

Министерство сельского хозяйства Российской Федерации
ФГБОУ ВО «Красноярский государственный аграрный университет»

*Т.П. Пушкарева, В.В. Калитина,
А.А. Брит, Н.В. Титовская*

ИНФОРМАТИКА

Рекомендовано Учебно-методическим советом федерального государственного бюджетного образовательного учреждения высшего образования «Красноярский государственный аграрный университет» для внутривузовского использования в качестве учебного пособия по укрупненной группе направлений подготовки 35.00.00 «Сельское хозяйство и сельскохозяйственные науки»

Электронное издание

Красноярск 2024

ББК 32.81я73

П 91

Рецензенты:

И.А. Панфилов, канд. техн. наук, доцент каф. системного анализа и исследования операций Сибирского государственного университета науки и технологий им. акад. М.Ф. Решетнева

Т.А. Степанова, канд. пед. наук, доцент каф. информатики и информационных технологий в образовании Красноярского государственного педагогического университета им. В.П. Астафьева

П 91 **Пушкарева, Т.П.**

Информатика [Электронный ресурс]: учебное пособие / Т.П. Пушкарева, В.В. Калитина, А.А. Брит, Н.В. Титовская; Краснояр. гос. аграр. ун-т. – Красноярск, 2024. – 183 с.

Учебное пособие представляет собой систематизацию теоретических знаний о сущности и специфике информации, а также наиболее целесообразных направлениях и методах использования современных компьютерных технологий в решении прикладных задач.

Предназначено для студентов для освоения курса дисциплин «Информатика», «Информационные технологии», обучающимися по различным направлениям и специальностям.

ББК 32.972

© Пушкарева Т.П., Калитина В.В.,
Брит А.А., Титовская Н.В., 2024
© ФГБОУ ВО «Красноярский государственный
аграрный университет», 2024

ВВЕДЕНИЕ

Стремительное развитие и внедрение информационных технологий во все сферы человеческой деятельности обуславливает необходимость в подготовке бакалавров и специалистов с развитым логическим мышлением, понимающих суть процессов вычислительной техники, способных к поиску и анализу больших объемов информации, а также к решению разнообразных задач, связанных с применением компьютерных технологий в своей учебной и будущей профессиональной деятельности.

Учебное пособие «Информатика» представляет собой систематизацию теоретических знаний о сущности и специфике информации, а также наиболее целесообразных направлениях и методах использования современных компьютерных технологий в решении прикладных задач.

Первая глава учебного пособия раскрывает сущность понятия «информация», описывает ее свойства и действия над ней; методы вычисления количества информации.

Учитывая, что сегодня востребован специалист, разбирающийся в процессах, происходящих в вычислительной технике, представляются достаточно важными такие темы, как архитектура персонального компьютера, его арифметические и логические основы, а также представление информации в персональном компьютере. Изучению этих тем посвящены вторая, третья, четвертая и пятая главы учебного пособия. Так, во второй главе описаны основные и дополнительные устройства персонального компьютера, в третьей и четвертой главах показано, как осуществляются арифметические и логические операции. В пятой главе описано представление различных видов информации в компьютерах, рассмотрено кодирование различных видов информации.

Шестая глава посвящена применению компьютерных технологий в решении учебных и профессиональных задач.

В пособии приведено достаточное количество иллюстраций, примеров и программ, которые помогут обучающимся решить любые возникшие перед ними проблемы.

В конце каждого параграфа приведены вопросы и задания в соответствии с содержанием учебной информации, они обеспечивают самоконтроль изученного материала и повышают уровень его понимания.

1 ДЕЙСТВИЯ НАД ИНФОРМАЦИЕЙ

1.1 Понятие информации

Сигналы и данные

Все, что нас окружает и с чем мы сталкиваемся ежедневно, относится либо к *физическим телам*, либо к *физическим полям*. В состоянии непрерывного движения и изменения, которое сопровождается обменом энергией и ее переходом из одной формы в другую, пребывают все физические объекты.

Все виды *энергообмена* сопровождаются появлением сигналов. В физических телах при взаимодействии с сигналами возникают определенные изменения свойств. Такое явление называется *регистрацией сигналов*. Изменения можно наблюдать, измерять или фиксировать другими способами.

Передается информация с помощью сигналов. *Сигнал* – физический процесс с информационным значением. Сигналы могут быть дискретными или непрерывными. Дискретный сигнал принимает конечное число значений. Он может передавать символическую и текстовую информацию. Непрерывный сигнал постоянно меняется по некоторой амплитуде во времени, используется в телевизионном вещании и радиовещании, телефонной связи и т.д.

Таким образом, при возникновении и регистрации новых сигналов образуются данные. Данные являются зарегистрированными сигналами.

Данные переносят *информацию* о событиях, которые произошли в материальном мире. Данные не тождественны информации. Станут ли данные информацией, зависит от очень многих обстоятельств.

Например, с помощью механического секундомера, наблюдая за состязаниями бегунов, мы регистрируем начальное и конечное положение стрелки прибора. Далее мы регистрируем данные, измеряя величину ее перемещения за время забега. Но информацию о времени прохождения дистанции мы пока не получаем. Необходимо использовать *метод* пересчета одной физической величины в другую, чтобы данные о перемещении стрелки дали информацию о времени прохождения забега. А для этого нужно знать единицу деления шкалы секундомера (или знать метод ее определения), а также, как умножается единица деления прибора на величину перемещения, то есть надо еще обладать математическим методом «умножение».

Понятие об информации

Строгого и общепризнанного определения информации до сих пор не существует. Большинство ученых и в наши дни считают, что информацию следует рассматривать как первичное, неопределимое понятие. Поэтому вместо определения обычно используют термин *понятие об информации*.

Понятия не даются однозначно, в отличие от определений, а вводятся на примерах. Каждая научная дисциплина делает это по-своему, определяя в качестве основных компонентов только те, которые наилучшим образом соответствуют ее предмету и задачам. При этом возможна ситуация, когда понятие об информации, введенное в рамках одной научной дисциплины, опровергается конкретными примерами и фактами, полученными в рамках другой науки.

Термин «информация» происходит от латинского слова *informatio*, что означает сведения, разъяснения, изложение.

Предлагается следующее понимание термина «информация».

Информация – это знания или сведения о ком-либо или о чем-либо, об их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

Информация – это сведения, которые можно собирать, хранить, передавать, обрабатывать, использовать.

Информация – это продукт взаимодействия данных и адекватных им методов.

Рассмотрим определение информации и обратим внимание на следующие обстоятельства.

1. *Динамический характер информации.* «Динамический» по определению – меняющийся, подвижный. Динамический характер информации заключается в том, что информация изменчива и как таковая существует в момент взаимодействия данных и методов. Таким образом, информация существует только в момент протекания *информационного процесса*. После окончания информационного процесса информация хранится в качестве данных.

Действия, выполняемые с информацией, называются *информационными процессами*. Выделяют следующие информационные процессы:

– процесс передачи информации, который включает в себя: ввод (сбор, получение) информации; вывод информации; передачу информации;

– процесс обработки (преобразования) информации;

– процесс хранения информации (в собственной памяти или на внешних носителях).

2. *Требование адекватности методов.* Одни и те же данные могут в момент потребления поставлять разную информацию в зависимости от степени адекватности взаимодействующих с ними методов. Например, для человека, не владеющего французским языком, письмо, полученное из Парижа, дает только ту информацию, которую можно получить методом наблюдения (количество страниц, цвет и сорт бумаги, наличие незнакомых символов и т.п.). Все это информация, но это не вся информация, заключенная в письме. Использование более адекватных методов даст иную информацию.

3. *Диалектический характер взаимодействия данных и методов.* Данные являются *объективными*, так как являются результатом регистрации объективно существовавших сигналов, вызванных изменениями в материальных телах или полях. А методы являются *субъективными*, так как в основе искусственных методов заложены упорядоченные последовательности команд (алгоритмы), которые составлены и подготовлены людьми (субъектами).

В основе естественных методов лежат биологические свойства субъектов информационного процесса. Таким образом, *информация возникает и существует в момент диалектического взаимодействия объективных данных и субъективных методов.*

Свойства информации

Информация является динамическим объектом, образующимся в момент взаимодействия объективных данных и субъективных методов. Как и всякий объект, она обладает свойствами (объекты различимы по своим свойствам). Характерной особенностью информации, отличающей ее от других объектов природы и общества, является то, что на свойства информации влияют как свойства данных, составляющих ее содержательную часть, так и свойства методов, взаимодействующих с данными в ходе информационного процесса. По окончании процесса свойства информации переносятся на свойства новых данных, то есть свойства методов могут переходить на свойства данных.

С точки зрения информатики наиболее важными представляются следующие свойства:

- объективность;
- полнота;
- достоверность;

- адекватность;
- доступность;
- актуальность информации.

Объективность и субъективность информации. Понятие объективности информации является относительным. Это понятно, если учесть, что методы являются субъективными.

Объективный – существующий вне и независимо от человеческого сознания. *Субъективный* – относящийся к субъекту, человеку, личности, определяемый субъектом.

Более объективной принято считать ту информацию, в которую методы вносят меньший субъективный элемент.

Объективную информацию можно получить с помощью исправно работающих датчиков, измерительных приборов. Информация перестает быть объективной, отражаясь в сознании конкретного человека, потому что преобразовывается в большей или меньшей степени в соответствии со мнением, суждениями, опытом, знаниями конкретного субъекта.

В ходе информационного процесса степень объективности информации всегда понижается. Это свойство учитывают, например, в правовых дисциплинах, где по-разному обрабатываются показания лиц, непосредственно наблюдавших события или получивших информацию косвенным путем (посредством умозаключений или со слов третьих лиц).

Полнота информации. Полнота информации во многом характеризует *качество информации* и определяет *достаточность* данных для принятия решений или для создания новых данных на основе имеющихся. Чем полнее данные, тем шире диапазон методов, которые можно использовать, тем проще подобрать метод, вносящий минимум погрешностей в ход информационного процесса. Неполная информация может привести к ошибочному выводу или решению.

Достоверность информации. Если информация отражает истинное положение дел, то она является достоверной. Объективная информация всегда достоверна.

Данные возникают в результате регистрации сигналов. Не все сигналы считаются «полезными». Всегда присутствует некоторый уровень посторонних сигналов, в результате чего «полезные» данные сопровождаются определенным уровнем «информационного шума».

Если полезный сигнал регистрируется более четко, чем посторонние сигналы, то достоверность информации может считаться бо-

лее высокой, при увеличении уровня шумов достоверность информации снижается. В таком случае для передачи того же количества информации требуется использовать либо большее количество данных, либо более сложные методы.

Достоверная информация помогает принять нам правильное решение.

Недостоверной информация может быть по следующим причинам:

– преднамеренное искажение или непреднамеренное искажение субъективных свойств;

– искажение в результате воздействия помех и недостаточно точных средств ее фиксации.

Адекватность информации – степень соответствия реальному объективному состоянию дела. Неадекватная информация может образовываться при создании новой информации на основе неполных или недостоверных данных. Применение неадекватных методов к полным и достоверным данным может приводить к созданию неадекватной информации.

Доступность информации – мера возможности получить ту или иную информацию. На степень доступности информации влияют одновременно как доступность данных, так и доступность адекватных методов для их интерпретации.

Актуальность информации – это степень соответствия информации текущему моменту времени. Нередко с актуальностью, как и с полнотой, связывают коммерческую ценность информации. Поскольку информационные процессы растянуты во времени, то достоверная и адекватная, но устаревшая информация может приводить к ошибочным решениям.

1.2 Операции с данными

Данные – составная часть информации. Они представляют собой зарегистрированные сигналы. При этом физический метод регистрации может быть любым: механическое перемещение физических тел, изменение их формы или параметров качества поверхности, электрических, магнитных, оптических характеристик, химического состава и (или) характера химических связей, состояния электронной системы и многое другое. В соответствии с методом регистрации данные могут храниться и транспортироваться на носителях различных видов.

Самым распространенным носителем данных является бумага. На бумаге данные регистрируются путем изменения оптических характеристик ее поверхности. В качестве носителей, использующих изменение магнитных свойств, можно назвать магнитные ленты и диски. Регистрация данных путем изменения химического состава поверхностных веществ носителя широко используется в фотографии. На биохимическом уровне происходит накопление и передача данных в живой природе.

