

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное агентство по образованию
Государственное образовательное учреждение высшего
профессионального образования
«Ростовский государственный строительный университет»

ИНФОРМАТИКА ЧАСТЬ 1:
Основные понятия, Системы счисления, Логика,
Лабораторный практикум в Excel

Учебное пособие по курсу «Информатика»

для студентов очной и заочной форм обучения

Ростов-на-Дону
2016

УДК 618.3.06

Информатика Часть 1: Основные понятия, Системы счисления, Логика,
Лабораторный практикум в Excel. Учебное пособие. -Ростов-на-Дону: РГСУ, 2016,
– 24 с.

Содержит краткий справочник основных понятий информатики, введение в системы счисления и логику высказываний.

Составители: к. ф.-м. н., доц. Ильичева О.А.,
к. ф.-м. н., ст.пр. Богачева М.Н.

Рецензенты: к. ф.-м. н., доц. Красий Н.П.,
к. ф.-м. н., доц. Глушкова В.Н.

1. Основные понятия. Краткий справочник по информатике

1.1. Техническое обеспечение компьютера

Компьютер – устройство, способное выполнять заданную последовательность операций (команд), например, операций численных расчетов, манипулирования данными, ввода и вывода информации. Простейшая конфигурация персонального компьютера состоит из центрального процессора (микропроцессора), оперативной памяти, устройства ввода (клавиатуры) и монитора, снабженного экраном (дисплеем). Эта конфигурация дополняется обычно внешней памятью, печатающим устройством, мышью.

Техническое обеспечение компьютера – аппаратные средства (*hardware*), составляющие компьютерное «железо». Это электрические и электронные схемы, электромеханические элементы – дисководы, механические стойки и т.п. Основная часть аппаратных средств находится в системном блоке компьютера.

Схема устройства компьютера представлена на рис. 1.

Системный блок содержит системную плату, блок питания, накопители на магнитных дисках, разъемы для дополнительных устройств и адаптеры внешних устройств, форматирующие поток данных.

Системная (материнская) плата – электронная схема, на которой размещаются:

- микропроцессор;
- генератор тактовых импульсов;
- блоки (микросхемы) оперативной (ОЗУ) и постоянной (ПЗУ) памяти;
- адаптеры клавиатуры, накопителей на жестких и гибких магнитных дисках;
- контроллеры (обработчики) прерываний;
- таймер (внутренние часы, работающие от автономного источника питания).

Генератор тактовых импульсов генерирует последовательность электрических импульсов. Частота генератора – основная характеристика персонального компьютера и определяет скорость его работы, т.к. каждая операция выполняется за определенное количество импульсов. Например, в описании процессора Pentium-100 число 100 – частота генератора в мегагерцах.

Адаптеры – это платы с контроллерами – устройствами (подсистемами), реализующими связь между различными техническими составляющими компьютера. Контроллеры управляют работой подключенных периферийных (внешних) устройств и каналов связи, обрабатывают (форматируют) информационные потоки.

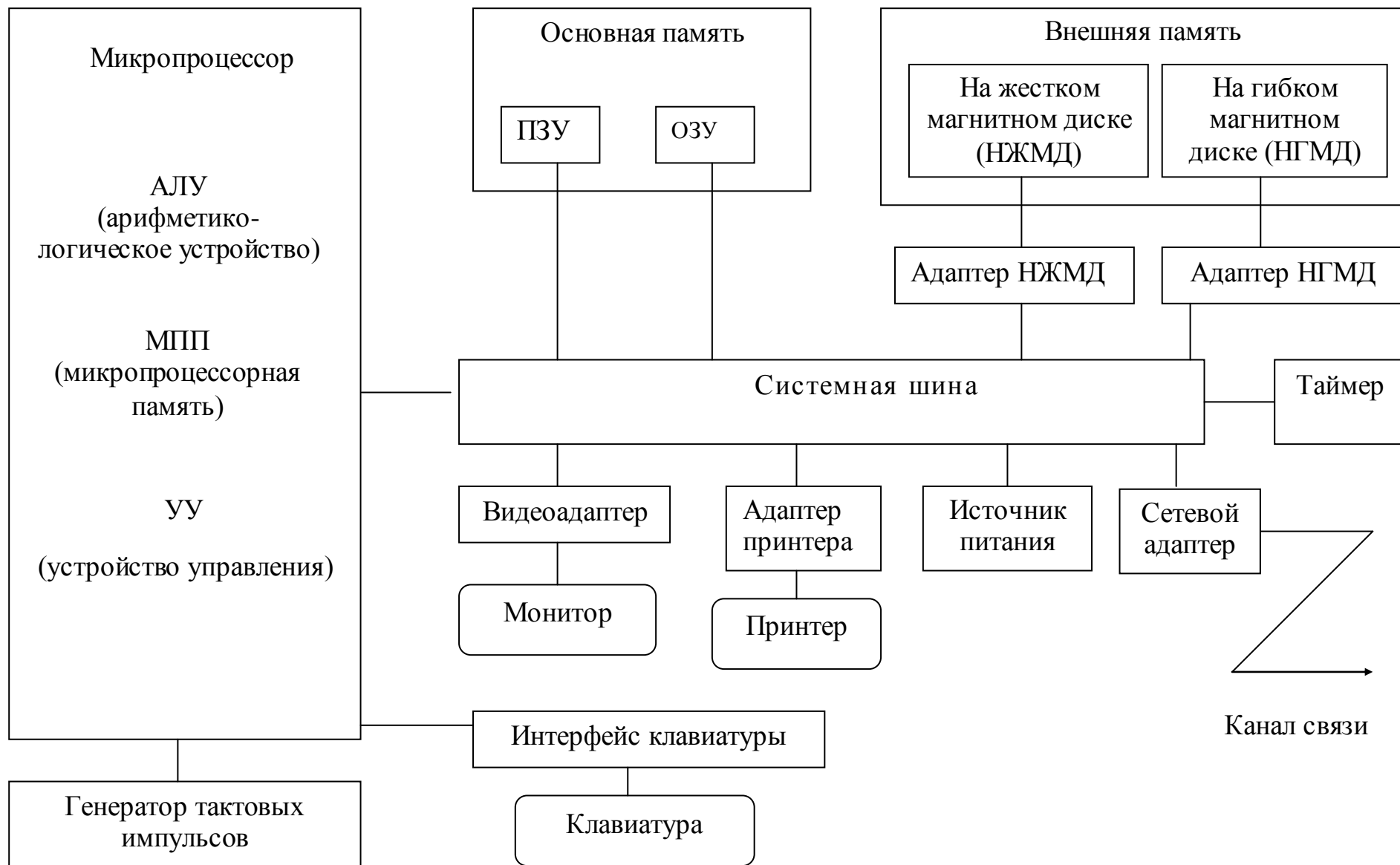


Рис. 1. Устройство персонального компьютера

Микропроцессор – центральный блок персонального компьютера, служит для выполнения арифметических и логических операций над информацией и для управления работой всех блоков машины. В его состав входят:

- УУ – устройство управления, которое формирует и подает во все блоки сигналы управления, адреса ячеек, участвующих в операциях. Базис импульсов получает от генератора тактовых импульсов.
- АЛУ – арифметико-логическое устройство для выполнения всех арифметических и логических операций над числовой, символьной информацией.
- МПП – микропроцессорная память, служит для кратковременного хранения, записи и выдачи информации, используемой в ближайших тактах. Строится на регистрах – самых «быстрых» ячейках.

