоуровневых, так и низкоуровневых языков программирования, что позволяет значительно повысить производительность созданных с помощью него программ, таких, как драйверы устройств, приложения для встраиваемых систем, высокопроизводительные серверы.

Язык программирования C# также относится к семье объектно-ориентированных языков программирования с C-подобным синтаксисом. Дан-

ный язык является основным инструментом разработки приложений для платформы Microsoft.NET. Этот факт predetermined выбор программистов, работающих над созданием проектов для платформы Microsoft.NET.

Язык программирования Java был создан для того, чтобы упростить разработку приложений на основе C++ путем исключения из него всех низкоуровневых возможностей. Главной особенностью этого языка является компиляция программы не в машинный код, а в платформно-независимый байт-код, который может выполняться с помощью специализированного интерпретатора – Java Virtual Machine. Высокую популярность этого языка среди разработчиков обеспечили два преимущества: поддержка всевозможных мобильных устройств и микрокомпьютеров и возможность создания платформно-независимых программных модулей, способных работать на серверах в глобальных и локальных сетях с различными операционными системами.

Язык программирования Objective-C появился благодаря необходимости создавать программы из готовых компонентов. При этом новый язык должен быть прост в освоении и основан на языке C. Основная идея корпорации Apple, которая и инициировала его разработку, заключается в упрощении создания программ на основе использования стандартных компонентов. На сегодняшний день **Objective-C** применяется при создании приложений для мобильных платформ корпорации Apple.

Язык программирования PHP представляет собой скриптовый язык программирования, используемый для создания Интернет-сайтов. В настоящее время этот язык поддерживается практически на всех серверах в глобальной сети Интернет. Он является лидером среди языков программирования, применяющихся для создания динамических сайтов. Это связано с тем, что в PHP реализовано множество встроенных средств для разработки Интернет-сайтов: взаимодействие с базами данных, работа с удаленными файлами, поддержка сессий и авторизации, обработка переменных окружения сервера.

Язык программирования Perl является высокоуровневым динамическим языком, который ориентирован на обработку больших текстовых файлов и генерацию текстовых отчетов. Данный язык программирования нашел широкое применение у разработчиков Интернет-сайтов в связи с тем, что статический сайт является текстовым документом. Следует отметить, что в последнее время Perl значительно сдает позиции.

Язык программирования Visual Basic сочетает в себе процедурный и объектно-ориентированный подход к программированию, разработанный корпорацией Microsoft. На сегодняшний день очень популярен диалект этого языка программирования Visual Basic for Application, представляющий собой общую языковую платформу для приложений Microsoft Office, что позволяет повысить производительность выполнения многих команд в этих приложениях. Достоинством языка Visual Basic for Application является то, что среда разработки включает инструменты для визуального конструирования программ.

Язык программирования Delphi разработан на основе языка Pascal, который первоначально предназначался для обучения студентов основным прие-

мам программирования. Этот язык программирования дает возможность эффективно решать не только учебные, но и прикладные задачи, поэтому стал использоваться многими программистами. Отличительной особенностью Delphi является хороший стиль программирования, благодаря которому возможно разрабатывать надежные, легко проверяемые программы с ясной и четкой структурой. Однако на сегодня Delphi практически не используется разработчиками программного обеспечения, так как значительно проигрывает по функциональным возможностям С-подобным языкам программирования.

Язык программирования Python ориентирован на производительность создания программ и читаемость кода. Функциональные возможности этого языка очень велики, так как стандартная библиотека включает большое количество полезных функций, и сторонние производители дополнительно разрабатывают модули расширений данного языка программирования. Поэтому с каждым годом увеличивается количество программистов, отдающих предпочтение при создании проектов именно языку программирования Python.

6.6 Введение в язык программирования Python

Изучение основ современного программирования будем осуществлять с помощью одного из популярнейших языков программирования Python, так как он имеет ряд значительных преимуществ перед конкурентами.

Во-первых, язык программирования Python позволяет создавать удобочитаемый и ясный программный код, что, в свою очередь, повышает производительность работы программного обеспечения и уменьшает количество времени на отладку кода. Во-вторых, большая часть разработанных программ на языке Python может выполняться без изменений на всех основных платформах. В-третьих, в состав Python входит большое число стандартных библиотек, позволяющих разработчику создавать программное обеспечение практически в любой предметной области.