Любой носитель можно характеризовать параметром *разрешающей способности* (количеством данных, записанных в принятой для носителя единице измерения) и *динамическим диапазоном* (логарифмическим отношением интенсивности амплитуд максимального и минимального регистрируемого сигналов). От этих свойств носителя нередко зависят такие свойства информации, как полнота, доступность и достоверность. Так, например, мы можем рассчитывать на то, что в базе данных, размещаемой на компакт-диске, проще обеспечить полноту информации, чем в аналогичной по назначению базе данных, размещенной на гибком магнитном диске, поскольку в первом случае плотность записи данных на единице длины дорожки намного выше. Для обычного потребителя доступность информации в книге заметно выше, чем той же информации на компакт-диске, поскольку не все потребители обладают необходимым оборудованием. И наконец, известно, что визуальный эффект от просмотра слайда в проекторе намного больше, чем от просмотра аналогичной иллюстрации, напечатанной на бумаге.

В ходе информационного процесса данные преобразуются из одного вида в другой с помощью методов. Обработка данных включает в себя множество различных операций.

В структуре возможных операций с данными можно выделить следующие основные.

1. *Сбор данных* – накопление информации с целью обеспечения *достаточной полноты* для принятия решений.

2. *Формализация данных* – приведение данных, поступающих из разных источников, к *одинаковой форме*, чтобы сделать их *сопоставимыми между собой*, то есть *повысить уровень их доступности*.

3. *Фильтрация данных* – отсеивание «лишних» данных, в которых нет необходимости для принятия решений; при этом должен уменьшаться уровень «шума», а *достоверность и адекватность данных должны возрастать*.

4. *Сортировка данных* – упорядочение данных по заданному признаку с целью удобства использования; *повышает доступность информации.*

5. *Архивация данных* – организация хранения данных в удобной и легкодоступной форме; служит для снижения экономических затрат по хранению данных и *повышает общую надежность информационного процесса в целом.*

6. *Защита данных* – комплекс мер, направленных на предотвращение утраты, воспроизведения и модификации данных.

7. *Транспортировка данных* – прием и передача (доставка и поставка) данных между удаленными участниками информационного процесса. При этом источник данных в информатике принято называть *сервером*, а потребителя – *клиентом*.

8. *Преобразование данных* – перевод данных из одной формы в другую или из одной структуры в другую. Преобразование данных часто связано с изменением типа носителя, например книги можно хранить в обычной бумажной форме, но можно использовать для этого и электронную форму, и микрофотопленку.

Приведенный список типовых операций с данными далеко не полон. Миллионы людей во всем мире занимаются созданием, обработкой, преобразованием и транспортировкой данных.

1.3 Измерение количества информации

Вероятностный подход к измерению информации

Основополагающая роль в *вероятностном* подходе принадлежит энтропии множества вероятностей, формула которой была получена в 1948 г. американским исследователем *К. Шенноном*. Предлагая для измерения количества информации свою знаменитую энтропийную меру, К. Шеннон руководствовался следующими соображениями.

Вероятность p – *количественная априорная* (т.е. известная до проведения опыта) характеристика одного из исходов (событий) некоторого опыта. Измеряется в пределах от 0 до 1. Если заранее известны все исходы опыта, сумма их вероятностей равна 1, а сами исходы составляют полную группу событий. Если все исходы могут свершиться с одинаковой долей вероятности, они называются *равновероятными*.

Пусть можно получить n сообщений по результатам некоторого опыта (т.е. у опыта есть n исходов), причем известны вероятности получения каждого сообщения (исхода) – p_i . Тогда, в соответствии с идеей Шеннона, количество информации I в сообщении i для равновероятных событий определяется по формуле

$$I = -\log_2 p_i,$$

где p_i – вероятность i -го сообщения (исхода).

Для вычисления количества информации в сообщении о *неравновероятном событии* используют следующую формулу:

$$I = \log_2(1/p),$$

где I – это количество информации; p – вероятность события.

Вероятность события выражается в долях единицы и вычисляется по формуле

$$P = K/N,$$

где K – величина, показывающая, сколько раз произошло интересующее нас событие; N – общее число возможных исходов какого-то процесса.

Пример 1. Определим количество информации, содержащейся в сообщении о результате сдачи экзамена для студента, имеющего оценки выше четверки.

Решение. Пусть $I(j)$ – количество информации в сообщении о получении оценки j . В соответствии с формулой Шеннона имеем:

$$I(5) = -\log_2 0,5 = 1,$$

$$I(4) = -\log_2 0,3 = 1,74,$$

$$I(3) = -\log_2 0,1 = 3,32,$$

$$I(2) = -\log_2 0,1 = 3,32.$$

Пример 2. Определим количество информации, содержащейся в сообщении о результате сдачи экзамена для неуспевающего студента.

Решение:

$$I(5) = -\log_2 0,1 = 3,32,$$

$$I(4) = -\log_2 0,2 = 2,32,$$

$$I(3) = -\log_2 0,4 = 1,32,$$

$$I(2) = -\log_2 0,3 = 1,74.$$

Пример 3. На столе лежали 24 тетради в клетку и 8 тетрадей в линейку. Студент выбрал одну тетрадь. Вычислим вероятность выбора тетради с разной разлиновкой и количество информации, которое при этом было получено.

Решение. Обозначим:

K_1 – количество тетрадей в клетку, $K_1=24$;

K_2 – количество тетрадей в линейку, $K_2 = 8$;

N – общее количество тетрадей, $N = K_1 + K_2 = 24 + 8 = 32$.

Вероятность выбора тетради в клетку: $p_1 = 24/32 = 3/4 = 0,75$.

Вероятность выбора тетради в линейку: $p_2 = 8/32 = 1/4 = 0,25$.

Обратим внимание на то, что в сумме все вероятности дают 1.

Вычислим количество информации, содержащееся в сообщении, что студент выбрал тетрадь в клетку:

$$I_1 = \log_2(1/p_1) = \log_2(1/0,75) = \log_2 1,3 = 1,15470 \text{ бит.}$$

Вычислим количество информации, содержащееся в сообщении, если была выбрана тетрадь в линейку:

$$I_2 = \log_2(1/p_2) = \log_2(1/0,25) = \log_2 4 = 2 \text{ бит.}$$

Таким образом, количество получаемой с сообщением информации тем больше, чем неожиданнее данное сообщение.

Содержательный подход к измерению информации

Информация – это знания человека. Отсюда следует вывод, что сообщение информативно (содержит ненулевую информацию), если оно пополняет знания человека. Например, прогноз погоды на завтра – информативное сообщение, а сообщение о вчерашней погоде неинформативно: нам это уже известно.

Нетрудно понять, что информативность одного и того же сообщения может быть разной для разных людей. Например, $2 \times 2 = 4$ информативно для первоклассника, изучающего таблицу умножения, и неинформативно для старшеклассника. Отсюда следует вывод, что сообщение информативно для человека, если оно содержит новые сведения, и неинформативно, если сведения старые, известные.

Однако в том случае, когда текст содержит новые сведения, но он непонятен, эти сведения будут неинформативны. Быть понятным – значит быть логически связанным с предыдущими знаниями человека. Для того чтобы понять данное определение, нужно изучить элементарную математику и знать начала высшей.

Получение всяких знаний должно идти от простого к сложному. И тогда каждое новое сообщение будет понятным, а значит, будет нести информацию для человека.

Сообщение несет информацию для человека, если содержащиеся в нем сведения являются для него новыми и понятными.

Неопределенность знаний и единица информации

Единица измерения информации была определена в науке, которая называется теорией информации. Эта единица называется «бит».

Ее определение звучит так: сообщение, уменьшающее неопределенность знаний в два раза, несет 1 бит информации.

Рассмотрим, что такое «неопределенность знаний», на примерах. Бросается монета. Есть всего два варианта возможного результата ее бросания. Причем ни один из этих вариантов не имеет преимуществ перед другим, т.е. они равновероятны.

В этом случае перед подбрасыванием монеты неопределенность знаний о результате равна двум.

Игральный кубик с шестью гранями может с равной вероятностью упасть на любую из них. Значит, неопределенность знаний о результате бросания кубика равна шести.

Еще пример: спортсмены-лыжники перед забегом путем жеребьевки определяют свой порядковый номер на старте. Допустим, неопределенность знаний спортсменом своего номера до жеребьевки равна ста.

Следовательно, можно сделать вывод: неопределенность знаний о некотором событии – это количество возможных результатов события (бросания монеты, кубика; вытаскивания жребия).

Вернемся к примеру с монетой. После того как брошена монета, получено зрительное сообщение, что выпал, например, орел. Произошло одно из двух возможных событий. Неопределенность знаний уменьшилась в два раза: было два варианта, остался один. Значит, узнав результат бросания монеты, мы получили 1 бит информации.

Сообщение о том, что произошло одно событие из двух равновероятных, несет 1 бит информации.

Рассмотрим задачу: студент на экзамене может получить одну из четырех оценок: «5» – «отлично», «4» – «хорошо», «3» – «удовлетворительно», «2» – «неудовлетворительно». Если студент учится очень «неровно», он может с одинаковой вероятностью получить любую оценку – от «2» до «5». Когда он вернулся, на вопрос: «Что получил? – ответил: «Четыре!».

Сколько бит информации содержится в его ответе?

Если сразу сложно ответить на этот вопрос, то к ответу можно подойти постепенно. Чтобы отгадать оценку, будем задавать вопросы, на которые можно ответить только «да» или «нет».

Вопросы следует ставить так, чтобы каждый ответ уменьшал количество вариантов в два раза и, следовательно, приносил 1 бит информации.

Первый вопрос:

Оценка выше тройки?

Ответ: Да!

После этого ответа число вариантов уменьшилось в два раза. Остались только «4» и «5». Получен 1 бит информации.

Второй вопрос:

Ты получил пять?

Ответ: Нет!

Выбран один вариант из двух оставшихся: оценка – «четыре». Получен еще 1 бит информации. В сумме имеем 2 бита. Сообщение о том, что произошло одно из четырех равновероятных событий, несет 2 бита информации.

Метод поиска, на каждом шаге которого отбрасывается половина вариантов, называется *методом половинного деления*.

Решим еще одну частную задачу, применив этот метод, а потом выведем общее правило.

На книжном стеллаже восемь полок. Книга может быть поставлена на любую из них. Сколько информации содержит сообщение о том, где находится книга?

Задаем вопросы:

– Книга лежит выше четвертой полки?

– Нет.

– Книга лежит ниже третьей полки?

– Да.

– Книга – на второй полке?

– Нет.

Теперь все ясно! Книга лежит на первой полке!

Каждый ответ уменьшал неопределенность в два раза. Всего было задано три вопроса. Значит набрано 3 бита информации. И если бы сразу было сказано, что книга лежит на первой полке, то этим сообщением были бы переданы те же 3 бита информации.

Теперь попробуем получить формулу, по которой вычисляется количество информации, содержащейся в сообщении о том, что произошло одно из множества равновероятных событий.

Обозначим буквой N количество возможных событий или неопределенность знаний. Буквой i будем обозначать количество информации в сообщении о том, что произошло одно из N событий.

В примере с монетой:

$$N = 2, i = 1.$$

В примере с оценками:

$$N = 4, i = 2.$$

В примере со стеллажом:

$$N = 8, i = 3.$$

Нетрудно заметить, что связь между этими величинами выражается формулой:

$$2^i = N.$$

Действительно: $2^1 = 2$; $2^2 = 4$; $2^3 = 8$.

Если величина N известна, то для определения i необходимо решить написанное выше показательное уравнение.

Например, пусть на стеллаже не 8, а 16 полок. Чтобы ответить на вопрос, сколько информации содержится в сообщении о том, где лежит книга, нужно решить уравнение:

$$2^i = 16.$$

Поскольку $16 = 2^4$, то $i = 4$.

Количество информации i , содержащееся в сообщении о том, что произошло одно из N равновероятных событий, определяется из решения показательного уравнения: $2^i = N$.

Если значение N равно целой степени числа 2 (4, 8, 16, 32, 64 и т.д.), то такое уравнение решается просто: i будет целым числом.

А чему равно количество информации в сообщении о результате бросания игральной кости, у которой имеется шесть граней и, следовательно, $N = 6$?