Основная память компьютера – оперативное запоминающее устройство (ОЗУ) и постоянное запоминающее устройство (ПЗУ). Объем основной памяти существенно влияет на производительность компьютера: увеличение емкости основной памяти в 2 раза ведет к повышению производительности примерно в 1,7 раз.

Оперативная память (ОЗУ) служит для запоминания (хранения и считывания) текущих вычислений. Это энергозависимая память, ее содержимое уничтожается (стирается) при выключении компьютера. Основу ее составляют большие интегральные схемы, содержащие матрицы полупроводниковых запоминающих элементов (триггеров). Они расположены на пересечении горизонтальных и вертикальных шин матрицы. *Запись и считывание* реализуется подачей электрических импульсов по тем шинам матрицы, которые соединены с элементами, принадлежащими выбранной ячейке памяти.

Постоянная память (ПЗУ) используется для хранения неизменяемой (только для чтения) информации - загрузочных программ операционной системы (например, модуля BIOS системы DOS), программ тестирования устройств компьютера и некоторых драйверов.

Внешняя память компьютера служит для долговременного хранения информации. В этой памяти хранится все программное обеспечение компьютера, пользовательские файлы. Внешнюю память составляют накопители на жестких магнитных дисках (НЖМД) – «винчестеры» – и накопители на гибких магнитных дисках (НГМД) – дискетах, CD, DVD -дисках. На жестких дисках информация записывается и считывается магнитными головками вдоль концентрических окружностей – дорожек (треков). Каждая дорожка разбита на сектора (обычно по 512 или 1024 байт). Обмен данными между жестким диском и оперативной памятью идет целым числом секторов.

Видеотерминал – устройство для визуального отображения информации. Обычно это **монитор** и **видеоадаптер**. Видеоадаптер находится на видеокарте, установленной в разъем материнской платы, управляет монитором и выводом на экран. Важна емкость видеопамати (количество хранимых пикселей и их атрибутов). Монитор реализует отображение информации на электронно-лучевой трубке или жидких кристаллах. Для отображения графики важна разрешающая способность мониторов (**разрешение экрана**) – максимальное количество пикселей (точек) по горизонтали и вертикали экрана. Стандарты:

640×480 (отображает 16 цветов), 800×600 (64 цвета), 1024×768 (256 цветов). Важен размер зерна – точки люминофора экрана: чем меньше зерно, тем больше четкость изображения.

Принтер – печатающее устройство. Различают матричные, струйные и лазерные принтеры. В *матричном принтере* символы для печати формируются в виде матрицы точек. Точки переносятся на бумагу кончиками игл, ударяющими по красящей ленте. Чем больше игловок содержит принтер, тем качественнее печать. Разрешающая способность (**разрешение принтера**), характеризующая четкость печати, матричного принтера обычно не превышает 14 точек/мм. Скорость печати – не более 2 страниц в минуту. В *струйном принтере* изображение формируется без механического удара, капли чернил выбрасываются на бумагу через крохотные сопла. Разрешающая способность струйного принтера примерно 20 точек/мм, скорость – до 4 страниц в минуту. В *лазерном принтере* луч создает электронное изображение на светочувствительном барабане. Порошок – тонер налипает на незаряженные участки барабана и переносится на бумагу, затем закрепляется разогревом. Средняя разрешающая способность – 50 точек/мм, скорость – 16 страниц в минуту. Лазерный принтер обеспечивает полиграфическое качество печати.

Модем – устройство, необходимое при подключении компьютера к сети. Его составляют модулятор и демодулятор – устройства, осуществляющие преобразование потока битов (цифровой информации) в аналоговые сигналы, пригодные для передачи по некоторому аналоговому каналу связи (например, телефонному), и принимаемые аналоговые сигналы обратно в цифровую форму. По способу функционирования различают внутренний и внешний модемы. *Внутренний модем* вставляется в разъем системной платы компьютера и использует его ресурсы. *Внешний модем* имеет отдельный корпус, собственные ресурсы и размещается рядом с компьютером, соединяясь с ним кабелем.

1.2. Измерение количества информации

Единицы измерения объемов информации – «порции» памяти компьютера, используемые для представления информации. Любая информация, перерабатываемая компьютером, преобразуется им в последовательности нулей и единиц в соответствии с двоичной системой счисления.

Бит – наименьшая единица информации, «хранящая» либо 0, либо 1. В электрической схеме ноль соответствует слабому сигналу (или его отсутствию), единица – сильному (или его наличию).

Байт – это цепочка из восьми битов. Все символы клавиатуры, включая строчные и прописные буквы алфавита, могут быть закодированы различными двоичными цепочками длиной в один байт.

Килобайт (КБ) = 8×1024 битов, или 2^{10} байтов. Объем текстовых (без графики) файлов может составлять несколько килобайт.

Мегабайт (МБ) = 8×1024^2 битов, или 2^{20} байтов. Объем оперативной памяти современных компьютеров – 256 - 512 мегабайт, позволяет работать с мультимедийными документами, играми, видеофайлами. Создание трехмерных моделей в приложениях может потребовать оперативной памяти свыше 1 ГБ.

Гигабайт (ГБ) = 8×1024^3 битов, или 2^{30} байтов. Объемы современных винчестеров (жестких дисков) достигают 80 ГБ, позволяя работать с серверами баз данных и другими многопользовательскими ресурсами.

Терабайт (ТБ) = 8×1024^4 битов, или 2^{40} байтов.

1.3. Компьютерные сети

Компьютерная сеть – совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных. В зависимости от территориального расположения выделяют три класса сетей: *локальные* (в пределах организации), *региональные* (в пределах региона, страны, протяженностью на десятки, сотни километров), *глобальные* (в пределах множества стран). В качестве средства передачи информации – каналов связи – используются коаксиальные и оптоволоконные кабели (для локальных сетей), телефонные, радио- и спутниковые каналы (для глобальных сетей). Сети характеризуются *скоростью* передачи данных (биты/сек), *пропускной способностью* (количество-знаков/сек, один знак – байт), *достоверностью* передачи информации (количество-ошибок/знак, должно быть не более 10^{-7} ошибок/знак), *надежностью* каналов связи и модемов (единица измерения – час, надежная сеть – не менее нескольких тысяч часов работы).

Протокол – набор правил, по которым производится обмен информацией в сети. Протоколы определяют содержимое, формат, параметры времени, последовательность и проверку ошибок в сообщениях, которыми обмениваются сетевые устройства. Протоколы обеспечивают одинаковое восприятие информации компьютерами разных типов, работающими под руководством различных операционных систем.

Internet – глобальная компьютерная сеть, объединяющая миллионы компьютеров по всему миру. Она не имеет хозяина, большая часть информационных услуг бесплатна. Плата при подключении к сети взимается организацией-провайдером за техническое обслуживание средств связи. Передача данных по сети Internet осуществляется на основе семейства протоколов TCP/IP. Данные передаются в виде коротких пакетов (до 1,4 кбайт). В заголовке каждого пакета вместе со служебной информацией указывается IP-адрес получателя.