Язык программирования Python используется многими лидерами в области информационных технологий. Например, компания Google применяет Python в своей поисковой системе; сервис YouTube – в службе коллективного использования видеоматериалов; компании Intel, Cisco, Hewlett-Packard и IBM – для тестирования аппаратного обеспечения; фирма Pixar – в производстве анимационных фильмов; NASA – для научных вычислений.

Для того чтобы использовать мощные возможности этого языка программирования, необходимо скачать установочный пакет с графическим интерфейсом по адресу www.python.org/download. После установки дистрибутива можно приступить к изучению языка программирования Python.

Программы, написанные на языке программирования Python, представляют собой текстовые файлы с расширением *.py и могут разрабатываться с помощью простого текстового редактора. Для более удобной работы можно использовать среду программирования IDLE, которая позволяет вводить программный код на языке Python с получением результатов прямо в командной

оболочке, предоставляет редактор программного кода с подсветкой синтаксиса языка Python и тестировать программный код в режиме пошагового выполнения.

6.6.1 Типы данных Python

Язык Python предоставляет несколько встроенных типов данных. Для представления целых чисел, используется тип `int`, для представления чисел с плавающей запятой – `float`. Размер чисел ограничивается только объемом памяти. Над значениями этих типов можно выполнять все известные арифметические операции, такие, как сложение, вычитание, умножение и деление.

```
-525
112348456879832451098345984569
2.71
3.7e15
```

Для представления строк используется тип `str`. Строковые значения заключаются в двойные кавычки (" "). Для доступа к элементам строк в последовательностях используются квадратные скобки ([]).

```
"Язык программирования Python"
"Улица Ленина дом 46"
```

В языке программирования Python типы данных являются неизменяемыми, установив значение, его уже нельзя будет изменить. Для преобразования значений из одного типа в другой можно использовать конструкции `int()`, `float()` и `str()`.

6.6.2 Ссылки на объекты Python

В языке Python нет переменных, вместо них используются ссылки на объекты. При работе со значениями интерпретатор Python сначала создает объект соответствующего типа, а затем создает ссылку на этот объект, при этом оператор (`=`) связывает ссылку со значением, что практически эквивалентно присваиванию:

```
N = 256
S = "Программирование"
```

Имена, используемые для идентификации ссылок, должны начинаться либо с латинской буквы, либо со знака подчеркивания, остальные символы могут быть либо латинскими буквами, либо цифрами, либо дополнительными знаками, кроме определенных конструкций языка.

Для вывода на экран значений объектов можно использовать функцию `print()`, для ввода значений – `input()`.

```
x = input("Введите x = ")
print("Значение x равно ", x)
```

6.6.3 Коллекции данных Python

В языке программирования Python имеется несколько типов коллекций, способных хранить произвольное число элементов данных любых типов. Рассмотрим два типа коллекций: кортежи и списки.

Кортежи являются неизменяемыми объектами, поэтому после создания кортеж нельзя изменить. В списках, в отличие от кортежей, можно изменять, добавлять или удалять элементы. Кортежи создаются следующим образом:

```
("один", "два", "три", "четыре", "пять")  
("Пн", "Вт", "Ср", "Чт", "Пт", "Сб", "Вс")
```

Списки заключаются в квадратные скобки, элементы списка также отделяются друг от друга запятыми:

```
[1, 1, 2, 3, 5, 8, 13, 21]  
["Иванов", "Петров", "Сидоров"]
```

Для того чтобы получить количество элементов кортежа или списка, можно использовать функцию `len()`, аргументом которой является интересующий кортеж или список. Для кортежа или списка можно создать именованную ссылку и работать с ней как с объектом, обладающим своими методами, например, вставка дополнительного элемента в список осуществляется следующим образом:

```
rupil = ["Иванов", "Петров", "Сидоров"]  
rupil.append("Давыдов")
```

Также можно удалять элементы с помощью метода `remove()`, вставлять элементы в заданную позицию с помощью метода `insert()`.

Получить доступ к отдельным элементам кортежей или списков можно с использованием оператора квадратных скобок:

```
num = [1, 2, 3, 4, 5]  
num[4] = num[0] + num[1] + num[2] + num[3]  
print(num[4])
```

После выполнения этого программного кода на экран будет выведено значение последнего элемента списка `num` – 10.