Решение уравнения

$$2^i = 6$$

будет дробным числом, лежащим между числами 2 и 3, поскольку $2^2 = 4$, а $2^3 = 8$. С точностью до пяти знаков после запятой решение такое: 2,58496.

Алфавитный подход к измерению информации

Способ, не связывающий количество информации с содержанием сообщения, называется *алфавитным подходом*.

Все множество используемых в языке символов называют *алфавитом*. Обычно под алфавитом понимают только буквы, но поскольку в тексте могут встречаться знаки препинания, цифры, скобки, то мы их тоже включим в алфавит. В алфавит также следует включить и пробел (промежуток между словами).

Полное число символов алфавита называется *мощностью алфавита*. Обозначим эту величину буквой N . Например, мощность алфавита из русских букв и дополнительных символов равна 54.

Допустим, что текст поступает последовательно, по одному знаку. Предположим, что каждый появляющийся символ с одинаковой вероятностью может быть любым из N символов алфавита.

Каждый символ несет i бит информации, число i можно определить из уравнения:

$$2^i = N.$$

Для $N = 54$, используя таблицу, получаем:

$$i = 5,755 \text{ бит.}$$

Для того чтобы найти количество информации во всем тексте, нужно посчитать число символов в нем и умножить на i .

Таким образом, при алфавитном подходе к измерению информации количество информации зависит не от содержания, а от объема текста (то есть от числа знаков в тексте) и от мощности алфавита.

Отсюда следует, например, что нельзя сравнивать информационные объемы текстов, написанных на разных языках, только по объему. У них отличаются информационные веса одного символа, так как мощности алфавитов разных языков – различные.

Но если книги написаны на одном языке, то понятно, что в толстой книге информации больше, чем в тонкой. При этом содержательная сторона книги в расчет не берется.

Сформулируем правило, как измерить информацию, используя для этого алфавитный подход.

Количество информации, содержащееся в символьном сообщении, равно $K \times i$, где K – число символов в тексте сообщения, а i – информационный вес символа, который находится из уравнения

$$2^i = N,$$

где N – мощность используемого алфавита.

Применение алфавитного подхода удобно прежде всего при использовании технических средств работы с информацией. В этом случае теряют смысл понятия «новые – старые», «понятные – непонятные» сведения. Алфавитный подход является объективным способом измерения информации в отличие от субъективного, содержательного, подхода.

В заключение приведем основные единицы измерения информации:

1 килобайт (<i>Кбайт</i>)	= 1024 байт	= 2^{10} байт;
1 мегабайт (<i>Мбайт</i>)	= 1024 Кбайт	= 2^{20} байт;
1 гигабайт (<i>Гбайт</i>)	= 1024 Мбайт	= 2^{30} байт;

$$\begin{aligned}
 1 \text{ терабайт (Тбайт)} &= 1024 \text{ Гбайт} = 2^{40} \text{ байт}; \\
 1 \text{ петабайт (Пбайт)} &= 1024 \text{ Тбайт} = 2^{50} \text{ байт}.
 \end{aligned}$$

Вопросы и задания для самоконтроля

1. Какие подходы к измерению информации вам известны?
2. Какова основная единица измерения информации?
3. Сколько байт содержит 1 Кб информации?
4. Приведите формулу подсчета количества информации при уменьшении неопределенности знания.
5. Как подсчитать количество информации, передаваемое в символьном сообщении?
6. Вариант текста в среднем имеет объем 20 килобайт (на каждой странице текста 40 строк по 64 символа в каждой, 1 символ занимает 8 бит), чему равно количество страниц в тексте?
7. Сведения о сотруднике хранятся в виде строки из 2048 символов. Сведения обо всех 8192 сотрудниках можно разместить на дисках емкостью 1,2 Мб. Сколько таких дисков потребуется?
8. Информационная емкость человеческой яйцеклетки приблизительно равна 2^{33} бит. Каково минимальное количество винчестеров (по 20 Мб), на которых можно уместить генетическую информацию одного человека?
9. Чему равно максимальное количество страниц книги (32 строки по 64 символа, 1 символ занимает 8 бит), которые поместятся в файле объемом 640 Кбайт?
10. Чему равно наибольшее натуральное число, кодируемое 16 битами?
11. Область экрана имеет размеры 512×128 точек и каждая точка может иметь один из 256 оттенков. Чему равен минимальный объем памяти (Кбайт), необходимый для хранения этой области?
12. Сообщение «школьники изучали Basic» несет 4 бита информации. Вероятность изучения Pascal в 8 раз меньше. Чему равно количество бит информации в сообщении «школьники изучали Pascal»?
13. Два сообщения содержат одинаковое количество символов. Количество информации в первом тексте в 1,5 раза больше, чем во втором. Сколько символов содержат алфавиты, с помощью которых записаны сообщения, если известно, что число символов в каждом алфавите не превышает 10 и на каждый символ приходится целое число битов?

2 АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

2.1 Магистрально-модульный принцип построения компьютера

Персональный компьютер (ПК) – это программируемое электронное устройство, предназначенное для обработки и хранения (накопления) информации. Современные ПК используются для автоматизации отдельных рабочих мест, обработки деловой информации, обучения и т.д. Все ПК имеют общую принципиальную схему (архитектуру).

Архитектура ПК – совокупность аппаратных и программных средств, с помощью которых обеспечивается выполнение задач пользователя и программирование задач.

Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера.

Архитектура включает:

- описание пользовательских возможностей программирования;
- описание системы команд и системы адресации;
- организацию памяти и т.д.

В основу архитектуры ПК положен *модульно-магистральный принцип* (рис. 2.1). Модульный принцип позволяет комплектовать нужную конфигурацию, модернизировать ее. Модульная организация опирается на магистральный (шинный) принцип обмена информацией.

Магистраль или системная шина – это аппаратная реализация стандартов взаимодействия различных узлов. Ее разрядность во многом определяет производительность компьютера, поскольку она связывает между собой процессор, оперативное запоминающее устройство (ОЗУ) и другие модули ПК. Существуют различные стандарты системной шины, которые сложились по мере развития техники: MCA, VESA, EISA, PCI-Express и SCSI.

Упрощенно системную шину можно представить как группу кабелей и электрических (токопроводящих) линий на материнской плате.

Системная шина включает в себя:

- шину данных,
- шину адреса,
- шину управления.

Обмен информацией между устройствами производится по этим трем многоуровневым шинам (многопроводные линии связи).

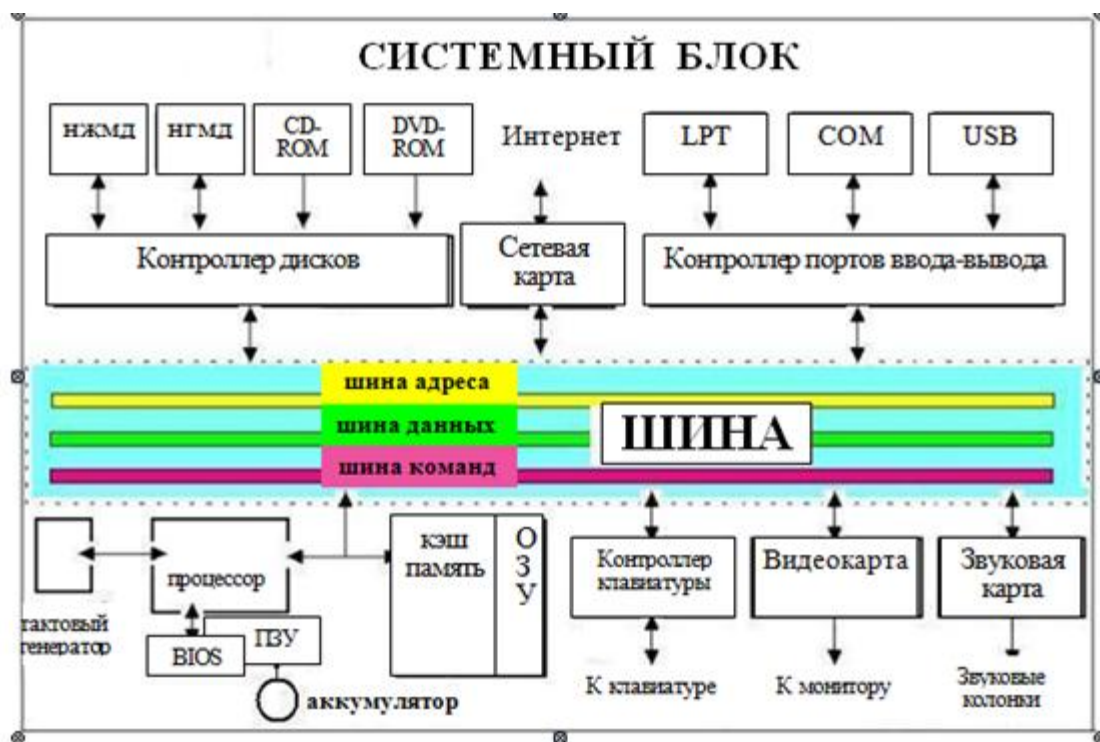


Рисунок 2.1 – Архитектура ПК

Адресная шина. Адресная шина предназначена для передачи по ней адреса того устройства (или той ячейки памяти), к которому обращается процессор. Адрес на шину выдает процессор. Адресная шина является однонаправленной, так как сигналы по ней всегда идут от процессора к устройству или оперативной памяти.

Шина данных. По этой шине происходит копирование данных. Шина является двунаправленной, так как данные перемещаются в двух направлениях в процессор, для обработки, и обратно.

Шина управления. Для того чтобы процессор мог управлять устройствами компьютера, ему нужны команды. Эти команды передаются по шине управления устройствам. Шина команд является однонаправленной.

Подключение отдельных модулей ПК к магистрали на физическом уровне осуществляется с помощью контроллеров, адаптеров, а на программном обеспечивается драйверами. Их совокупность называется *интерфейсом*.

Принцип открытой архитектуры – это возможность постоянного усовершенствования компьютера IBM PC в целом и его отдельных частей с использованием новых устройств, которые полностью совместимы друг с другом независимо от фирмы-изготовителя. Это дает наибольшую выгоду пользователям, которые могут расширять

возможности своих машин, покупая новые устройства и вставляя их в свободные разъемы (слоты) на системной (материнской) плате.

Принципы построения компьютеров

В основу построения подавляющего большинства компьютеров положены общие принципы, сформулированные в 1945 г. американским ученым Джоном фон Нейманом.

Принцип программного управления: программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности. Таким образом, процессор исполняет программу автоматически, без вмешательства человека.

Принцип однородности памяти: программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти – число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Принцип адресности: структурно основная память состоит из пронумерованных ячеек, процессору в произвольный момент времени доступна любая ячейка.

Отсюда следует возможность давать имена областям памяти, так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Компьютеры, построенные на этих принципах, относятся к типу фон-неймановских.

У персональных компьютеров выделяют 2 части: *аппаратную часть* – Hardware и *программное обеспечение* – Software.

2.2 Основные устройства персонального компьютера

К основным устройствам персонального компьютера относятся (стандартная конфигурация ПК):

- системный блок,
- монитор,
- клавиатура,
- мышь.

Персональные компьютеры, как было сказано ранее, проектируются на основе принципа открытой архитектуры. Регламентируются и стандартизируются только описание принципа действия компьютера и его конфигурация (определенная совокупность аппаратных

средств и соединений между ними). Следовательно, компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.

Компьютер легко расширяется и модернизируется за счет наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.

Для того чтобы соединить друг с другом различные устройства компьютера, они должны иметь одинаковый интерфейс (англ. Interface от inter – между, и face – лицо).

Интерфейс – это 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*.

Каждый из функциональных элементов (память, монитор или другое устройство) связан с шиной определенного типа – адресной, управляющей или шиной данных.

Для согласования интерфейсов периферийные устройства подключаются к шине не напрямую, а через свои контроллеры (адаптеры) и порты примерно по такой схеме (рис. 2.2)

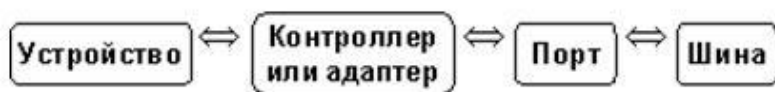


Рисунок 2.2 – Схема подключения периферийных устройств

Контроллеры и адаптеры представляют собой наборы электронных цепей, которыми снабжаются устройства компьютера с целью совместимости их интерфейсов. Контроллеры, кроме этого, осуществляют непосредственное управление периферийными устройствами по запросам микропроцессора.