WWW – World Wide Web- подсеть Internet, реализующая передачу гипертекстовых документов. *Гипертекст* – это мультимедийная страница, содержащая ссылки на другие подобные документы. Ссылки (говорят, *гиперссылки*) представляют из себя *адреса* (обычно скрытые под заголовком или изображением) Интернет-страниц.

IP-адрес – цифровой (32 бита) уникальный адрес компьютера в сети.

URL-адрес – представляет имя документа в сети Internet. Структура адреса: название протокола, по которому осуществляется доступ к ресурсу, символы `://`, доменный адрес компьютера-сервера, символ `/`, имя файла на нем. Например, в адресе <http://www.osp.ru/conf/file1.htm> часть `http` – протокол, www.osp.ru – адрес компьютера (и головной страницы организации), `conf/file1.htm` – полное имя (включая путь) файла `file1`. Доменный адрес составляют несколько слов, разделенных точкой, которые отражают иерархию доменов разного уровня. Например, в адресе `uic.rnd.runnet.ru` внешний домен (домен первого уровня) – это `ru` (Россия), `runnet` – домен второго уровня (российские университеты) принадлежит домену `ru`, `rnd` – домен третьего уровня (Ростов-на-Дону), зарегистрированный в домене `runnet`, `uic` – имя компьютера в домене `rnd`, закрепленное за конкретным IP-адресом. Назначением адресов ведаёт специальная *доменная служба имен DNS*, осуществляющая преобразование доменных имен в IP-адреса. Домены верхнего уровня разделяют на географические и организационные. Например, географические домены: *ru* – Россия, *de* – Германия, *hu* – Венгрия, *uk* – Великобритания; организационные домены: *edu* – образовательные учреждения, *gov* – правительственные, *mil* – военные, *com* – коммерческие.

Сетевой домен – группа компьютеров, образующих часть сети и использующих общую базу данных каталога. Администрируется как единый объект с определенными правилами и процедурами.

http – протокол передачи гипертекстовых документов в сети WWW.

ftp – протокол передачи файлов в сети Internet.

Открытая система – система, взаимодействующая с другими системами в соответствии с принятыми стандартами (протоколами). Международная организация стандартов *ISO* разработала модель взаимодействия открытых систем *OSI*. В данной модели средства сетевого взаимодействия разделены на 7 уровней: *прикладной, представительный, сеансовый, транспортный, сетевой, канальный, физический*. Каждый уровень отвечает за способ и средства представления и передачи информационного потока. Функции физического уровня всегда реализуются в аппаратуре, функции остальных уровней реализуются в виде программных модулей – драйверов.

Модель сетевого взаимодействия «клиент-сервер» – технология взаимодействия компьютеров в сети, в которой один компьютер, обладающий некоторым ресурсом (например, базой данных, электронной почтой, файловой системой) и называемый *сервером* этого ресурса, предоставляет услуги другому компьютеру – *клиенту*, который хочет этим ресурсом воспользоваться. По отношению к другому ресурсу сервер может выступать в роли клиента и наоборот. Если говорят, что проектируется информационная система по технологии «клиент-сервер», это значит, что данная система (приложение) будет иметь *распределенный характер*, т.е. часть функций приложения будет реализована в программе клиента, а другая – в программе сервера, и для них будет определен некоторый протокол.

1.4. Программное обеспечение компьютера

Программа – набор команд, написанный на каком-либо (специальном) языке программирования, предназначенный для исполнения компьютером.

Программное обеспечение компьютера – программные средства (*Software*), исполняемые вычислительной системой. Различают системное программное обеспечение, которое является необходимым дополнением к техническим средствам (операционные системы, компиляторы, драйверы, сервисные программы), и прикладное – комплекс программ, обслуживающих специфические задачи организации.

Операционная система – комплекс программ, обеспечивающий диалог пользователя с компьютером, распределение ресурсов компьютера и управление ими. Эффективное управление ресурсами – памятью и процессорным временем позволяет осуществить многозадачный, многопользовательский режим работы, когда несколько пользователей (задач) могут одновременно работать с информацией, содержащейся в памяти компьютера.

Windows – операционная система, разработанная фирмой Microsoft как графическая оболочка (среда) над операционной системой DOS (Disk Operation System). Windows значительно облегчила работу с компьютером для непрофессиональных пользователей, предоставив возможность диалога с помощью меню, окон, графических кнопок, подсказок. Последние версии этой системы развиваются в направлении упрощения работы с сетью Internet.

Файл – именованная совокупность данных любого типа, хранимая и обрабатываемая как единое целое. Является учетной единицей информации для файловой системы, которая снабжает файл атрибутами-характеристиками, в число которых входят: полное имя файла (включающее путь к нему и его тип), объем в байтах, дата и время создания, дата и время изменения, тип доступа. Физически файл хранится во внешней памяти. Тип информации, содержащейся в файле, представлен в имени файла трехбуквенной аббревиатурой, называемой *расширением* имени. Расширение записывается через точку после пользовательского имени, например, письмо.txt. Некоторые расширения и связанные с ними типы файлов:

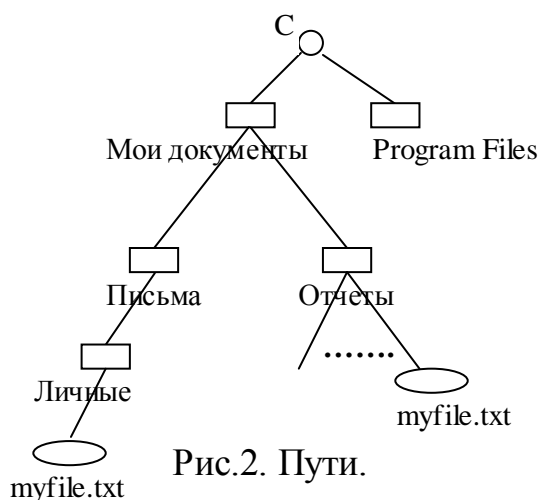
- .txt – текстовый файл;
- .doc – документ приложения Word – текст с возможной графикой (таблицами, рисунками, диаграммами и т.п.);
- .xls – электронная таблица, созданная в приложении Excel;
- .exe – исполняемый файл, например, программа на машинном языке;
- .com – командный, системный, исполняемый файл;
- .sys – файл операционной системы, драйвер;
- .bmp – графический файл формата BITMAP, например, точечный рисунок; файлы этого формата практически несжимаемы, имеют поэтому большой размер;
- .gif – графический файл формата GIF – самого плотного из форматов, хорошо подходящего для малоцветных изображений;

- .jpg – графический файл формата JPEG, обеспечивающего огромную степень сжатия с незначительной потерей информации, хорошо подходит для хранения и пересылки в сети цветных фото;

- .png – графический файл, передаваемый в сети WWW без потери качества;

- .htm – гипертекстовый документ – страница Интернет.

Папка, каталог, директория – именованная совокупность файлов и (вложенных) папок. Каталоги и директории – более ранние термины, папки – термин Windows. Каталоги имеют структуру «дерева», корнем кверху. Корневой каталог создается системой, имеет обозначение \ (в операционной системе DOS) и не может быть удален средствами этой системы.