6.6.4 Инструкции управления Python

Инструкции Python выполняются в порядке строгой очередности. Однако порядок выполнения инструкций может изменяться с помощью условных операторов и операторов цикла, которые были введены еще Чарльзом Бэббиджем и Адой Лавлейс. Рассмотрим синтаксис условного оператора `if`:

```
if логическое_выражение_1:  
    блок_программного_кода_1  
elif логическое_выражение_2:  
    блок_программного_кода_2  
else:  
    блок_программного_кода
```

Логическое выражение может принимать два значения: True и False. Логические выражения можно строить с помощью операторов сравнения: больше (>), меньше (<), больше либо равно (>=), меньше либо равно (<=), равно (==) и не равно (!=). Также можно применять оператор идентичности is, который устанавливает, не ссылаются ли ссылки на один объект. Дополнительно используется оператор членства in, который определяет, существует ли в коллекции данных заданное значение или нет.

Блок программного кода Python представляет собой последовательность одной или более инструкций. Причем особенностью этого языка программирования является то, что каждый блок программного кода выделяется отступом.

Приведем пример использования условного оператора:

```
temp = int(input("Введите температуру = "))
if temp > 20:
    print("жарко")
elif temp > 10:
    print("тепло")
else:
    print("холодно")
```

Как видно из примера, идентификатор temp ссылается на числовое значение, введенное пользователем. Затем происходит проверка введенного значения, если оно больше 20, то выводится на экран сообщение «жарко», если больше 10, но не превышает 20, то выводится сообщение «тепло», в противном случае – «холодно».

Рассмотрим синтаксис оператора цикла while:

```
while логическое_выражение_1:
    блок_программного_кода
```

Эта инструкция применяется для выполнения блока программного кода столько раз, пока логическое выражение принимает значение True. Внутри блока программного кода можно использовать операторы break и continue. Инструкция break прерывает выполнения цикла, инструкция continue передает управление в начала цикла.

Приведем пример использования оператора цикла:

```
n=1
sum=0
while n<=10:
    sum+=n
    n+=1
print("Сумма арифметической прогрессии равна ", sum)
```

Как видно из примера, создаются два значения 1 и 0, на которые ссылаются идентификаторы n и sum соответственно. После этого проверяется условие, если значение, на которое ссылается n, не превышает 10, то блок команд цикла выполняется. При этом каждый раз идентификатор sum будет ссылаться на значение, увеличенное по сравнению с предыдущим на значение, которое

связано с n . Далее идентификатор n будет ссылаться на значение, увеличенное по сравнению с предыдущим на 1. После того, как логическое выражение цикла вернет значение `False`, на экран выведется сообщение «Сумма арифметической прогрессии равна 55».

6.6.5 Использование функций Python

Язык программирования Python предоставляет возможность оформления блоков программного кода в виде функций, которые позволяют создавать программы более компактными. Рассмотрим синтаксис определения функции:

```
def имя_функции(аргументы):  
    блок_программного_кода
```

В качестве аргументов могут быть переданы один или несколько параметров, либо параметры могут отсутствовать. Функция возвращает значение с помощью инструкции `return`. Возвращаемое значение может быть единственным значением или кортежем возвращаемых значений.

Приведем пример использования функции:

```
def degree(a, n):  
    i=1  
    b=1  
    while i<=n:  
        b*=a  
        i+=1  
    return b  
  
i=1  
sum=0  
while i<=10:  
    sum+=degree(2, i)  
    i+=1  
print(sum)
```

Как видно из примера, создается функция `degree`, которая принимает в качестве параметров два числа: n – степень, в которую необходимо возвести основание a . Функция возвращает значение b , которое содержит n произведений числа a само на себя. Далее в цикле 10 раз вызывается разработанная функция `degree`, чтобы получить сумму степеней числа 2. После чего результат выполнения программы выводится на экран.

6.6.6 Объектно-ориентированная модель Python

Язык Python одновременно поддерживает различные стили программирования. Он позволяет программировать в процедурном стиле, объектно-ориентированном стиле и функциональном стиле, а также допускает смешивание стилей. На сегодняшний день при создании крупных программных продук-

тов предпочтение отдается объектно-ориентированному стилю программирования, рассмотрим его более подробно.