Рассмотрим основные устройства персонального компьютера.

Системный блок представляет собой основной узел, внутри которого установлены наиболее важные компоненты компьютера.

По внешнему виду системные блоки различаются формой корпуса (горизонтальное и вертикальное исполнение) (рис. 2.3).



Рисунок 2.3 – Виды системного блока

В системном блоке содержатся следующие основные устройства:

- материнская плата;
- процессор;
- память;
- жесткий диск;
- блок питания;
- материнская плата.

Материнская (или системная) плата – важнейший элемент компьютера. Она служит для соединения всех компонентов системного блока, а также устройств, подключаемых к ПК, в единое целое (рис. 2.4).



Рисунок 2.4 – Системная плата компьютера класса Pentium

Именно к материнской плате подключаются процессор, жесткий диск, дисководы, видеокарта, монитор, клавиатура, мышь, принтер, модем и прочее.

Для этого на плате располагаются разъемы (или слоты), одни из которых имеют выход наружу, а другие нет. На задней стенке системного блока можно увидеть разъемы, которые выходят из компьютера и предназначены для подключения внешних устройств (клавиатуры, мыши, принтера, монитора и др.).

Материнская плата часто содержит в себе множество встроенных устройств.

В материнскую плату могут быть встроены видео- и звуковая карты (устройства для визуализации информации и обработки звука соответственно). Встроенное оборудование можно отключать, заменяя его более производительными устройствами.

Электронные компоненты при работе выделяют много тепла, поэтому очень важно их охлаждать. Для этого может применяться специальная система охлаждения, состоящая из небольших радиаторов или трубок. Помимо этого, устройства, устанавливаемые на материнскую плату, зачастую снабжаются собственными системами охлаждения: кулерами, радиаторами и др.

Главная информационная магистраль (шина) связывает их воедино.

Процессор

Процессор – это основная микросхема ЭВМ, выполняющая логические и арифметические операции, и осуществляющее управление всеми компонентами ПК. Процессор представляет собой миниатюрную тонкую кремниевую пластинку прямоугольной формы, на которой размещается огромное количество транзисторов, реализующих все функции, выполняемые процессором. Кремневая пластинка – очень хрупкая, а так как ее любое повреждение приведет к выходу из строя процессора, то она помещается в пластиковый или керамический корпус (рис. 2.5).

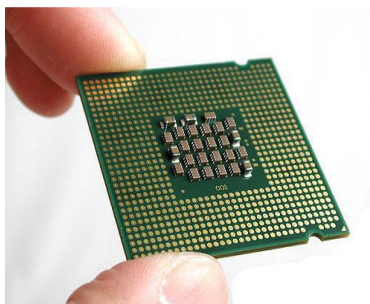


Рисунок 2.5 – Процессор

Конструктивно процессор состоит из ячеек, в которых данные могут не только храниться, но и изменяться. Внутренние ячейки процессора называют *регистрами*. Данные, попавшие в некоторые регистры, рассматриваются не как данные, а как команды, управляющие обработкой данных в других регистрах. Среди регистров процессора есть и такие, которые в зависимости от своего содержания способны модифицировать исполнение команд. Таким образом, управляя засылкой данных в разные регистры процессора, можно управлять обработкой данных. На этом и основано исполнение программ.

Сегодняшний процессор – это целая система важных устройств.

К основным устройствам процессора относят

Арифметико-логическое устройство – часть процессора, выполняющая машинные команды.

Устройство управления – часть процессора, выполняющая функции управления устройствами компьютера.

Шины данных и шины адресов(на физическом уровне) – многопроводные линии с гнездами для подключения электронных схем. Совокупность проводов магистрали разделяется на отдельные группы: шину адреса, шины данных и шину управления. Шина адреса предназначена для передачи адреса того устройства (или той ячейки памяти), к которому обращается процессор. По шине данных передается вся информация при записи и считывании. По шине управления передается управляющий сигнал. Процесс взаимодействия процессора и памяти сводится к двум операциям – записи и считывания информации. При записи процессор по специальным проводникам (шина адреса) передает биты, кодирующие адрес, по другим проводникам – управляющий сигнал «запись», и еще по другой группе проводников (шины данных) передает записываемую информацию. При чтении по шине адреса передается соответствующий адрес оперативной памяти (ОП), а с шины данных считывается нужная информация.

Регистры – ячейки памяти, которые служат для кратковременного хранения и преобразования данных и команд. На физическом уровне регистр – совокупность триггеров, способных хранить один двоичный разряд и связанных между собой общей системой управления.

Счетчик команд – регистр управляющего устройства компьютера, содержимое которого соответствует адресу очередной выполняемой команды. Счетчик команд служит для автоматической выборки программы из последовательных ячеек памяти.

Сопроцессор – специальный блок для операций с «плавающей точкой». Применяется для особо точных и сложных вычислений, а также для работы с рядом графических программ.

Кэш-память (буферная память) – сверхбыстрая память, предназначенная для хранения промежуточных результатов. Для того чтобы уменьшить количество обращений к оперативной памяти, внутри процессора создают буферную область – кэш-память. Пока процессор выполняет какие-то действия, она считывает заранее заказанную информацию из ОЗУ, а затем быстро передает ее процессору. С введением кэш-памяти сократились вынужденные простои процессора и увеличилось его быстродействие (более подробно кэш-память описана ниже). В современных процессорах используется три типа кэш-памяти.

Кэш-память первого уровня $L1$ – это блок высокоскоростной памяти, расположенный прямо на ядре процессора. В него копируются данные, извлеченные из оперативной памяти.

Кэш-память второго уровня $L2$ – это блок высокоскоростной памяти, выполняющий те же функции, что и кэш $L1$, однако имеющий более низкую скорость и больший объем.

Интегрированная кэш-память третьего уровня $L3$ в сочетании с быстрой системной шиной формирует высокоскоростной канал обмена данными с системной памятью. Как правило, кэш-памятью третьего уровня комплектуются только топовые процессоры и серверные решения.

Обмен данными внутри процессора происходит в несколько раз быстрее, чем, например, с оперативной памятью.

К основным параметрам процессора относятся

Разрядность процессора – число бит данных, которое он может принять и обработать в своих регистрах за один раз (за один такт). Современные процессоры семейства Intel Pentium работают с 64-разрядной шиной данных (разрядность процессора определяется не разрядностью шины данных, а разрядностью командной шины).

Рабочая тактовая частота – величина, показывающая, сколько инструкций способен выполнить процессор в течение 1 секунды. Тактовая частота обозначается цифрой в названии процессора (например, Pentium 4 3600, процессор Pentium 4 с тактовой частотой 3600 МГц).

Коэффициент внутреннего умножения тактовой частоты. Тактовые сигналы процессор получает от материнской платы, кото-

рая, в отличие от процессора, представляет собой не кристалл кремния, а большой набор проводников и микросхем. По чисто физическим причинам материнская плата не может работать со столь высокими частотами, как процессор. Для получения более высоких частот в процессоре происходит внутреннее умножение частоты на коэффициент 3; 3,5; 4; 4,5; 5 и более.

Объем кэш-памяти. Чем больше размер кэш-памяти, тем выше производительность процессора.

Память компьютера

Различают два основных вида памяти – *внутреннюю* и *внешнюю*.

Внешняя память – это память, реализованная в виде внешних (относительно материнской платы) запоминающих устройств с разными принципами хранения информации.

Внутренняя память – это запоминающее устройство, напрямую связанное с процессором и предназначенное для хранения выполняемых программ и данных, непосредственно участвующих в вычислениях. Обращение к внутренней памяти ПК осуществляется с высоким быстродействием, но она имеет ограниченный объем, определяемый системой адресации машины.

Обычно под компьютерной памятью понимают только внутреннюю память компьютера, которая в свою очередь классифицируется:

- *на оперативную* (ОЗУ – оперативное запоминающее устройство);
- *постоянную* (ПЗУ – постоянное запоминающее устройство).

Оперативная память (англ. *Random Access Memory, RAM*, память с произвольным доступом) – энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором.

Конструкция оперативной памяти выполнена таким образом, что данные в ней сохраняются только, пока на нее подается напряжение, поэтому она является энергозависимой памятью. Выключение компьютера, перезагрузка очищают оперативную память, и все данные, находящиеся в ней в этот момент, удаляются. Даже кратковременный перебой в подаче напряжения на планки памяти способен обнулить их или вызвать повреждение отдельной части информации.

Оперативная память компьютера состоит из ячеек, каждая из которых предназначена для хранения одного бита информации. Биты объединены в группы по 8 битов, которые называются байтами (едини-

цы измерения памяти совпадают с единицами измерения информации). Все байты пронумерованы. Номер байта называется его *адресом*.

Ячейки представляют собой конденсаторы, способные накапливать электрический заряд. Конденсатор хранит один бит данных, то есть либо логическую единицу (когда он заряжен), либо логический ноль (когда он разряжен).

ОЗУ может изготавливаться как отдельный внешний модуль или располагаться на одном кристалле с процессором, например в однокристалльных ЭВМ или однокристалльных микроконтроллерах.

Постоянное запоминающее устройство (ПЗУ) представляет собой интегральную микросхему, запрограммированную при изготовлении определенными данными, и предназначено для хранения постоянной программной и справочной информации. Информацию, хранящуюся в ПЗУ, можно только считывать, но не изменять.

Блок ПЗУ состоит из двух частей:

BIOS (Basic Input/Output System) – базовая система ввода и вывода. В ней хранится постоянная информация, заложенная на заводе-изготовителе;

CMOS (Complementary Metal-Oxide-Semiconductor) – переменная часть ПЗУ, где хранится информация о конфигурации ПК (перечень устройств, входящих в комплект ПК, и их характеристики).

В ПЗУ находятся:

- программа управления работой процессора;
- программа запуска и останова компьютера;
- программы тестирования устройств, проверяющие при каждом включении компьютера правильность работы его блоков;
- программы управления дисплеем, клавиатурой, принтером, внешней памятью;
- информация о том, где на диске находится операционная система.

После выключения питания компьютера информация в ПЗУ сохраняется за счет энергии от специальных автономных батарей. Таким образом, ПЗУ является энергонезависимой памятью.

Жесткий диск

Жесткий диск (*винчестер*) – запоминающее устройство (устройство хранения информации) произвольного доступа, основанное на принципе магнитной записи (рис. 2.6).



Рисунок 2.6 – Винчестер

Он является основным накопителем данных в большинстве компьютеров. Основа винчестера – блок металлических дисков, покрытых специальным веществом, которое способно хорошо хранить воздействие магнитного поля (например, оксидом железа). Современные винчестеры содержат от одного до трех таких дисков. Диски имеют отличную балансировку и очень ровную поверхность, так как скорости вращения довольно велики (стандартные скорости – 7200 или 10000 об/мин), а точность позиционирования головок должна быть высокой.

Для записи на диск используются специальные магнитные головки (обычно по две на каждый диск – с обеих сторон диска), способные формировать магнитное поле под воздействием токовых импульсов. Такая магнитная головка намагничивает участок диска магнитным моментом определенной направленности (логический «ноль» или логическая «единица», в зависимости от направленности магнитного момента). Процесс намагничивания происходит путем подачи токового импульса в определенный момент времени, когда магнитная головка позиционирована в нужном месте.

Для чтения с диска используются специальные магниторезистивные головки, которые реагируют на изменение магнитного поля путем изменения силы тока, возбуждаемого в головке. Такой аналоговый сигнал считывается, преобразуется в цифровую форму и подается в компьютерную систему.

Информация на диске хранится в виде концентрических окружностей. Блок головок является одним целым, и все головки перемещаются одновременно. Каждая головка обслуживает одну сторону одного диска. В любой момент времени все головки находятся над одной и той же дорожкой, но над разными дисками. В вертикальной плоскости такая совокупность дорожек образует *цилиндр*.

Дорожка делится на *сектора*. Каждый сектор хранит 512 байт информации. Сектор – минимальный элемент дискового пространства.

Внешняя память

Внешняя память – это память, предназначенная для длительного хранения программ и данных. Целостность содержимого внешней памяти не зависит от того, включен или выключен компьютер.