Путь – цепочка соподчиненных каталогов, которую нужно пройти, добираясь до искомого файла. Путь изображается последовательностью имен вложенных друг в друга папок, начинающейся логическим (однобуквенным) именем диска; имена отделяются символом \. Например, путь к файлу myfile.txt, представленный как

C:\Мои документы\Письма\Личные\myfile.txt, говорит о том, что этот файл находится в папке Личные, содержащейся в папке

Письма, которая в свою очередь находится в папке Мои документы на диске C. Разные папки могут хранить файлы с одинаковыми именами, но, поскольку пути к ним разные, эти файлы считаются различными, даже если они хранят одну и ту же информацию. Например, схема иерархии каталогов на рисунке демонстрирует кроме приведенного выше пути путь к другому файлу myfile.txt – цепочку C:\Мои документы\ Отчеты\myfile.txt.

Файловая система – комплекс программ, ведающий созданием, удалением, чтением, записью, модификацией и перемещением файлов, а также управлением доступом к файлам и организацией их хранения.

Транслятор – программа, которая преобразует (переводит) программу, написанную на одном языке, в программу, написанную на другом языке.

Компилятор – частный случай транслятора, перевод осуществляется на язык ассемблера (язык, близкий к машинному коду) или в машинный код.

Интерпретатор – языковой процессор, который построчно анализирует исходную программу и одновременно выполняет предписанные действия, не формируя скомпилированную программу, выполняемую впоследствии.

База данных – структурированная совокупность связанных между собой наборов данных – текстовой, числовой, графической информации. В зависимости от способа структурирования и связывания информации различают иерархические, сетевые и реляционные базы данных. Наиболее популярный тип баз данных – реляционные – представляют собой набор связанных между собой «плоских» (не содержащих дополнительных структур)

таблиц. Обслуживающее базы данных программное обеспечение – системы управления базами данных (СУБД) осуществляют доступ к данным по запросам, модификацию данных, пополнение и удаление информации, организуют хранение данных, следят за непротиворечивостью и неизбыточностью хранимой информации (контроль целостности), обеспечивают восстановление базы после сбоев и аварий.

1.5. Компьютерная графика

Компьютерная графика – раздел приложений информатики для создания, редактирования и преобразования разнообразной графической информации. Различные ее виды – растровая, векторная и фрактальная графика отличаются принципами формирования изображения при отображении на экране или бумаге.

Растровая графика применяется при разработке полиграфических и мультимедийных изданий; соответствующие программы-редакторы в большей степени ориентированы не на создание, а на обработку изображений, вводимых с цифровых фото- и видеокамер. Основным элементом растрового изображения является *точка* (на экране она называется пикселем). В зависимости от разрешения экрана на нем могут размещаться изображения, имеющие 640×480, 800×600, 1024×768 и более пикселей. С размером изображения связано его разрешение.

Разрешение изображения измеряется в точках на дюйм (dot per inch – dpi). Например, у монитора с диагональю 15 дюймов размер изображения на экране составляет примерно 28×21 см. В одном дюйме 25,4 мм, поэтому при работе монитора в режиме 800×600 пикселей разрешение экранного изображения равно 72 dpi. При печати разрешение должно быть намного выше. Полиграфическая печать полноцветного изображения требует разрешения 200–300 dpi. Стандартный фотоснимок 10×15 см должен содержать примерно 1000×1500 пикселей. Всего такое изображение будет иметь 1,5 млн точек и если на кодирование одной точки, включая ее цвет, использованы 3 байта, то обычный цветной фотоснимок займет массив размером более 4 МБ. Основные проблемы использования растровых изображений – большие размеры файлов и невозможность увеличения без потери качества (четкости) изображения. Различные форматы растровой графики – BITMAP, JPEG, GIF, PNG, TIFF отличаются друг от друга степенью сжимаемости изображений, влияющей на объемы графических файлов, что особенно существенно при пересылке изображений по сети.

Цветовое разрешение – глубина цвета – определяет метод кодирования цветовой информации, и показывает, сколько цветов можно отобразить на экране одновременно. Для кодирования черно-белого изображения достаточно одного бита на представление цвета одного пикселя. Одним байтом можно кодировать 256 цветов (режим High Color), двумя – 65536 цветов (режим High Color), тремя – 16,5 млн цветов (режим True Color).

Цветовая модель – способ разделения цветового оттенка на составляющие компоненты. Чаще применяются три модели: RGB, CMYK, HSB. Наиболее простая модель, в которой работают мониторы и телевизоры – RGB. Любой цвет в этой модели считается состоящим из трех компонентов: Red, Green, Blue. Эти цвета называют основными. Новые оттенки получаются суммированием яркостей составляющих компонентов. Считается, что при наложении одного компонента на другой яркость суммарного цвета увеличивается. Наложение всех трех компонентов дает серый цвет, который при большей яркости становится белым.

Цветовая палитра – таблица данных, в которой хранится информация о том, каким кодом закодирован тот или иной цвет. Эта таблица создается и хранится вместе с графическим файлом. Самый удобный для компьютера способ кодирования цвета – трехбайтовый (24 разряда). В этом режиме на кодирование каждой цветовой составляющей R (красной), G (зеленой) и B (синей) отводится по одному байту. Яркость каждой составляющей выражается числом от 0 до 255, и любой цвет из 16,5 миллионов компьютер может воспроизвести по трем кодам. В этом случае цветовая палитра не нужна, поскольку в трех байтах достаточно информации о цвете конкретного пикселя.

Векторная графика имеет основным элементом изображения *линию*.

Объем памяти, занимаемый линией не зависит от размеров линии, т.к. линия представлена формулой – несколькими параметрами. Манипуляции с линией изменяют только значения этих параметров. Для их хранения достаточно нескольких байтов памяти. Примеры линий:



Рис.3. Линии векторной графики.

Векторная иллюстрация состоит из линий, простейшие объекты объединяются в более сложные. Например, четырехугольник представляется четырьмя связанными линиями, куб – двенадцатью или шестью связанными четырехугольниками. Линии имеют свойства: толщину, цвет, форму, характер (сплошная линия или пунктир). Замкнутые линии имеют свойство заполнения внутренней области (текстурой, узором, цветом). В основе представления векторных изображений лежат математические формулы задания фигур: точки (два числа – координаты точки на плоскости), прямой (два параметра – коэффициента в уравнении прямой), отрезка (еще два параметра – координаты начала и конца отрезка), кривых (пять параметров для кривой второго порядка и девять – для кривой третьего порядка) и т. п. При выводе векторного изображения на экран программа производит вычисления координат экранных точек в изображении объекта, т.к. экран выводит все изображения в виде точек. Векторная графика легко масштабируется, увеличение рисунка не ведет к «размыванию» изображения, поэтому она широко используется в картографии, конструкторских системах автоматизированного проектирования и системах архитектурного проектирования.