Синтаксис создания собственного класса выглядит следующим образом:

```
class имя_класса:  
    блок_программного_кода  
class имя_класса(имя_базового_класса):  
    блок_программного_кода
```

Единым базовым классом является класс `object`, если создаваемый класс является прямым наследником данного класса, то его имя указывать не нужно, иначе необходимо явно записывать название наследуемого класса. Именно таким образом в Python реализуется принцип наследования.

Синтаксис создания класса дает возможность создавать классы динамически, когда в этом возникнет необходимость. Методы класса создаются с помощью инструкции `def` внутри блока программного кода класса. Экземпляры класса создаются посредством обращения к имени класса.

При вызове метода интерпретатор автоматически передает ему первый аргумент – ссылку на сам объект. Этот параметр необходимо включать в список под именем `self`. Все атрибуты объекта должны квалифицироваться именно этим именем.

Следовательно, все атрибуты и реализация методов класса скрыты от пользователя, который работает с объектом как с единым целым. В этом заключается принцип инкапсуляции в Python.

Чтобы создать объект, необходимо выполнить два действия. Сначала необходимо создать неинициализированную заготовку объекта, а затем необходимо подготовить объект к использованию, инициализировав его. Когда создается объект, то сначала вызывается специальный метод `__new__()`, который создает объект, а затем выполняется инициализация объекта вызовом специального метода `__init__()`, который называют конструктором и используют для инициализации состояния объектов. Предопределенные специальные методы могут быть перегружены, если в этом есть необходимость. Для этого язык программирования Python предоставляет специальные обработчики, которые могут использоваться классами для перехвата и реализации любой встроенной операции.

Рассмотрим пример создания двух классов:

```
class Thin_Lens():  
    def __init__(self, n, R1, R2):  
        self.n = n  
        self.R1 = R1  
        self.R2 = R2  
    def D(self):  
        return (self.n - 1) * (1/self.R1 - 1/self.R2)  
    def F(self):  
        return (1/self.D())  
    def pr(self):
```

```
    if self.D()>0:
        return "собирающая"
    else:
        return "рассеивающая"
def __str__(self):
    x = "Оптическая сила D = %s\n" %(self.D())
    x+= "Фокусное расстояние F = %s\n" %(self.F())
    x+= "Преломляющие свойства - %s\n" %(self.pr())
    return x
def __add__(self, other):
    x = "Оптическая сила комбинации линз D = "
    return x + str(self.D() + other.D())

class Lens(Thin_Lens):
    def __init__(self, n, R1, R2, d):
        self.n = n
        self.R1 = R1
        self.R2 = R2
        self.d = d
    def D(self):
        x = (self.n-1)*self.d/(self.n*self.R1*self.R2)
        return (self.n-1)*(1/self.R1 - 1/self.R2 + x)

L1 = Thin_Lens(1.5, 0.2, 0.4)
print(L1)
L2 = Lens(2.0, 0.1, 0.15, 0.02)
print(L2)
print(L1 + L2)
```

Как видно из примера, сначала создается класс тонких линз `Thin_Lens`, который является прямым наследником класса `object`. В этом классе сначала перегружается метод инициализации класса `__init__(self, n, R1, R2)`, где устанавливаются начальные значения атрибутов: относительный показатель преломления n , радиусы кривизны линзы $R1$ и $R2$. Затем создается метод `D(self)`, необходимый для расчета оптической силы линзы по начальным значениям. Этот метод создается с помощью инструкции `def`, после чего происходит расчет значения оптической силы, которое возвращается с помощью инструкции `return`. Далее создается еще один метод `F(self)`, применяемый для нахождения фокусного расстояния линзы. Расчет фокусного расстояния осуществляется на основе вызова ранее описанного метода этого же класса `self.D()`, после чего полученное значение также возвращается с помощью инструкции `return`. После этого создается метод `pr(self)` для оценки преломляющих свойств линзы, который построен на применении условного оператора и вызове метода `self.D()`. В зависимости от значения оптической силы

линзы метод возвращает либо значение «собирающая», либо значение «рас-сеивающая».