Дисковод (накопитель) – устройство записи или считывания информации. Накопители имеют собственное имя – буква латинского алфавита, за которой следует двоеточие. Для подключения к компьютеру одного или несколько дисководов и управления их работой нужен *дисковый контроллер*.

Носитель информации (носитель записи) – материальный объект, способный хранить информацию. Информация записывается на носитель посредством изменения физических, химических и механических свойств запоминающей среды.

По типу доступа к информации внешнюю память делят на два класса:

устройства прямого (произвольного) доступа – время обращения к информации не зависит от места ее расположения на носителе;

устройство последовательного доступа, когда в памяти производится последовательный просмотр участков носителя информации, пока нужный участок носителя не займет некоторое исходное положение. Характерным примером является ЗУ на магнитных лентах. Время доступа может в неблагоприятных случаях расположения информации достигать нескольких минут.

В современных компьютерах в состав внешней памяти входят:

- накопители на жестких магнитных дисках, НЖМД (винчестеры);
- накопители на магнитооптических компакт-дисках;
- накопители на оптических дисках (CD-ROM);
- флеш-накопители (Flash Drive);
- SSD (Solid-State Drive) – твердотельные накопители.

НЖМД, как было описано выше, предназначены для хранения той информации, которая наиболее часто используется в работе –

программ операционной системы, компиляторов, сервисных программ, прикладных программ пользователя, текстовых документов, файлов базы данных.

Это носители с произвольным доступом к информации.

Для обращения к НЖМД используется имя, задаваемое прописной латинской буквой, начиная с C:, но с помощью специальной системной программы можно разбить свой физический ЖД на несколько логических дисков, каждому из которых дается соответствующее имя.

Оптические (лазерные) CD и DVD диски предназначены для хранения любого вида информации. Информация на CD (Compact Disk) записывается с помощью лазерного луча. Лазерные диски относятся к носителям прямого (произвольного) доступа к информации

CD изготавливают из органических материалов с напылением на поверхность тонкого алюминиевого слоя. Лазерный диск имеет одну дорожку в виде спирали. Информация записывается отдельными секторами мощным лазерным лучом, выжигаящим на поверхности диска углубления, и представляет собой чередование впадин и выпуклостей. При считывании информации выступы отражают свет слабого лазерного луча и воспринимаются как «1», впадины поглощают луч и воспринимаются как «0». Это бесконтактный способ считывания информации. Срок хранения – 50–100 лет.

DVD – Digital Video Disk. Имеет те же размеры, что и CD. Может быть односторонним или двухсторонним, а на каждой стороне может быть один или два рабочих слоя.

USB флеш-накопитель является устройством хранения данных, в основе которого лежит флеш-память и USB интерфейс для физического соединения с компьютером или другим устройством.

Под корпусом флеш-накопителя скрывается небольшая печатная плата, на которой размещаются интегральные схемы и схемы питания.

SSD – твердотельные накопители– компьютерное немеханическое запоминающее устройство на основе микросхем памяти (рис. 2.7).



Рисунок 2.7 – Современный 2,5" SSD-накопитель

Кроме них SSD содержит управляющий контроллер. В настоящее время твердотельные накопители используются не только в компактных устройствах – ноутбуках, нетбуках, коммуникаторах и смартфонах, планшетах, но могут быть использованы и в стационарных компьютерах для повышения производительности.

По сравнению с традиционными жесткими дисками (HDD) твердотельные накопители имеют меньший размер и вес, но в несколько раз (6–7) большую стоимость за гигабайт и значительно меньшую износостойкость (ресурс записи).

Небольшие твердотельные накопители могут встраиваться в один корпус с магнитными жесткими дисками, образуя гибридные жесткие диски (SSHD, Solid-state hybrid drive). Подобное объединение позволяет воспользоваться частью преимуществ флеш-памяти (быстрый произвольный доступ) при сохранении небольшой стоимости хранения больших объемов данных.

Монитор

Монитор (дисплей) – устройство, предназначенное для вывода на экран текстовой и графической информации (рис. 2.8).



Рисунок 2.8 – Монитор

Современный монитор состоит из экрана (дисплея), блока питания, плат управления и корпуса. Информация для отображения на мониторе поступает с электронного устройства, формирующего видеосигнал (в компьютере – видеокарта). В некоторых случаях в качестве монитора может применяться и телевизор.

Мониторы имеют стандартный размер диагонали в 17, 19, 20 и 21 дюйм и выше. От качества монитора зависит сохранность зрения и общеутомляемость при работе. Наиболее распространенными сегодня являются LCD (жидкокристаллический дисплей) мониторы – энергетически эффективные мониторы компьютера.

LCD монитор – это компактный, тонкий и маловесящий дисплей. Способ работы LCD монитора такой же, как у цифровых часов. В нем есть транзисторы, которые заряжают жидкие кристаллы, расположенные между двумя пластинами стекла. Флуоресцентная трубка предоставляет источник света, который проходит через один слой жидких кристаллов. Обычно LCD монитор дольше работает, меньше подвержен перегреву, является компактным и энергетически эффективным.

Основной характеристикой монитора является его разрешающая способность. Разрешающая способность (или разрешение) монитора связана с размером отображаемого изображения и выражается в количестве точек по ширине (по горизонтали) и высоте (по вертикали) отображаемого изображения. Например, если говорят, что монитор имеет разрешение 640×480 , это означает, что изображение состоит из $640 \times 480 = 307200$ точек в прямоугольнике, чьи стороны соответствуют 640 точкам по ширине и 480 точкам по высоте. Поэтому более высокое разрешение соответствует отображению более высокого качества изображения на экране.

Клавиатура

Клавиатура – устройство, предназначенное для ввода в компьютер информации от пользователя (рис. 2.9). Современная клавиатура содержит от 101 до 105 укрепленных в едином корпусе клавиш.

Расположение клавиш на АТ-клавиатуре подчиняется единой общепринятой схеме, спроектированной в расчете на английский алфавит.



Рисунок 2.9 – Клавиатура

По своему назначению клавиши на клавиатуре условно делятся на шесть групп:

- функциональные;
- алфавитно-цифровые;
- управления курсором;
- цифровая панель;
- специализированные;
- модификаторы.

Мышь

Мышь – механический манипулятор, преобразующий движение в управляющий сигнал (рис. 2.10).



Рисунок 2.10 – Компьютерная мышь

В частности, сигнал может быть использован для позиционирования курсора или прокрутки страниц. Он необходим для работы с графическими пакетами, чертежами, при разработке схем и при работе в новых операционных системах. Основной характеристикой мыши является разрешающая способность, измеряемая в точках на дюйм (dpi).

2.3 Периферийные устройства персонального компьютера

Периферийные устройства – это дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных.

Периферийные устройства являются опциональными и технически могут быть отключены от компьютера без потери его работоспособности. Однако абсолютное большинство компьютеров используется вместе с теми или иными периферийными устройствами.

Выделяют три основных типа периферийных устройств:

устройства ввода, использующиеся для ввода информации в компьютер: мышь, сенсорный экран, сканер, веб-камера и видеозахват;

устройства вывода, например принтеры;

устройства хранения, служащие для накопления информации, обрабатываемой компьютером: дисковые устройства, накопители «флеш», твердотельные накопители.

Принтер – печатающее устройство, предназначенное для перевода текста или графики на физический носитель из электронного вида малыми тиражами (от единиц до сотен) без создания печатной формы. Этим принтеры отличаются от полиграфического оборудования, которое за счет печатной формы быстрее и дешевле на крупных тиражах (сотни и более экземпляров).

Существует несколько видов принтеров.

Матричный принтер – компьютерный принтер, создающий изображение на бумаге из отдельных маленьких точек ударным способом (рис. 2.11).



Рисунок 2.11 – Матричный принтер

Струйный принтер – принцип действия струйных принтеров похож на матричный принтер тем, что изображение на носителе формируется из точек. Но вместо головок с иглами в струйных принтерах используется матрица, печатающая жидкими красителями (рис. 2.12).



Рисунок 2.12 – Струйный принтер

Лазерный принтер – один из видов принтеров, позволяющий быстро изготавливать высококачественные отпечатки текста и графики на обычной (не специальной) бумаге (рис. 2.13). Подобно фотокопировальным аппаратам лазерные принтеры используют в работе процесс ксерографической печати, однако отличие состоит в том, что формирование изображения происходит путем непосредственной

экспозиции (освещения) лазерным лучом фоточувствительных элементов принтера.



Рисунок 2.13 – Лазерный принтер

Плоттер – устройство, позволяющее выводить графическую информацию на бумагу или другие носители (рис. 2.14).



Рисунок 2.14 – Плоттер

Сканер – это устройство, которое, анализируя какой-либо объект (обычно изображение, текст), создает цифровую копию изображения объекта. Процесс получения этой копии называется сканированием.

Виды сканеров

Ручной сканер – устройства, сканирование которыми производится путем проведения по обрабатываемому тексту или изображению (рис. 2.15).



Рисунок 2.15 – Ручной сканер

Планшетный сканер – незаменим при работе с компьютером, если есть потребность делать вставки графических изображений или текстов с бумажных носителей в документы, создаваемые при помощи компьютера. Современные настольные сканеры достаточно просты в использовании, имеют интуитивно-понятный интерфейс (рис. 2.16).



Рисунок 2.16 – Планшетный сканер

Настольный сканер – с удобными в использовании функциями и мощным программным обеспечением, позволяющими ускорить рабочий процесс (рис. 2.17).



Рисунок 2.17 – Настольный сканер

Напольный сканер – сканер большого размера (рис. 2.18).



Рисунок 2.18 – Напольный сканер

Модем (модулятор-демодулятор) – устройство, позволяющее компьютеру выходить на связь с другим компьютером посредством телефонных линий.

Виды модемов

Внутренний модем представляет собой плату, вставляемую в PCI-слот в материнской плате (рис. 2.19).

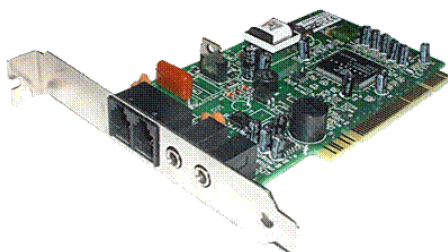


Рисунок 2.19 – Внутренний модем

Внешний модем – это отдельное устройство, которое подключается к компьютеру через один из последовательных портов (COM) или по порту USB (рис. 2.20).



Рисунок 2.20 – Внешние модемы

Колонки – акустическая система, состоящая из большого количества одинаковых громкоговорителей, расположенных вертикально (рис. 2.21).



Рисунок 2.21 – Колонки

Наушники – устройство для персонального прослушивания звуковой информации (рис. 2.22). В комплекте с микрофоном могут служить головной гарнитурой – средством для ведения переговоров по телефону или иному средству голосовой связи.



Рисунок 2.22 – Наушники

ТВ-тюнер – род телевизионного приемника (тюнера), предназначенный для приема телевизионного сигнала в различных форматах вещания, с показом на мониторе компьютера (рис. 2.23).



Рисунок 2.23 – ТВ-тюнер

Вопросы для самоконтроля

1. Для чего предназначен винчестер?
2. От чего зависит производительность работы компьютера (быстрота выполнения операций)?
3. Что является характеристикой монитора?
4. Что обеспечивают шины персонального компьютера?
5. В чем измеряется тактовая частота процессора?
6. В каком виде обрабатывает информацию процессор?
7. Что размещается на материнской плате?

8. Какова максимальная информационная емкость стандартных CD-ROM дисков ?
9. Что такое персональный компьютер?
10. Что такое дисковод?
11. Где записана программа тестирования персонального компьютера в момент его включения?
12. Что включает минимальная комплектация персонального компьютера?
13. Для чего поверхность магнитного диска разбита на секторы?
14. К какому виду памяти относится постоянно запоминающее устройство (ПЗУ)?
15. Какое устройство ПК производит обработку информации?
16. Кем сформулированы общие принципы функционирования вычислительных машин в 40-х гг. XX столетия?
17. Какое из устройств предназначено для ввода информации?
18. Что такое манипулятор «мышь»?
19. Какое устройство ПК может оказывать вредное воздействие на здоровье человека?

3 АРИФМЕТИЧЕСКИЕ ОСНОВЫ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

3.1 Системы счисления

Система счисления – это способ представления чисел и соответствующие ему правила действия над числами. Разнообразные системы счисления можно разделить на *непозиционные* и *позиционные*.