Размер изображения. Физический размер изображения измеряют как в пикселях, так и в единицах длины (миллиметрах, сантиметрах, дюймах). Он задается при создании изображения и хранится вместе с файлом. Если изображение предназначено для демонстрации на экране, то его ширину и высоту задают в пикселях, чтобы знать, какую часть экрана оно займет. Если изображение печатается, то удобно использовать единицы длины для определения его положения на листе бумаги. С размером изображения связано его разрешение, являющееся свойством самого изображения. Разрешение измеряется в точках на дюйм и это значение хранится в файле изображения. Соответствие между размерами изображения и его разрешением представлено в следующих таблицах.

Таблица 1. Связь между линейным размером иллюстрации и размером файла при разных разрешениях изображения

Размер отпечатка	75 dpi	150 dpi	300 dpi	600 dpi
10×15 см (фотоснимок)	380 КБ	1,5 МБ	6 МБ	24 МБ
25×30 см (обложка журнала)	1,9 МБ	7,5 МБ	30 МБ	120 МБ
50×30 см (разворот журнала)	3,8 МБ	15 МБ	60 МБ	240 МБ

Таблица 2. Связь между размером иллюстрации (в пикселях) и размером отпечатка (в мм) при разных разрешениях изображения

Размер иллюстрации	75 dpi	150 dpi	300 dpi	600 dpi
640 × 480	212×163 мм	108×81 мм	55×40 мм	28×20 мм
800 × 600	271×203 мм	136×102 мм	68×51 мм	34×26 мм
1024 × 768	344×260 мм	173×130 мм	88×66 мм	44×33 мм
1152 × 864	390×293 мм	195×146 мм	98×73 мм	49×37 мм
1600 × 1200	542×406 мм	271×203 мм	136×102 мм	68×51 мм

Высококачественная печать полноцветного изображения обеспечивается при его разрешении в 200-300 dpi. При печати изображения, занимающего полный экран очень большого монитора образуется отпечаток размером с небольшую фотографию.

2. Представление числовой информации

2.1. Системы счисления

Система счисления – это способ представления чисел и соответствующие ему правила действия над числами. Разнообразные системы счисления, которые существовали и раньше и которые используются и в наше время, можно разделить на непозиционные и позиционные. Знаки, используемые при записи чисел, называются цифрами.

В **непозиционных системах счисления** от положения цифры в записи числа не зависит величина, которую она обозначает. Примером непозиционной системы счисления является римская система (римские цифры).

В **позиционных системах счисления** величина, обозначаемая цифрой в записи числа, зависит от ее позиции. Количество используемых цифр называется **основанием** позиционной системы счисления.

Примеры алфавитов нескольких систем:

Таблица 1. Алфавиты систем счисления

Основание	Название	Алфавит
n=2	Двоичная	0 1
n=3	Троичная	0 1 2
n=8	Восьмеричная	0 1 2 3 4 5 6 7
n=16	Шестнадцатеричная	0 1 2 3 4 5 6 7 8 9 A B C D E F

Если требуется указать основание системы, к которой относится число, то оно приписывается нижним индексом к этому числу. Например: 101101_2 , 3671_8 , $3B8F_{16}$. Позиция цифры в числе называется разрядом. Разряд числа возрастает справа налево от младших разрядов к старшим. Число 555_{10} записано в привычной для нас свернутой форме. Если расписать число, используя различные степени числа 10, то получим следующее выражение $555_{10} = 500 + 50 + 5 = 5 \cdot 100 + 5 \cdot 10 + 5 = 5 \cdot 10^2 + 5 \cdot 10^1 + 5 \cdot 10^0$. Таким образом, число в позиционной системе счисления записывается в виде суммы числового ряда степеней основания (в данном случае 10), в качестве коэффициентов которых выступают цифры данного числа.

Для перевода чисел из двоичной системы в десятичную используется ряд степеней двойки:

Таблица 2. Степени числа 2

Значение степени	1024	512	256	128	64	32	16	8	4	2	1
Показатель степени	10	9	8	7	6	5	4	3	2	1	0

Если все слагаемые в развернутой форме недесятичного числа представить в десятичной системе и вычислить полученное выражение по правилам десятичной арифметики, по получится число в десятичной системе, равное данному.

Пример С1. Перевести число 101101_2 из двоичной системы в десятичную систему счисления.

Решение. Над числом запишем степени основания двоичной системы, т.е. степени двойки, затем рассмотрим развернутую запись числа,
 543210
 $101101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45_{10}$
 ■

2.2. Перевод десятичных чисел в другие системы счисления

Перевод целых чисел производят следующим образом:

1) Основание новой системы счисления выразить в десятичной системе счисления и все последующие действия производить в десятичной системе счисления;

2) последовательно выполнять деление данного числа и получаемых неполных частных на основание новой системы счисления до тех пор, пока не получим неполное частное, меньшее делителя;

3) полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления;

4) составить число в новой системе счисления, записывая его, начиная с последнего частного.

Пример С2. Перевести число 37_{10} в двоичную систему счисления.

Решение. Для обозначения цифр в записи числа используем символику: $a_5 a_4 a_3 a_2 a_1 a_0$.

$$\begin{array}{r}
 \begin{array}{l}
 \begin{array}{r}
 \underline{37} \\
 \underline{36} \\
 \hline
 1
 \end{array} \\
 a_0 = 1
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 \underline{18} \\
 \underline{18} \\
 \hline
 0
 \end{array} \\
 a_1 = 0
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 \underline{9} \\
 \underline{8} \\
 \hline
 1
 \end{array} \\
 a_2 = 1
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 \underline{4} \\
 \underline{4} \\
 \hline
 0
 \end{array} \\
 a_3 = 0
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 \underline{2} \\
 \underline{2} \\
 \hline
 1
 \end{array} \\
 a_4 = 0
 \end{array}
 \begin{array}{l}
 \begin{array}{r}
 \underline{2} \\
 \underline{2} \\
 \hline
 1
 \end{array} \\
 a_5 = 1
 \end{array}
 \end{array}$$

Отсюда $37_{10} = 100101_2$. ■

2.3. Арифметические операции в двоичной и кратных ей системах счисления.

Арифметические операции в позиционных системах счисления производятся по единому алгоритму. Так, сложение двоичных чисел происходит по классическому алгоритму «столбиком» с переносом двойки в следующий ряд. Необходимо помнить о следующих правилах сложения и умножения чисел в двоичной системе счисления.

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=10$

$0*0=0$
$0*1=0$
$1*0=0$
$1*1=1$

Важно обратить внимание на то, что при сложении двух единиц происходит переполнение разряда и производится перенос в старший разряд. Переполнение разряда наступает тогда, когда величина числа в нем становится равной или большей основания. Проверим, действительно ли $1_2 + 1_2 = 10_2$: переведем слагаемые в десятичную систему счисления. $1_2 = 1_{10}$, поэтому $1_2 + 1_2 = 1_{10} + 1_{10} = 2_{10} = 10_2$.

Пример С3. Найти сумму чисел 1010101_2 и 110111_2 .