Далее применяется перезагрузка метода `__str__()` и `__add__()`. Результатом работы метода `__str__(self)` является формирование текстовой строки, содержащей вызов всех методов данного класса. Полученный текст с параметрами линзы и пояснениями можно выводить на экран для восприятия пользователя. Метод `__add__(self, other)` вызывается тогда, когда над объектами выполняется операция сложения. В данном случае при сложении двух объектов класса тонких линз будут складываться их оптические силы, что соответствует законам оптики, и выведен комментарий.

После этого создается еще один класс `Lens(Thin_Lens)`, который является прямым наследником класса `Thin_Lens`, наследующий все его атрибуты и методы. Однако созданный класс линз должен учитывать толщину линзы `d`, поэтому метод инициализации класса `__init__(self, n, R1, R2, d)` необходимо перегрузить. Также нужно изменить метод расчета оптической силы данной линзы `self.D()` с учетом атрибута `d`. Остальные методы останутся неизменными.

В данном случае можно видеть воплощение полиморфизма в Python, означающего, что смысл операции зависит от объекта, над которым она выполняется. То есть расчет оптической силы линзы вызывается одним и тем же методом `self.D()`, однако результат будет зависеть от того, к какому классу относится объект.

В результате выполнения программы будет создан объект `L1` класса `Thin_Lens` с относительным показателем преломления $n = 1.5$, радиусами кривизны линзы $R1 = 0.4$ и $R2 = 0.4$. Затем информация об этом объекте будет выведена на экран. После этого будет создан объект `L2` класса `Lens` с относительным показателем преломления $n = 2.0$, радиусами кривизны линзы $R1 = 0.1$ и $R2 = 0.15$, толщиной линзы $d = 0.01$. Затем информация об этом объекте также будет выведена на экран. В результате сложения этих двух объектов будет получена оптическая сила комбинации линз и выведена на экран:

Оптическая сила $D = 1.25$
Фокусное расстояние $F = 0.8$
Преломляющие свойства - собирающая

Оптическая сила $D = 4.0$
Фокусное расстояние $F = 0.25$
Преломляющие свойства - собирающая

Оптическая сила комбинации линз $D = 5.25$

Объектно-ориентированный стиль программирования практически всегда уменьшает избыточность программного кода в сравнении с процедурным подходом, делает его более читабельным и универсальным.

6.6.7 Работа с графическими файлами

Язык программирования Python должен быть интересен оптотехникам, так как содержит множество мощных библиотек обработки графических данных. Широкая поддержка операций с графическими файлами обеспечивается различными модулями сторонних разработчиков, из которых наиболее функциональной является библиотека Python Imaging Library (www.pythonware.com/products/pil). Рассмотрим принципы работы с данной библиотекой на конкретном примере:

```
from PIL import Image
from PIL import ImageFilter

im = Image.open("krater.jpg")
out = im.rotate(45)
out.save("krater45.jpg")
out = im.rotate(180)
out.save("krater180.jpg")
out = im.filter(ImageFilter.BLUR)
out.save("blur.jpg")

imageW = im.size[0]
imageH = im.size[1]
r, g, b = 0, 1, 2
pixels = list(im.getdata())
if (im.mode == "RGB"):
    y = 0
    while y < imageH:
        x = 0
        while x < imageW:
            color = pixels[y*imageW + x]
            Red=color[r]*x*x/imageW/imageW*0.9
            Green=color[g]*y*y/imageH/imageH*0.8
            Blue=color[b]*x*y/imageW/imageH*0.7
            im.putpixel((x, y), (Red, Green, Blue))
            x += 1
        y += 1
im.show()
im.save("filter.jpg")
```

Как видно из примера, сначала подключаются два класса Image и ImageFilter из библиотеки PIL. Затем вызывается метод open() класса Image, возвращающий объект графического файла krater.jpg, которому присваивается имя im (рисунок 6.10, а). Этот объект можно преобразовывать стандартными методами класса Image, например, повернув графический файл с помощью метода rotate() на 45° (рисунок 6.10, б) или 180° (рисунок 6.10, в).

Для этого создается новый объект `out`, сохранить информацию в отредактированном графическом файле необходимо с помощью метода `save()`.