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 римской системе в качестве цифр используются латинские буквы:

<i>I</i>	<i>V</i>	<i>X</i>	<i>L</i>	<i>CD</i>	<i>M</i>	
1	5	10	50	100	500	1000

Пример 1. Число *CCXXXII* складывается из двух сотен, трех десятков и двух единиц и равно двумстам тридцати двум.

В римских числах цифры записываются слева направо в порядке убывания. В таком случае их значения складываются. Если же слева записана меньшая цифра, а справа – большая, то их значения вычитаются.

Пример 2. $VI = 5 + 1 = 6$, а $IV = 5 - 1 = 4$.

Пример 3. $MCMXCVIII = 1000 + (-100 + 1000) + (-10 + 100) + 5 + 1 + 1 + 1 = 1998$.

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

Система счисления, применяемая в современной математике, является *позиционной десятичной системой*. Ее основание равно десяти, так как запись любых чисел производится с помощью десяти цифр:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Для записи чисел в позиционной системе с основанием n нужно иметь *алфавит* из n цифр. Обычно для этого при $n < 10$ используют n первых арабских цифр, при $n > 10$ к десяти арабским цифрам добавляют буквы. В таблице 3.1 приведены примеры алфавитов нескольких систем.

Таблица 3.1 – Примеры алфавитов

Основание	Система счисления	Алфавит
$n = 2$	Двоичная	0 1
$n = 8$	Восьмеричная	0 1 2 3 4 5 6 7
$n = 10$	Десятичная	0123456789
$n = 16$	Шестнадцатеричная	0123456789ABCDEF

Запись чисел в каждой из систем счисления с основанием q означает сокращенную запись выражения:

$$a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m},$$

где a_i – цифры системы счисления; n и m – число целых и дробных разрядов соответственно.

Например:

Разряды 3 2 1 0 -1

Число 1 0 1 1, $1_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1}$;

Разряды 2 1 0 -1 -2

Число 2 7 6, $5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} + 2 \cdot 8^{-2}$.

Пример 4. Получить развернутую форму десятичных чисел 32478; 26,387.

$$32478_{10} = 3 \cdot 10000 + 2 \cdot 1000 + 4 \cdot 100 + 7 \cdot 10 + 8 =$$

$$= 3 \cdot 10^4 + 2 \cdot 10^3 + 4 \cdot 10^2 + 7 \cdot 10^1 + 8 \cdot 10^0.$$

$$26,387_{10} = 2 \cdot 10^1 + 6 \cdot 10^0 + 3 \cdot 10^{-1} + 8 \cdot 10^{-2} + 7 \cdot 10^{-3}.$$

Пример 5. Получить развернутую форму чисел 112_3 , 101101_2 , $15FC_{16}$, $101,11_2$.

$$112_3 = 1 \cdot 10^2 + 1 \cdot 10^1 + 2 \cdot 10^0.$$

$$101101_2 = 1 \cdot 10^{101} + 0 \cdot 10^{100} + 1 \cdot 10^{11} + 1 \cdot 10^{10} + 0 \cdot 10^1 + 1 \cdot 10^0.$$

$$15FC_{16} = 1 \cdot 10^3 + 5 \cdot 10^2 + F \cdot 10^1 + C.$$

$$101,11_2 = 1 \cdot 10^{10} + 0 \cdot 10^1 + 1 \cdot 10^0 + 1 \cdot 10^{-1} + 1 \cdot 10^{-10}.$$

Компьютеры используют двоичную систему счисления, так как она имеет ряд преимуществ перед другими системами:

– для ее реализации нужны технические устройства с двумя устойчивыми состояниями (есть ток – нет тока, намагничен – не намагничен и т.п.);

– представление информации посредством только двух состояний надежно и помехоустойчиво;

– возможно применение аппарата булевой алгебры для выполнения логических преобразований информации; двоичная арифметика намного проще десятичной.

Недостаток двоичной системы – быстрый рост числа разрядов, необходимых для записи чисел.

Двоичная система, удобная для компьютеров, для человека не удобна из-за ее громоздкости и непривычной записи. Перевод чисел из десятичной системы счисления в двоичную, и наоборот выполняет компьютер. Однако, чтобы профессионально использовать его, следует научиться понимать слово машины. Для этого и разработаны восьмеричная и шестнадцатеричная системы. Числа в этих системах читаются почти так же легко, как десятичные, требуют соответственно в три (восьмеричная) и четыре (шестнадцатеричная) раза меньше разрядов, чем в двоичной системе (ведь числа 8 и 16 соответственно третья и четвертая степени числа 2).

$$\begin{array}{cccccccccccc}
 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & , & 0 & 0 & 1_2 & = & 5 & 3 & 7, & 1_8 \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow & & & & \downarrow & & & & & & & \\
 & & 5 & & 3 & & 7 & & & & 1 & & & & & & &
 \end{array}$$

$$\begin{array}{cccccccccccc}
 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & , & 1 & 1 & 1 & 1_2 & = & 1 & A & 3, & F_{16} \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow & & & & \downarrow & & & & & & & \\
 & & 1 & & A & & 3 & & & & F & & & & & & &
 \end{array}$$

$$\begin{array}{cccccccc}
 2 & 5 & 1, & 5 & 6_8 & = & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1, & 1 & 0 & 1 & 1 & 1 & 0 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 010 & 101 & 001, & 101 & 110 & & 2 & & 5 & & 1 & & 5 & & 6 & & & & & & &
 \end{array}$$

$$\begin{array}{cccccccc}
 A & 9, & B & 8_{16} & = & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1, & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0_2 \\
 \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 1010 & 1001 & 1011 & 1000 & & A & & 9 & & B & & 8 & & & & & & & & & & &
 \end{array}$$

3.2 Перевод чисел из одной системы счисления в другую

Перевод целых чисел из одной системы счисления в другую осуществляется по следующей схеме:

- основание новой системы счисления выразить в десятичной системе счисления и все последующие действия производить в ней;
- последовательно выполнять деление данного числа и получаемых неполных частных на основание новой системы счисления до тех пор, пока не получим неполное частное, меньшее делителя;
- полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления;
- составить число в новой системе счисления, записывая его, начиная с последнего частного.

Пример 1. Перевести число 37_{10} в двоичную систему.

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

$$\begin{array}{r|l}
 \begin{array}{r}
 _37 \\
 \underline{36} \\
 a_0=1
 \end{array} & \begin{array}{l}
 2 \\
 _18 \\
 \underline{18} \\
 a_1=0
 \end{array} & 2 \\
 & \begin{array}{l}
 _9 \\
 \underline{8} \\
 a_2=1
 \end{array} & 2 \\
 & \begin{array}{l}
 _4 \\
 \underline{4} \\
 a_3=0
 \end{array} & 2 \\
 & \begin{array}{l}
 _2 \\
 \underline{2} \\
 a_4=0
 \end{array} & 2 \\
 & \begin{array}{l}
 _1 \\
 \underline{1} \\
 1=a_5
 \end{array} &
 \end{array}$$

Отсюда $37_{10} = 100101_2$.

Пример 2. Перевести десятичное число 315 в восьмеричную и в шестнадцатеричную системы.

Решение:

$$\begin{array}{r|l}
 \begin{array}{r}
 _315 \\
 \underline{24} \\
 _75 \\
 \underline{72} \\
 3
 \end{array} & \begin{array}{l}
 8 \\
 _39 \\
 \underline{32} \\
 7
 \end{array} & 8 \\
 & 4 &
 \end{array}
 \qquad
 \begin{array}{r|l}
 \begin{array}{r}
 _315 \\
 \underline{16} \\
 _155 \\
 \underline{144} \\
 11
 \end{array} & \begin{array}{l}
 16 \\
 _19 \\
 \underline{16} \\
 3
 \end{array} & 16 \\
 & 1 &
 \end{array}$$

Отсюда следует: $315_{10} = 473_8 = 13B_{16}$.

Напомним, что $11_{10} = B_{16}$.

Для перевода дробных чисел из одной системы счисления в другую необходимо выполнить следующие действия:

– основание новой системы счисления выразить в десятичной системе и все последующие действия производить в десятичной системе счисления;

– последовательно умножать данное число и получаемые дробные части произведений на основание новой системы до тех пор, пока дробная часть произведения не станет равной нулю или не будет достигнута требуемая точность представления числа в новой системе счисления;

– полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления;

– составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

Пример 3. Перевести десятичную дробь 0,1875 в двоичную, восьмеричную и шестнадцатеричную системы.

Решение. Вертикальная черта отделяет целые части чисел от дробных частей.

0	1875
	·2
0	3750
	·2
0	7500
	·2
1	5000
	·2
1	10000

0	1875
	·8
1	5000
	·8
4	0000

0	1875
	·16
1	1250
	·16
1	875
	·16
3	0000

Отсюда: $0,1875_{10} = 0,0011_2 = 0,14_8 = 0,3_{16}$.

Перевод смешанных чисел, содержащих целую и дробную части, осуществляется в два этапа. Целая и дробная части исходного числа переводятся отдельно по соответствующим алгоритмам. В итоговой записи числа в новой системе счисления целая часть отделяется от дробной запятой.

Пример 4. Перевести десятичное число 315,1875 в восьмеричную и в шестнадцатеричную системы счисления.

Решение. Из рассмотренных выше примеров следует:

$$315,1875_{10} = 473,14_8 = 13B,3_{16}.$$

3.3 Выполнение арифметических операций в позиционных системах счисления

Рассмотрим основные арифметические операции: сложение, вычитание, умножение и деление. Правила выполнения этих операций в десятичной системе хорошо известны – это сложение, вычитание, умножение столбиком и деление углом. Эти правила применимы и ко всем другим позиционным системам счисления. Только надо пользоваться особыми таблицами сложения и умножения для каждой системы счисления.

Сложение. Сложение двоичных чисел выполняется в соответствии со следующими правилами, действительными для каждого разряда двоичного числа:

$$\begin{array}{ll} 0+0=0 & 0+1=1 \\ 1+1=0+\text{перенос} & 1+1+1=1+\text{перенос} \end{array}$$

Под переносом понимается добавление единицы к старшему разряду, стоящему слева от суммируемого.

Пример 1. Вычислить сумму трех двоичных чисел.

Решение. Представлено на рисунке 3.1.

$$\begin{array}{r} \begin{array}{r} \overset{14}{+} \\ \overset{12}{12} \\ \hline 26 \end{array} \quad \begin{array}{r} 11 \\ 01110 \\ +01100 \\ \hline 11010 \end{array} \begin{array}{l} \leftarrow \text{разряды переноса} \\ \leftarrow \text{первое слагаемое} \\ \leftarrow \text{второе слагаемое} \\ \leftarrow \text{направление суммирования} \end{array} \\ \text{сумма} \rightarrow \end{array}$$

Рисунок 3.1 – Решение примера 1

Легко видеть, что сложение в двоичной системе счисления аналогично сложению в десятичной.

Двоичное вычитание также аналогично десятичному и выполняется в соответствии со следующими правилами:

$$\begin{array}{ll} 0-0=0 & 1-0=1 \\ 1-1=0 & 0-1=1 \text{ и заем единицы старшего разряда} \end{array}$$

Пример 2. Вычислить разность двоичных чисел.

Решение. Представлено на рисунке 3.2.

$$\begin{array}{r} 22 \\ -12 \\ \hline 10 \end{array} \quad \begin{array}{r} {}^{(0)} \\ 10110 \\ -01100 \\ \hline 01010 \end{array}$$

Рисунок 3.2 – Решение примера 2

Операция двоичное вычитание в чистом виде в компьютерах почти не используется. Вычитание обычно выполняется косвенно в виде сложения чисел с учетом их знака.

Таблицы сложения для восьмеричной и шестнадцатеричной систем счисления представлены ниже (табл. 3.2; 3.3).

Таблица 3.2 – Сложение в восьмеричной системе

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Таблица 3.3 – Сложение в шестнадцатеричной системе

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

При сложении цифры суммируются по разрядам и если при этом возникает избыток, то он переносится влево.

Пример 3. Вычислить сумму чисел 15 и 6 в различных системах счисления.