Решение.

$$\begin{array}{r}
 \text{Первое слагаемое} \quad + \quad 1010101 \\
 \text{Второе слагаемое} \quad \underline{0110111} \\
 \text{Сумма} \quad \quad \quad \underline{10001100}
 \end{array}$$

Проверим результат нашего сложения, для чего переведем слагаемые и сумму в десятичную систему счисления:

$$1010101_2 = 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 64 + 0 + 16 + 0 + 4 + 1 = 85_{10}$$

$$110111_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32 + 16 + 4 + 2 + 1 = 55_{10}$$

$$10001100_2 = 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 128 + 8 + 4 = 140_{10}$$

Как видим, действительно $85 + 55 = 140$. ■

Двоичная система, являющаяся основой всей компьютерной арифметики, тем не менее весьма громоздка и не удобна для использования человеком. Поэтому программисты пользуются двумя кратными двоичной системами счисления: восьмеричной и шестнадцатеричной.

Приведем в качестве примера запись натуральных чисел от единицы до шестнадцати в четырех системах счисления. Для удобства перевода из двоичной в восьмеричную и шестнадцатеричную системы рассмотрим следующую таблицу:

Таблица 3. Перевод чисел из одной системы счисления в другую

10-ная	2-ная	8-ная	16-ная	2-8	2-16
0	0	0	0	000	0000
1	1	1	1	001	0001
2	10	2	2	010	0010
3	11	3	3	011	0011
4	100	4	4	100	0100
5	101	5	5	101	0101
6	110	6	6	110	0110
7	111	7	7	111	0111
8	1000	10	8		1000
9	1001	11	9		1001
10	1010	12	A		1010
11	1011	13	B		1011
12	1100	14	C		1100
13	1101	15	D		1101
14	1110	16	E		1110
15	1111	17	F		1111

Из этой таблицы видно, что в двоичной системе счисления запись числе второй восьмерки чисел (от 8 до 15) отличается от записи первой восьмерки (от 0 до 7) наличием единицы в четвертом (слева) разряде. На этом основан алгоритм перевода двоичных чисел в восьмеричные «по триадам». Для применения этого алгоритма надо разбить двоичное число на тройки цифр (естественно, отсчитывая справа по три цифры для целого числа и слева для дробного числа) и записать вместо каждой из троек восьмеричную цифру.

Пример С4. Перевести число 1010101_2 в восьмеричную систему счисления.

Решение.

Для решения необходимо разбить число справа (т.к. оно целое) на «триады». Если до крайней слева тройки не хватает цифр, то дописываем незначащие нули слева.

$$1010101_2 \rightarrow 1\ 010\ 101_2 \rightarrow \underbrace{001}_1 \underbrace{010}_2 \underbrace{101}_5_2 \rightarrow 1\ 2\ 5_8 \rightarrow 125_8. \blacksquare$$

Для перевода чисел из восьмеричной системы в двоичную используется обратный алгоритм: (восьмеричные цифры заменяются на тройки двоичных цифр).

$$\text{Например, } 327_8 \rightarrow 3\ 2\ 7_8 \rightarrow \underbrace{011}_3 \underbrace{010}_2 \underbrace{111}_7_2 \rightarrow 11010111_2.$$

Для перевода чисел из двоичной системы в шестнадцатеричную используется алгоритм «по тетрадам». Строка двоичных цифр разбивается на четверки и вместо них записываются шестнадцатеричные цифры.

Пример С5. Перевести двоичное число $11011110101110111,101$ в шестнадцатеричную систему счисления.

Решение. Разделим данное число на группы по четыре цифры, начиная справа для целой части числа и слева для дробной части числа. Если в крайней левой группе (для целой части) и в крайней правой (для дробной части) окажется меньше четырех цифр, то дополним их нулями.

$$\begin{aligned} \underline{11011110101110111,101}_2 &\rightarrow 11\ 0111\ 1010\ 1110\ 1111,101_2 \rightarrow \\ \underbrace{0011}_3\ \underbrace{0111}_7\ \underbrace{1010}_A\ \underbrace{1110}_E\ \underbrace{1111}_F, \underbrace{1010}_A &\rightarrow 37AEF, A_{16} \end{aligned}$$

Следовательно, $11011110101110111,101_2 = 37AEF, A_{16}. \blacksquare$

3. Основы логики высказываний

3.1. Высказывания. Логические операции, выражения

Высказывание – это утверждение (предложение), о котором можно сказать, истинно оно или ложно. Например, высказывание «город Сочи расположен на берегу Черного моря» истинно, а высказывание «город Ростов расположен на берегу Черного моря» ложно. Из простых высказываний А, В можно образовать более сложные высказывания: «А и В», «А или В», «неверно, что А», «если А, то В» («из А следует В»). Зная истинность или ложность утверждений А, В, можно установить истинность или ложность сложного (составного) высказывания.

Высказывание можно формализовать с помощью логической формулы. Логическая формула включает в себя логические переменные и логические связки (знаки логических операций). Переменные представляют утверждения и обозначаются обычно буквами латинского или русского алфавита. Связки – это *операции*:

- конъюнкция (обозначения \wedge , $\&$, \cdot , AND, И);

- дизъюнкция (обозначения \vee , OR, ИЛИ);
- отрицание (обозначения \neg , \sim , NOT, НЕ, \bar{A} для высказывания A);
- импликация (обозначения \rightarrow , \supset).

Далее используются первые из указанных в списках обозначений.

Например, высказывание «если будет дождь, мы не поедem в гости, будем сидеть дома» можно формально представить формулой $A \rightarrow \neg B \wedge C$, где переменная A в данном случае представляет высказывание «будет дождь», B – высказывание «поедем в гости», C – «будем сидеть дома». Прочитать такую формулу можно так: «из A следует не B и C ».

Операнды дизъюнкции называют *дизъюнктами*, операнды конъюнкции – *конъюнктами*. В импликации $\phi \rightarrow \psi$ левый операнд, формулу ϕ , называют *посылкой*, а правый операнд, формулу ψ , – *заключением*. Читают импликацию как «из ϕ следует ψ », или « ϕ влечет ψ ».

Приоритеты операций: существует договоренность о порядке выполнения логических операций, если этот порядок не размечен круглыми скобками. Наивысший приоритет имеют отрицание и скобки, затем конъюнкция, далее выполняется дизъюнкция и последней – импликация.

Чтение формул. Формулы необходимо читать с учетом приоритетов операций. Например, формулу $\neg (x \rightarrow \neg y) \rightarrow z$ можно прочесть так: из того, что не выполняется условие «из x следует не y », вытекает (логически следует) z . Другой правильный вариант: если неверно, что из x следует не y , то выполняется z . Прочтение «если не x , то не y влечет z » является неверным и неоднозначным, т.к. соответствует формуле: $\neg x \rightarrow (\neg y \rightarrow z)$ и формуле: $(\neg x \rightarrow \neg y) \rightarrow z$. ■

3.2. Построение формул по высказываниям

Пример Л1. «для того, чтобы треугольники были равны, необходимо, чтобы они были подобны». Обозначим простые высказывания переменными: x – «треугольники равны» и y – «треугольники подобны». Тогда формула: $x \rightarrow y$ соответствует исходному высказыванию. Обратите внимание, что *необходимое* условие идет *справа* от операции следования: если треугольники равны, то они точно будут подобны. Обратное, $y \rightarrow x$, неверно – из подобия равенства не следует. ■

Пример Л2. Высказывание: «Для того, чтобы были лужи, достаточно, чтобы прошел дождь». Обозначим x – «были лужи», y – «прошел дождь». Формула: $y \rightarrow x$ формализует исходное высказывание. Обратите внимание, что *достаточное* условие идет *слева* от операции следования: если был дождь, то есть и лужи. Обратное, $x \rightarrow y$ неверно, т.к. лужи могут быть вызваны не дождем, а, например, водопроводной аварией. ■

Пример Л3. Высказывание: «для того, чтобы число было четным, необходимо и достаточно, чтобы оно делилось на два без остатка». Обозначим

x – «число четное», y – «число делится на два без остатка». Формула: $(x \rightarrow y) \wedge (y \rightarrow x)$ формализует исходное высказывание. ■

Примеры ошибочного толкования следования:

1) Из высказываний «все зебры полосаты» и «это животное полосато» следует, что «это животное – зебра». Так, полосатый кот становится зеброй.