Рисунок 6.10 – Изображение графического файла `krater.jpg`

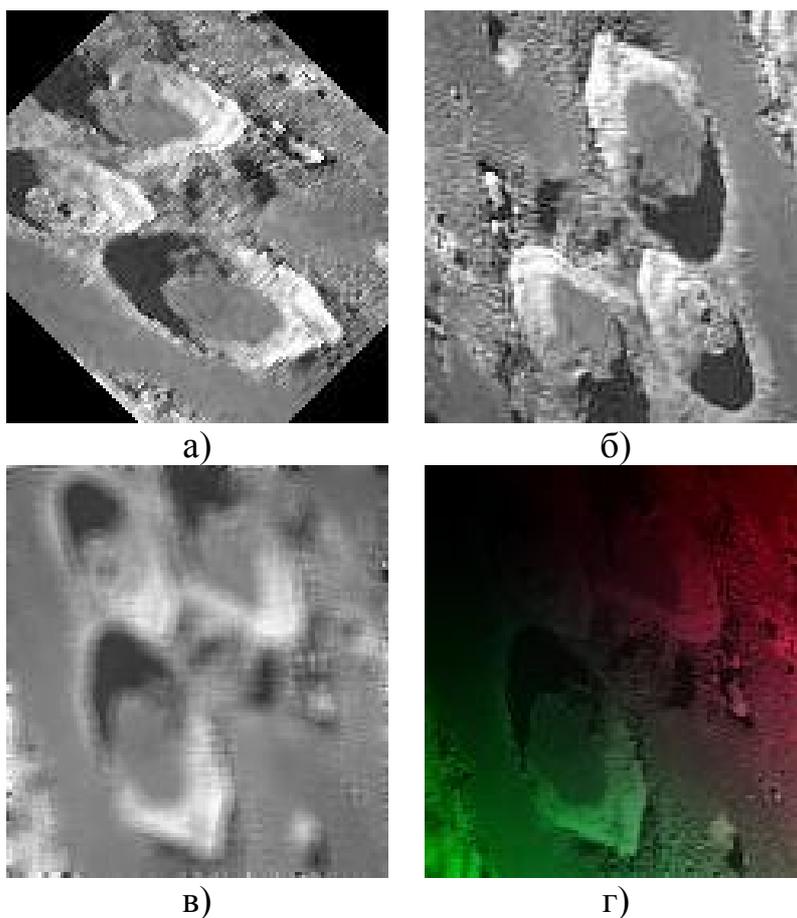


Рисунок 6.11 – Примеры обработки графического файла `krater.jpg` с помощью Python

Также можно использовать стандартные фильтры обработки графических файлов, которые находятся в классе `ImageFilter`. Для того чтобы получить размытие графического файла, нужно вызвать метод `filter` класса `Image`, параметром которого является атрибут `BLUR` класса `ImageFilter` (рисунок 6.10, г).

При необходимости можно разрабатывать и свои собственные фильтры обработки изображений. Рассмотрим, как реализуется градиентный фильтр. Для этого с помощью атрибута `size` класса `Image` определяется ширина `imageW` и высота `imageH` исходного изображения. Затем создаются три ссылки `r`, `g`, `b` на значения 0, 1, 2, соответствующие индексам основных цветов цветовой модели RGB: красного, зеленого и синего цветовых компонентов. Далее с помощью метода `getdata()` происходит обращение к данным графического файла, значения основных цветовых компонент которого для каждого пикселя изображения помещаются в список `pixels` с помощью функции `list`.

В случае, если цветовая модель графического файла RGB, то в цикле происходит перебор значений списка `pixels`, причем переменная цикла `y` указывает на номер строки и изменяется от 0 до `imageH - 1`, переменная цикла `x` указывает на номер столбца и изменяется от 0 до `imageW - 1`. Каждому значению списка `pixels` ставится в соответствие три значения: `color[r]` – красный цветовой компонент, `color[g]` – зеленый цветовой компонент, `color[b]` – синий цветовой компонент. Затем значение каждой из этих составляющих можно изменять. Например, значение красного цветового компонента нового изображения `Red` равно произведению старого значения, квадрата отношения координаты `x` к ширине изображения `imageW` и коэффициента 0.9. Следовательно, чем ближе пиксель будет находиться к левому краю изображения, тем он будет темнее, чем ближе к правому краю – тем светлее. Подобным образом формируются значения зеленого `Green` и синего `Blue` цветовых компонентов нового изображения. После этого с помощью метода `putpixel()` новое значение цвета устанавливается для пикселя с координатами (`x`, `y`) и значением цветовых компонентов (`Red`, `Green`, `Blue`). Затем вновь сформированное изображение графического файла выводится на экран с помощью метода `show()` и сохраняется в файл (рисунок 6.10, д).