Решение:

Десятичная $15_{10} + 6_{10} = 21_{10}$

$$\begin{array}{r}
 + \quad \begin{array}{r} 1 \ 5 \\ \hline 2 \ 1 \end{array} \\
 \downarrow \quad \downarrow \\
 \downarrow \quad 5+6=11=10+1 \\
 \downarrow \quad 1+1=2
 \end{array}$$

Двоичная $1111_2 + 110_2 = 10101_2$

$$\begin{array}{r}
 + \quad \begin{array}{r} 1 \ 1 \ 1 \ 1 \\ \hline 1 \ 1 \ 0 \end{array} \\
 \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
 \downarrow \quad \downarrow \quad \downarrow \quad 1+0=1 \\
 \downarrow \quad \downarrow \quad 1+1=2=2+0 \\
 \downarrow \quad 1+1+1=3=2+1 \\
 \downarrow \quad 1+1=2=2+0
 \end{array}$$

Восьмеричная:

$$17_8 + 6_8 = 25_8$$

$$\begin{array}{r}
 + \quad \begin{array}{r} 1 \ 7 \\ \hline 2 \ 5 \end{array} \\
 \downarrow \quad \downarrow \\
 \downarrow \quad 7+6=13=8+5 \\
 \downarrow \quad 1+1=2
 \end{array}$$

Шестнадцатеричная:

$$F_{16} + 6_{16} = 15_{16}$$

$$\begin{array}{r}
 + \quad \begin{array}{r} F \\ \hline 6 \end{array} \\
 \downarrow \\
 \downarrow \quad 15+6=21=16+5
 \end{array}$$

Ответ: $15 + 6 = 21_{10} = 10101_2 = 25_8 = 15_{16}$.

Проверка: преобразуем полученные суммы к десятичному виду:

$$0101_2 = 2^4 + 2^2 + 2^0 = 16 + 4 + 1 = 21;$$

$$25_8 = 2 \cdot 8^1 + 5 \cdot 8^0 = 16 + 5 = 21;$$

$$15_{16} = 1 \cdot 16^1 + 5 \cdot 16^0 = 16 + 5 = 21.$$

Пример 4. Вычислить сумму чисел 15, 7 и 3.

Решение:

Десятичная: $15_{10} + 7_{10} + 3_{10}$

Двоичная: $1111_2 + 111_2 + 11_2$

$$\begin{array}{r}
 + \quad 15 \\
 \quad 7 \\
 \quad 3 \\
 \hline
 25 \\
 \left\{ \begin{array}{l} 15+7+3 = 25 = 20+5 \\ 1+1 = 2 \end{array} \right.
 \end{array}$$

$$\begin{array}{r}
 + \quad 1111 \\
 \quad 111 \\
 \quad 11 \\
 \hline
 1\ 1001 \\
 \left\{ \begin{array}{l} 1+1+1 = 3 = 2+1 \\ 1+1+1+1 = 4 = 2+2+0 \\ 1+1 = 2 = 2+0 \\ 1+1+1 = 3 = 2+1 \end{array} \right.
 \end{array}$$

Восьмеричная: $17_8 + 7_8 + 3_8$. Шестнадцатеричная: $F_{16} + 7_{16} + 3_{16}$.

$$\begin{array}{r}
 + \quad 15 \\
 \quad 7 \\
 \quad 3 \\
 \hline
 31 \\
 \left\{ \begin{array}{l} 7+7+3 = 17 = 2 \cdot 8 + 1 \\ 2+1 = 3 \end{array} \right.
 \end{array}$$

$$\begin{array}{r}
 + \quad 15 \\
 \quad 7 \\
 \quad 3 \\
 \hline
 19 \\
 \left\{ \begin{array}{l} 15+7+3 = 25 = 16+9 \end{array} \right.
 \end{array}$$

Ответ: $15 + 7 + 3 = 25_{10} = 11001_2 = 31_8 = 19_{16}$.

Проверка:

$$11001_2 = 2^4 + 2^3 + 2^0 = 16 + 8 + 1 = 25;$$

$$31_8 = 3 \cdot 8^1 + 1 \cdot 8^0 = 24 + 1 = 25;$$

$$19_{16} = 1 \cdot 16^1 + 9 \cdot 16^0 = 16 + 9 = 25.$$

Пример 5. Вычислить сумму чисел 141,5 и 59,75.

Решение:

Десятичная:

$$141,5_{10} + 59,75_{10}$$

Двоичная:

$$10001101,1_2 + 111011,11_2$$

$$\begin{array}{r}
 \quad 111 \\
 141,50 \\
 + \quad 59,75 \\
 \hline
 201,25 \\
 \left\{ \begin{array}{l} 0+5=5 \\ 5+7=12=10+2 \\ 1+9+1=11=10+1 \\ 4+5+1=10=10+0 \\ 1+1=2 \end{array} \right.
 \end{array}$$

$$\begin{array}{r}
 \quad 1111111 \\
 10001101,10 \\
 + \quad 111011,11 \\
 \hline
 11001001,01 \\
 \left\{ \begin{array}{l} 0+1=1 \\ 1+1=2=2+0 \\ 1+1+1=3=2+1 \\ 0+1+1=2=2+0 \\ 1+0+1=2=2+0 \\ 1+1+1=3=2+1 \\ 0+1+1=2=2+0 \\ 0+1+1=2=2+0 \\ 0+1=1 \end{array} \right.
 \end{array}$$

Восьмеричная:
 $215,4_8 + 73,6_8$

$$\begin{array}{r}
 111 \\
 215,4 \\
 + 73,6 \\
 \hline
 311,2 \\
 \begin{array}{l}
 \left| \begin{array}{l} \left| \begin{array}{l} 4+6=10=8+2 \\ 5+3+1=9=8+1 \\ 1+7+1=9=8+1 \end{array} \right. \\ 2+1=3 \end{array} \right.
 \end{array}
 \end{array}$$

Шестнадцатеричная:
 $8D,8_{16} + 3B,C_{16}$

$$\begin{array}{r}
 11 \\
 8D,8 \\
 + 3B,C \\
 \hline
 C9,4 \\
 \begin{array}{l}
 \left| \begin{array}{l} \left| \begin{array}{l} 8+12=20=16+4 \\ 13+11+1=25=16+9 \\ 8+3+1=12=C_{16} \end{array} \right. \end{array} \right.
 \end{array}
 \end{array}$$

Ответ: $141,5 + 59,75 = 201,25_{10} = 11001001,01_2 = 311,2_8 = C9,4_{16}$

Проверка. Преобразуем полученные суммы к десятичному виду:

$$\begin{aligned}
 11001001,01_2 &= 2^7 + 2^6 + 2^3 + 2^0 + 2^{-2} = 201,25; \\
 311,2_8 &= 3 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 + 2 \cdot 8^{-1} = 201,25; \\
 C9,4_{16} &= 12 \cdot 16^1 + 9 \cdot 16^0 + 4 \cdot 16^{-1} = 201,25.
 \end{aligned}$$

Пример 6. Вычитание чисел в различных системах счисления.

Решение:

Двоичная:
 $10_2 - 1_2$

$$\begin{array}{r}
 1 \\
 10 \\
 - 1 \\
 \hline
 1 \\
 \left| 2-1=1 \right.
 \end{array}$$

Восьмеричная:
 $10_8 - 1_8$

$$\begin{array}{r}
 1 \\
 10 \\
 - 1 \\
 \hline
 7 \\
 \left| 8-1=7 \right.
 \end{array}$$

Шестнадцатеричная:
 $10_{16} - 1_{16}$

$$\begin{array}{r}
 1 \quad \text{заемы} \\
 10 \\
 - 1 \\
 \hline
 F \\
 \left| 16-1=15=F_{16} \right.
 \end{array}$$

Умножение. Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо заимствовать из соответствующих, рассматриваемой системе, таблиц умножения (табл. 3.3; 3.4) и сложения (табл. 3.1; 3.2).

Таблица 3.4 – Умножение в двоичной системе

·	0	1
0	0	0
1	0	1

Таблица 3.5 – Умножение в восьмеричной системе

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Ввиду чрезвычайной простоты таблицы умножения в двоичной системе умножение сводится лишь к сдвигам множимого и сложениям.

Пример 7. Перемножить числа 5 и 6 в различных системах счисления.

Решение:

Десятичная:

$$\begin{array}{r} 5_{10} \cdot 6_{10} \\ 5 \\ \times \quad 6 \\ \hline 30 \end{array}$$

Двоичная:

$$\begin{array}{r} 101_2 \cdot 110_2 \\ 101 \\ \times \quad 110 \\ \hline 000 \\ + \quad 101 \\ \quad 101 \\ \hline 11110 \end{array}$$

Восьмеричная:

$$\begin{array}{r} 5_8 \cdot 6_8 \\ 5 \\ \times \quad 6 \\ \hline 36 \end{array}$$

Ответ: $5 \cdot 6 = 30_{10} = 11110_2 = 36_8$.

Проверка. Преобразуем полученные произведения к десятичному виду:

$$11110_2 = 2^4 + 2^3 + 2^2 + 2^1 = 30;$$

$$36_8 = 3 \cdot 8^1 + 6 \cdot 8^0 = 30.$$

Пример 8. Перемножить числа 115 и 51 в различных системах счисления.

Решение:

Десятичная: $1155_{10} \cdot 51_{10}$	Двоичная: $1110011_2 \cdot 110011_2$	Восьмеричная: $163_8 \cdot 63_8$
$\begin{array}{r} 115 \\ \times 51 \\ \hline 115 \\ + 575 \\ \hline 5865 \end{array}$	$\begin{array}{r} 1110011 \\ \times 110011 \\ \hline 1110011 \\ + 1110011 \\ 1110011 \\ \hline 1011011101001 \end{array}$	$\begin{array}{r} 163 \\ \times 63 \\ \hline 531 \\ + 1262 \\ \hline 13351 \end{array}$

Ответ: $115 \cdot 51 = 5865_{10} = 1011011101001_2 = 13351_8$.

Проверка. Преобразуем полученные произведения к десятичному виду:

$$1011011101001_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^6 + 2^5 + 2^3 + 2^0 = 586;$$

$$13351_8 = 1 \cdot 8^4 + 3 \cdot 8^3 + 3 \cdot 8^2 + 5 \cdot 8^1 + 1 \cdot 8^0 = 586.$$

Деление. Деление в любой позиционной системе счисления производится по тем же правилам, как и деление, углом в десятичной системе. В двоичной системе деление выполняется особенно просто, ведь очередная цифра частного может быть только нулем или единицей.

Пример 9. Разделить число 30 на число 6.

Решение:

Десятичная: $30_{10} : 6_{10}$	Двоичная: $11110_2 : 110_2$	Восьмеричная: $36_8 : 6_8$
$\begin{array}{r} 30 \overline{) 6} \\ - 30 \overline{) 5} \\ \hline 0 \end{array}$	$\begin{array}{r} 11110 \overline{) 110} \\ - 110 \overline{) 101} \\ \hline 110 \\ 110 \\ \hline 0 \end{array}$	$\begin{array}{r} 36 \overline{) 6} \\ - 36 \overline{) 5} \\ \hline 0 \end{array}$

Ответ: $30 : 6 = 5_{10} = 101_2 = 5_8$.

Вопросы и задания для самоконтроля

1. Что такое система счисления?
2. Какие виды систем счисления вы знаете?
3. Что называют основанием системы счисления?
4. Что представляет собой развернутая форма числа?
5. Чему будет равняться десятичное число $16,0110$ в шестнадцатеричной системе счисления?
6. Чему будет равняться десятичное число $88,8810$ в восьмеричной системе счисления?
7. Укажите, какое число имеет недопустимую запись в восьмеричной системе счисления, выберите один ответ:
а) 1111 ; б) 776 ; в) 207 ; г) 678 .
8. Перевести из десятичной системы счисления в двоичную:
а) 123 ; б) 4510 ; в) $139,4510$; г) $137,3510$; д) $111,2510$.
9. Перевести из двоичной системы счисления в десятичную:
а) $111001,1011_2$; б) $110101,1101_2$; в) $110011,111_2$; г) $110111,1001_2$; д) $111011,101_2$.
10. Перевести из двоичной системы счисления в шестнадцатеричную и восьмеричную 111110011_2 .
11. Перевести из шестнадцатеричной и восьмеричной систем счисления в двоичную $13E7_{16}, 6423_8$.
12. Выполнить арифметические действия в двоичной системе счисления и проверить их правильность в десятичной:
а) $111001,01+101011,101$; ж) $10001000-1101000$;
б) $1101,1 \cdot 101$; з) $11110111:1101$;
в) $10111011-1000111$; и) $101111,001+101111,11$;
г) $1111001:1011$; к) $1001,1 \cdot 111$;
д) $101011,001+100111,10$; л) $1111001-1100010$;
е) $1001,1 \cdot 1011$; м) $100100000:110$.
13. В каких системах счисления справедливы следующие равенства:
а) $21+24=100$; б) $22+44=110$; в) $20+25=100$?
14. Десятичное число 59 эквивалентно числу 214 в некоторой другой системе счисления. Найти основание этой системы.
15. Найти такое число XY в некоторой системе счисления с основанием Z не более чем 16 , что выполнилось бы тождество $XY_Z = X^Y_{10}$.