2) Из высказываний «людей много» и «Сократ – человек» следует, что «Сократов много».

3.3. Определение истинности формул

Задача определения истинности формул решается в соответствии с принятыми правилами интерпретации высказываний в логике. Стандартная (главная) интерпретация операций в исчислении высказываний представляется следующей таблицей:

Таблица 4. Таблица истинности логических операций

x	y	$\neg x$	$x \wedge y$	$x \vee y$	$x \rightarrow y$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

Здесь цифрой 0 обозначено значение «ложь», цифрой 1 – значение «истина».

Эквивалентность формул означает совпадение их значений истинности для всех возможных наборов входящих в них переменных. Операцию эквивалентности обозначают, обычно, знаком тождества \equiv .

Существуют формулы, имеющие одно и то же значение, при различных значениях входящих в них переменных. К ним относятся тавтология и противоречие.

Тавтология – это формула, истинная при любой интерпретации входящих в нее переменных. Так, формула $x \vee \neg x$ всегда истинна. Действительно, значение дизъюнкции есть истина, если хотя бы один ее операнд истинен, а в этой формуле, если x - ложь, то $\neg x$ - истина.

Противоречие – это формула, ложная при любой интерпретации входящих в нее переменных. Так, формула $x \wedge \neg x$ всегда ложна. Действительно, значение конъюнкции есть ложь, если хотя бы один ее операнд ложен, а в этой формуле, если x - истина, то $\neg x$ – ложь.

Если заданы значения переменных, то, используя стандартную интерпретацию, можно определить, истинна или нет данная формула.

3.4. Определение истинности формул с помощью таблиц истинности

Пример Л4. Определить, истинна или нет формула $\neg(x \vee \neg y) \rightarrow x \wedge y$, если а) $x=1, y=0$; б) $x=0, y=1$.

Решение. Для решения задачи нужно подставить данные значения x и y в формулу и использовать интерпретацию операций, учитывая их приоритет. Так, для а): $\neg(1 \vee \neg 0) \rightarrow 1 \wedge 0 \equiv \neg 1 \rightarrow 0 \equiv 0 \rightarrow 0 \equiv 1$. Ответ: при $x=1, y=0$ данная формула истинна. Для б): $\neg(0 \vee \neg 1) \rightarrow 0 \wedge 1 \equiv \neg(0 \vee 0) \rightarrow 0 \equiv \neg 0 \rightarrow 0 \equiv 1 \rightarrow 0 \equiv 0$. Ответ: при $x=0, y=1$ данная формула ложна. Этот процесс можно представить таблицей:

x	y	$\neg y$	$x \vee \neg y$	$\neg(x \vee \neg y)$	$x \wedge y$	$\neg(x \vee \neg y) \rightarrow x \wedge y$
1	0	1	1	0	0	1
0	1	0	0	1	0	0

Таковыми таблицами удобно пользоваться, если формула сложная и требуется определить ее истинность для всех возможных значений переменных. ■

Пример Л5. Определить истинность формулы $\varphi \equiv \neg(x \wedge y) \rightarrow \neg x \vee \neg y \vee \neg z \rightarrow x \vee y \vee z$ для всех значений переменных x, y, z .

Решение. Решаем задачу с помощью таблицы, разбивая исходную формулу на подформулы:

x	y	z	$x \wedge y$	$\neg(x \wedge y)$ $\equiv \varphi_1$	$\neg x$	$\neg y$	$\neg z$	$\neg x \vee \neg y \vee \neg z$ $\equiv \varphi_2$	$x \vee y \vee z$ $\equiv \varphi_3$	$\varphi_1 \rightarrow \varphi_2$ $\equiv \varphi_4$	$\varphi_4 \rightarrow \varphi_3$ $\equiv \varphi$
0	0	0	0	1	1	1	1	1	0	1	0
0	0	1	0	1	1	1	0	1	1	1	1
0	1	0	0	1	1	0	1	1	1	1	1
0	1	1	0	1	1	0	0	1	1	1	1
1	0	0	0	1	0	1	1	1	1	1	1
1	0	1	0	1	0	1	0	1	1	1	1
1	1	0	1	0	0	0	1	1	1	1	1
1	1	1	1	0	0	0	0	0	1	1	1

Создав такую таблицу, можно ответить на вопрос: является ли данная формула тавтологией? Ответ – нет, т.к. при $x=0, y=0, z=0$ ее значение есть 0, а тавтология истинна при любых x, y, z . Является ли эта формула противоречием? Нет, т.к. есть наборы переменных, при которых она истинна. Из таблицы видно, что, например, «усеченная» формула $\neg(x \wedge y) \rightarrow \neg x \vee \neg y \vee \neg z$ является тавтологией (все ее значения истинны), а, соответственно, ее отрицание $\neg(\neg(x \wedge y) \rightarrow \neg x \vee \neg y \vee \neg z)$ будет противоречием (все значения ложны). ■

3.5. Упрощение формул

Следующая важная задача в приложениях логики высказываний – *упрощение формул*. Под упрощением понимается получение более простой (например, более короткой, не содержащей знаков \rightarrow , скобок, отрицаний над составными формулами) формулы, эквивалентной данной. Для этого используются эквивалентные преобразования формул, основанные на следующих известных тождествах (правилах):

- | | |
|--|---|
| 1) $\neg (A \vee B) \equiv \neg A \wedge \neg B$ | 12) $A \vee B \equiv B \vee A$ |
| 2) $\neg (A \wedge B) \equiv \neg A \vee \neg B$ | 13) $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$ |
| 3) $A \wedge (B \vee C) \equiv A \wedge B \vee A \wedge C$ | 14) $(A \vee B) \vee C \equiv A \vee (B \vee C)$ |
| 4) $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ | 15) $(A \equiv B) \equiv (B \equiv A)$ |
| 5) $A \vee (A \wedge B) \equiv A$ | 16) $A \rightarrow (B \rightarrow C) \equiv A \wedge B \rightarrow C$ |
| 6) $A \wedge (A \vee B) \equiv A$ | 17) $A \vee A \equiv A$ |
| 7) $(A \wedge B) \vee \neg B \equiv A \vee \neg B$ | 18) $A \wedge A \equiv A$ |
| 8) $\neg \neg A \equiv A$ | 19) $A \vee B \vee \neg B \equiv B \vee \neg B$ |
| 9) $A \rightarrow B \equiv \neg A \vee B$ | 20) $A \vee B \wedge \neg B \equiv A$ |
| 10) $A \rightarrow B \equiv \neg B \rightarrow \neg A$ | 21) $A \wedge (B \vee \neg B) \equiv A$ |
| 11) $A \wedge B \equiv B \wedge A$ | |