Тезаурус

Алгоритм, блок-схема алгоритма, линейный алгоритм, ветвящийся алгоритм, циклический алгоритм, рекурсивный алгоритм, программа, алгоритмическое программирование, структурное программирование, объектно-ориентированное программирование, язык программирования, синтаксис, семантика, инкапсуляция, наследование, полиморфизм, типы данных, ссылки на объекты, ввод и вывод информации на экран, коллекции данных, кортеж, список, условный оператор, оператор цикла, функция, класс, объект, метод, атрибут, конструктор, перезагрузка метода.

Контрольные вопросы

1. Как в блок-схемах алгоритмов обозначается предопределенный процесс, состоящий из одной или нескольких операций, которые определены в другом месте?
2. Раскройте смысл применения рекурсивного алгоритма при построении программ. Приведите практический пример использования рекурсивного алгоритма.
3. В чем принципиальное отличие процедурного подхода от объектно-ориентированного подхода в программировании?
4. Какой основной недостаток свойственен компиляторам при генерации текста программы в машинный код?
5. Перечислите базовые понятия объектно-ориентированного программирования и поясните их смысл.
6. В какой предметной области нашел применение язык программирования Objective-C?
7. С помощью какой функции можно получить количество элементов кортежа или списка в Python?
8. Каково назначение метода `__init__()` в объектно-ориентированной модели Python?
9. Зачем внутри блока программного кода оператора цикла используются команды `break` и `continue`?
10. Опишите порядок создания собственных фильтров обработки изображения в Python.

Список рекомендуемой литературы

1. Воройский, Ф. С. Информатика : энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах [Текст] / Ф. С. Воройский. – М. : ФИЗМАТЛИТ, 2006. – 768 с.
2. ГОСТ 19.701-90 Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения [Текст] – Введ. 1992–01–01. – М. : Изд-во стандартов, 1992. – 24 с.
3. Информатика : базовый курс [Текст] / под ред. С. В. Симоновича. – СПб. : Питер, 2009. – 640 с.
4. Конев, Ф. Б. Информатика для инженеров [Текст] / Ф. Б. Конев. – М. : Высшая школа, 2004. – 272 с.
5. Лутц, М. Изучаем Python [Текст] / М. Лутц. – СПб. : Символ-Плюс, 2009. – 848 с.
6. Саммерфилд, М. Программирование на Python 3 : подробное руководство [Текст] / М. Саммерфилд. – СПб. : Символ-Плюс, 2009. – 608 с.
7. Соболев, Б. В. Информатика [Текст] / Б. В. Соболев, А. Б. Галин, Ю. В. Панов. – Ростов н/Д : Феникс, 2007. – 446 с.

Алфавитный указатель

Администрирование	127	Информатика	12
Адресная шина	30	Информационный объект	115
Алгоритм	153	Информация	10
ветвящийся	156	Каталог	65
линейный	156	Килобайт	18
рекурсивный	156	Клавиатура	44
циклический	156	Класс	115, 168
Анимация	88	Кластер	25
Антивирус	149	Клиент	143
Аппаратная конфигурация	27	Ключ	92, 115
Архитектура компьютера	23	Коаксиальный кабель	132
Атрибут	160	Кодирование	12
База данных	90	Коллекция данных	166
Байт	17	Коллизия	141
Бит	17	Коммутатор	139
Битрейт	18	Компилятор	159
Блок-схема	154	Компьютер	12
Бод	18	Компьютерная графика	100
Браузер	144	Конструктор	169
Веб-камера	49	Контроллер	24
Векторная графика	102	Конфигурация	27
Видеокарта	37	Концентратор	138
Винчестер	39	Конъюнкция	14
Вирус	146	Кортеж	166
Витая пара	132	Кэш-память	31
Вычислительные задачи	109	Латентность	36
Гигабайт	18	Линия передачи данных	131
Глобальная сеть	127	Локальная сеть	127
Голографическая память	43	Маркер	129
Данные	9	Маршрутизатор	139
Дизъюнкция	14	Материнская плата	27
Домен	145	Мегабайт	18
Дорожка	39	Метод	160
Драйвер	63	Микропроцессор	29
Жесткий диск	39	Многозадачность	64
Импликация	14	Моделирование	110
Индекс	92	Модель	110
Инкапсуляция	160	Монитор	50
Интерполяция	47	Мост	139
Интернет	142	Мультипроцессорная обработка	64
Интерпретатор	159	Мышь	45
Интерфейс	24	Наследование	161

Нормализация.....	93	Сканер.....	46
Объект.....	160	Служба.....	143
Оперативная память.....	35	Служебная программа.....	60
Оператор цикла.....	167	Спам.....	149
Операционная система.....	63	Список.....	166
Оптоволокно.....	133	Ссылки на объекты.....	165
Отношение.....	91	Стиль.....	78
Отрицание.....	14	Суперкомпьютер.....	24
Первичный ключ.....	92	Теория моделирования.....	110
Перезагрузка метода.....	171	Терабайт.....	18
Пиксель.....	101	Терминатор.....	139
Плоттер.....	57	Тип файла.....	65
Повторитель.....	139	Типы данных.....	165
Поколения ЭВМ.....	21	Топология.....	127
Полиморфизм.....	160	звезда.....	128
Политика сети.....	127	кольцо.....	129
Порт ввода-вывода.....	24	шина.....	128
Поток.....	64	ячеистая.....	130
Презентация.....	86	Точка доступа.....	136
Прерывание.....	65	Трансивер.....	135
Прикладная программа.....	62	Троянская программа.....	148
Примитив.....	102	Условный оператор.....	166
Принтер.....	55	Утилита.....	61
Программа.....	153	Файл.....	65
Программирование.....	153	Файловая система.....	65
Протокол.....	24	Фишинг.....	149
Прототип.....	110	Флопс.....	24
Процедура.....	160	Флэш-память.....	44
Разрешение.....	46	Форматирование.....	77
Растровая графика.....	101	Формула.....	84
Регистр.....	32	Функциональные задачи.....	109
Реквизит.....	115	Функция.....	160
Реляционная модель данных.....	91	Хост.....	126
Ресурс.....	126	Цвет.....	103
Ретранслятор.....	138	Цветовая модель.....	103
Сектор.....	39	Цилиндр.....	39
Семантика.....	158	Чипсет.....	29
Сетевая архитектура.....	141	Шина данных.....	30
Сетевая карта.....	138	Шина управления.....	30
Сжатие.....	102	Шлюз.....	139
Сигнал.....	9	Экземпляр.....	115, 160
Сигнатура.....	150	Электронная почта.....	144
Синтаксис.....	158	Энтропия.....	17
Система счисления.....	12	Язык программирования.....	158

НАПИСАНИЕ на ЗАКАЗ:

1. Дипломы, курсовые, чертежи...
 2. Диссертации и научные работы.
 3. Школьные задания.
- Онлайн-консультации.

ЛЮБАЯ тематика,
в том числе ТЕХНИКА.

Приглашаем авторов.

сты на заказ
on.shtml

УЧЕБНИКИ, ДИПЛОМЫ, ДИССЕРТАЦИИ:
полные тексты в электронной библиотеке
www.учебники.информ2000.рф.

Анастасия Геннадьевна Зрюмова,
Евгений Александрович Зрюмов,
Сергей Петрович Пронин

ИНФОРМАТИКА

Учебное пособие

Редактор В. Красильникова

Компьютерная верстка, макет: Е. А. Зрюмов
Дизайн обложки: Е. А. Зрюмов, В. В. Немыкин

Подписано в печать 7.02.2011. Формат 60x84 1/8.
Печать – цифровая. Усл. п. л. 20,92.
Тираж 100 экз. Заказ 2011 – 40

Издательство Алтайского государственного
технического университета им. И. И. Ползунова
656038, г. Барнаул, пр-т Ленина, 46
<http://izdat.secna.ru>

Лицензия на издательскую деятельность
ЛР № 020822 от 21.09.98 г.

Отпечатано в типографии АлтГТУ
656038, г. Барнаул, пр-т Ленина, 46
тел. (8-3852) 29-09-48

Лицензия на полиграфическую деятельность
ПЛД №28-35 от 15.07.97 г.