Здесь A, B, C – (под)формулы, в частности, логические переменные. Обычно, при преобразованиях вначале избавляются от импликаций с помощью правила 9, затем от отрицаний над составными формулами (правила 1, 2, 8) и скобок. Если в конечном результате преобразований получена тавтология, например, $x \vee \neg x$, то исходная формула также является тавтологией, т.к. она эквивалентна полученной. Аналогично, результат $x \wedge \neg x$ говорит о противоречивости исходной формулы. Правило 19 говорит о том, что в дизъюнкции подформула-тавтология и будет результатом, т.к. она всегда истинна, а для истинности дизъюнкции достаточно истинности хотя бы одного операнда. Правило 20 говорит о том, что противоречие не влияет на результат дизъюнкции, т.к. оно всегда ложно и результат определяется истинностью или ложностью оставшейся формулы. Соответственно тавтология не влияет на результат конъюнкции (правило 21), т.к. она всегда истинна и окончательный результат зависит только от значения оставшейся формулы.

Пример Л6. Упростить формулу: $\neg(\neg x \wedge \neg y) \vee (x \rightarrow y) \wedge x$.

Решение. Проводим цепочку преобразований (в скобках указывается номер применяемого правила):

$$\neg(\neg x \wedge \neg y) \vee (x \rightarrow y) \wedge x \equiv (9) \equiv \neg(\neg x \wedge \neg y) \vee (\neg x \vee y) \wedge x \equiv (2) \equiv \neg \neg x \vee \neg \neg y \vee (\neg x \vee y) \wedge x \equiv (8, 11, 3) \equiv x \vee y \vee \neg x \wedge x \vee y \wedge x \equiv (20) \equiv x \vee y \vee y \wedge x \equiv (5) \equiv x \vee y. \blacksquare$$

Пример Л7. Определить, является ли формула $\neg(\neg x \wedge \neg y) \vee (x \rightarrow y) \wedge x$ противоречием?

Решение. Прделаав упрощения, приведенные в примере 1, получим, что исходная формула эквивалентна формуле $x \vee y$, которая противоречием не является. Ответ – нет.

Пример Л8. Определить, является формула $x \vee \neg((y \vee z) \wedge x)$ тавтологией?

Решение. Упростим формулу: $x \vee \neg((y \vee z) \wedge x) \equiv (2) \equiv x \vee \neg(y \vee z) \vee \neg x \equiv (12) \equiv x \vee \neg x \vee \neg(y \vee z) \equiv (19) \equiv x \vee \neg x$. Ответ – да. \blacksquare

3.6. Упражнения по теме для самостоятельного решения

1. Переведите целые числа из десятичной системы счисления в двоичную:
а) 513 (Ответ: 1000000001); б) 600 (Ответ: 1001011000).
2. Переведите десятичные дроби в двоичную систему счисления (ответ записать с шестью двоичными знаками):
а) 0,4622 (Ответ: 0,011101); б) 0,5198 (Ответ: 0,100001);
3. Переведите целые числа из десятичной в восьмеричную систему счисления: а) 8700 (Ответ: 20774); б) 8888 (Ответ: 21270).
4. Переведите целые числа из десятичной в шестнадцатеричную систему счисления: а) 266 (Ответ: 10A); б) 1023 (Ответ: 3FF).
5. Переведите двоичное число 1010001001011 в восьмеричную систему счисления (Ответ: 12113).
6. Вычислите сумму чисел x и y , если $x = 1110101_2$, $y = 1011011_2$. Результат представьте в виде восьмеричного числа. (Ответ: 320).
7. Вычислить значение суммы в десятичной системе счисления:
 $10_2 + 10_8 + 10_{16} = ?_{10}$ (Ответ: 26).
8. В шестнадцатеричной системе счисления сумма чисел F_{16} и 1011_2 равна? (Ответ: 1A)
9. Записать с помощью логических формул: если влажность так высока, то либо после полудня, либо вечером будет дождь. (Ответ: обозначим высказывания: A – «влажность высока», B – «дождь после полудня», C – «дождь вечером». Тогда формула: $A \rightarrow (B \vee C)$).
10. Проверить истинность формул: известно, что x имеет значение 1. Что можно сказать о значениях импликации: $\neg x \wedge y \rightarrow z$? (Ответ: импликация истинна независимо от значений y и z , т.к. ее посылка всегда ложна при $x=1$).
11. Противоречива ли формула: $P \rightarrow \neg P \vee P \wedge \neg P$? (Ответ: Нет: упростив формулу, получим формулу $\neg P$, которая противоречием не является).
12. Упростить формулу: $\neg(\neg x \wedge \neg y) \vee (x \rightarrow y) \wedge x$. (Ответ: $x \vee y$)
13. Не строя таблицы истинности доказать эквивалентность: $\neg(x \rightarrow y) \equiv x \wedge \neg y$ (Ответ: Упростив формулу слева от знака тождества, получим формулу $x \wedge \neg y$, что и требовалось).

Контрольное задание по системам счисления и логике.
(Задание выполнить на листах с подробным решением)

1. Переведите целые числа из десятичной системы счисления в двоичную:
а) 2304; б) 5001.
2. Переведите целые числа из десятичной в восьмеричную систему счисления:
а) 8900; б) 9300.
3. Переведите целые числа из десятичной в шестнадцатеричную систему счисления: а) 1280; б) 2041.
4. Переведите числа из десятичной системы счисления в восьмеричную:
а) 2936; б) 481.
5. Переведите двоичные числа в восьмеричную систему счисления: а) 1011001101111; б) 110001000100.
6. Вычислите сумму чисел x и y , если $x = 111010_2$, $y = 101011_2$. Результат представьте в виде восьмеричного числа и в виде шестнадцатеричного числа.
7. Вычислите значение суммы в десятичной системе счисления:
 $110_2 + 100_8 + 101_{16} = ?_{10}$.
8. В шестнадцатеричной системе счисления сумма чисел $1F_{16}$ и 1001_2 равна?
9. Записать с помощью формул: для того, чтобы число делилось на три, достаточно, чтобы оно делилось на девять.
10. Проверить истинность формулы $((x \rightarrow y) \wedge (y \rightarrow z)) \rightarrow (x \rightarrow z)$ при $x=0, y=1, z=0$ и при $x=1, y=1, z=0$.
11. Является ли тавтологией формула: $(P \vee Q) \rightarrow P \vee Q \wedge \neg Q$?
12. Упростить формулу: $(x \wedge \neg(x \wedge \neg x \rightarrow y \wedge \neg y) \rightarrow z) \vee x \vee y \wedge z \vee (y \wedge z)$.
13. Проверить эквивалентность формул, преобразовав формулы слева и справа от знака \equiv к одной и той же формуле: $(P \rightarrow Q) \wedge (P \rightarrow R) \equiv (P \rightarrow Q \wedge R)$.