4 ЛОГИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРА

4.1 Логика высказываний

Утверждения, не требующие доказательства, называются *аксиомами*. *Основными* называются элементарные понятия, которые нельзя определить через другие.

Высказыванием (суждением) называется повествовательное предложение, относительно которого можно однозначно сказать, *истинно* оно либо *ложно*. Говорят, что истинное высказывание имеет логическое значение «истина», а ложное – «ложь». В математике логические значения «истина» и «ложь» обозначают обычно буквами «И» и «Л» соответственно.

Цепи умозаключений складываются в рассуждения и доказательства, в которых заключение предшествующего умозаключения становится посылкой следующего. Условием правильности доказательства является не только истинность исходных суждений, но и истинность каждого входящего в его состав умозаключения. Доказательства должны быть построены по *законам логики*.

Пусть A и B обозначают некоторые произвольные *высказывания*. Высказывание называется *простым*, если его нельзя разбить на части, которые сами являются высказываниями.

Сложное высказывание состоит из нескольких высказываний, которые соединены между собой по некоторым правилам (законам). Обычно в логике рассматривают следующие логические операции над этими высказываниями (табл. 4.1).

Таблица 4.1 – Логические операции

Операция	Название операции	Краткое прочтение полученного высказывания	Полное прочтение полученного высказывания
$\neg A$ (или \bar{A})	<i>Отрицание</i>	Не A	Неверно, что A
$A \wedge B$ (или $A \& B$)	<i>Конъюнкция</i>	A и B	Верно, что A , и верно, что B
$A \vee B$	<i>Дизъюнкция</i>	A или B	Верно, что A , или верно, что B , или верны оба высказывания
$A \oplus B$	<i>Исключающая дизъюнкция</i>	или A , или B	Верно что-то одно: или A , или B

Операция	Название операции	Краткое прочтение полученного высказывания	Полное прочтение полученного высказывания
$A \leftrightarrow B$ (или $A \equiv B$)	Эквивалентность	A эквивалентно B	Верно, что A , тогда и только тогда, когда верно, что B
$A \rightarrow B$	Импликация	Если A , то B . A влечет B	Если верно, что A , то верно, что B

Можно попытаться прочесть полученные высказывания, когда A – это высказывание «Шумел камыш», а B – высказывание «Деревьягнулись».

Пример 1. Записать сложное высказывание «Если пойдет дождь, то я либо возьму зонт, либо надену пальто» в формализованном виде, обозначая его латинскими буквами.

Решение. Обозначим за A высказывание «пойдет дождь», за B – «возьму зонт», за C – «надену пальто». Тогда наше высказывание запишется следующим образом: $A \rightarrow (B \vee C)$.

Следующая таблица определяет, какие значения принимают высказывания, полученные с помощью этих операций, если исходные высказывания A и B принимают значения $И$ или $Л$. Такая таблица называется *таблицей истинности*. (табл. 4.2)

Таблица 4.2 – Таблица истинности

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
И	И	Л	И	И	И	И
И	Л	Л	Л	И	Л	Л
Л	И	И	Л	И	И	Л
Л	Л	И	Л	Л	И	И

Сложные формулы состоят из нескольких логических операций.

Все операции располагаются в порядке «старшинства»: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.

Скобки также регулируют порядок операций, как в арифметике. Иногда опускают не все скобки, которые можно не ставить, чтобы формула легче воспринималась.

Для работы с высказываниями используют следующие законы.

1. Закон двойного отрицания:

$$\neg(\neg A) = A.$$

2. Переместительный (коммутативный) закон:

– для логического сложения: $A + B = B + A$;

– для логического умножения: $A \cdot B = B \cdot A$.

В обычной алгебре $a + b = b + a$, $ab = ba$.

3. Сочетательный (ассоциативный) закон:

– для логического сложения: $(A + B) + C = A + (B + C)$;

– для логического умножения: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$.

В обычной алгебре $(a + b) + c = a + (b + c) = a + b + c$,

$$(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c.$$

4. Распределительный (дистрибутивный) закон:

– для логического сложения:

$$(A + B) \cdot C = (A \cdot C) + (B \cdot C);$$

– для логического умножения:

$$(A \cdot B) + C = (A + C) \cdot (B + C).$$

В обычной алгебре $(a + b) \cdot c = a \cdot c + b \cdot c$.

5. Закон общей инверсии (законы де Моргана):

– для логического сложения: $\neg(A + B) = \neg A \cdot \neg B$;

– для логического умножения: $\neg(A \cdot B) = \neg A + \neg B$.

6. Закон равносильности (идемпотентности – от лат. слов *idem* – тот же самый и *potens* – сильный):

– для логического сложения: $A + A = A$;

– для логического умножения: $A \cdot A = A$.

7. Законы исключения констант:

– для логического сложения: $A + 1 = 1$, $A + 0 = A$;

– для логического умножения: $A \cdot 1 = A$, $A \cdot 0 = 0$.

8. Закон противоречия: $A \cdot \neg A = 0$.

Невозможно, чтобы противоречивые высказывания были истинны одновременно.

9. Закон исключенного третьего:

$A + \neg A = 1$. Из двух противоречивых высказываний одно истинно.

10. Закон поглощения:

– для логического сложения: $A + (A \cdot B) = A$;

– для логического умножения: $A \cdot (A + B) = A$.

11. Закон **исключения** (склеивания):

- для логического сложения: $(A \cdot B) + (A \cdot (\neg B)) = A$;
- для логического умножения: $(A + B) \cdot (A + (\neg B)) = A$.

12. Закон **контрапозиции** (правило перевертывания):

$$(A \rightarrow B) = (\neg B \rightarrow \neg A).$$

Построение логической функции по ее таблице истинности. Известному немецкому математику и логика Эрнесту Шредеру принадлежит идея обозначения в качестве знака для ложного суждения – 0, а для истины – 1. Тогда таблица истинности приобретает вполне арифметический вид (табл. 4.3).

Таблица 4.3 – Таблица истинности в двоичной системе

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

Переменные принимают различные значения *И* и *Л*, формула соответственно проводимым операциям принимает итоговое значение.

Пример 2. Составить таблицу истинности для сложного высказывания $(\neg X \wedge \neg Y) \vee \neg X$.

Решение. Решение примера 2 приведено в таблице 4.4.

Таблица 4.4 – Таблица истинности для высказывания $(\neg X \wedge \neg Y) \vee \neg X$

X	Y	$\neg X$	$\neg Y$	$\neg X \wedge \neg Y$	$(\neg X \wedge \neg Y) \vee \neg X$
И	И	Л	Л	Л	Л
И	Л	Л	Л	Л	Л
Л	И	И	Л	Л	И
Л	Л	И	И	И	И

Пример 3. Проверить с помощью таблиц истинности закон «двойного отрицания».

Решение. Закон «двойного отрицания» формально можно записать:

$$\neg(\neg X) \leftrightarrow X.$$

В таблице 4.5 представлена таблица истинности для данной функции.

Таблица 4.5 – Таблица истинности для высказывания $\neg(\neg X) \leftrightarrow X$

X	$\neg X$	$\neg(\neg X)$	$\neg(\neg X) \leftrightarrow X$
И	Л	И	И
Л	И	Л	И

Итак, закон «двойного отрицания» выполняется при всех значениях переменной X .

4.2 Методы решения логических задач

1. Решение задач средствами алгебры логики

Обычно используется следующая схема решения:

- изучается условие задачи;
- вводится система обозначений для логических высказываний;
- конструируется логическая формула, описывающая логические связи между всеми высказываниями условия задачи;
- определяются значения истинности этой логической формулы;
- из полученных значений истинности формулы определяются значения истинности введенных логических высказываний, на основании которых делается заключение о решении.

Пример 4. Трое друзей, болельщиков автогонок «Формула-1», спорили о результатах предстоящего этапа гонок.

– Вот увидишь, Шумахер не придет первым, – сказал Джон. Первым будет Хилл.

– Да нет же, победителем будет, как всегда, Шумахер, – воскликнул Ник. – А об Алезе и говорить нечего, ему не быть первым.

Питер, к которому обратился Ник, возмутился:

– Хиллу не видать первого места, а вот Алезии пилотирует самую мощную машину.

По завершении этапа гонок оказалось, что каждое из двух предположений двоих друзей подтвердилось, а оба предположения третьего из друзей оказались неверны. Кто выиграл этап гонки?

Решение. Введем обозначения для логических высказываний:

Ш – победит Шумахер; X – победит Хилл; А – победит Алезия.

Реплика Ника «Алезия пилотирует самую мощную машину» не содержит никакого утверждения о месте, которое займет этот гонщик, поэтому в дальнейших рассуждениях не учитывается.

Зафиксируем высказывания каждого из друзей:

Джон: $\neg Ш \cdot X$, Ник: $Ш \cdot (\neg А)$, Питер: $\neg X$.

Учитывая то, что предположения двух друзей подтвердились, а предположения третьего неверны, запишем и упростим истинное высказывание:

$$\begin{aligned} & (\neg Ш \cdot X) \cdot (Ш \cdot (\neg А)) \cdot (\neg (\neg X)) \vee (\neg Ш \cdot X) \cdot (\neg (Ш \cdot (\neg А))) \cdot \\ & \quad \cdot (\neg X) \vee (\neg (\neg Ш \cdot X) \cdot (Ш \cdot (\neg А))) \cdot (\neg X) = \\ & = (Ш \cdot (\neg X)) \cdot Ш \cdot (\neg А) \cdot (\neg X) = Ш \cdot (\neg А) \cdot (\neg X). \end{aligned}$$

Высказывание $Ш \cdot (\neg А) \cdot (\neg X)$ истинно только при $Ш=1$, $А=0$, $X=0$.

Ответ. Победителем этапа гонок стал Шумахер.

Пример 5. Некий любитель приключений отправился в кругосветное путешествие на яхте, оснащенной бортовым компьютером. Его предупредили, что чаще всего выходят из строя три узла компьютера – a , b , c , и дали необходимые детали для замены. Выяснить, какой именно узел надо заменить, он может по сигнальным лампочкам на контрольной панели. Лампочек тоже ровно три: x , y и z .

Инструкция по выявлению неисправных узлов такова:

1) если неисправен хотя бы один из узлов компьютера, то горит по крайней мере одна из лампочек x , y , z ;

2) если неисправен узел a , но исправен узел c , то загорается лампочка y ;

3) если неисправен узел c , но исправен узел b , загорается лампочка y , но не загорается лампочка x ;

4) если неисправен узел b , но исправен узел c , то загораются лампочки x и y или не загорается лампочка x ;

5) если горит лампочка x при этом либо неисправен узел a , либо все три узла a, b, c исправны, то горит и лампочка y .

В пути компьютер сломался. На контрольной панели загорелась лампочка x . Тщательно изучив инструкцию, путешественник починил компьютер. Но с этого момента и до конца плавания его не оставляла тревога. Он понял, что инструкция несовершенна, и есть случаи, когда она ему не поможет.

Какие узлы заменил путешественник? Какие изъяны он обнаружил в инструкции?

Решение. Введем обозначения для логических высказываний:

a – неисправен узел a ; x – горит лампочка x ;

b – неисправен узел b ; y – горит лампочка y ;

c – неисправен узел c ; z – горит лампочка z .

Правила 1–5 выражаются следующими формулами:

$$a \vee b \vee c \rightarrow x \vee y \vee z \quad (1)$$

$$a \cdot (\neg c) \rightarrow y \quad (2)$$

$$c \cdot (\neg b) \rightarrow y \cdot (\neg x) \quad (3)$$

$$b \cdot (\neg c) \rightarrow (x \cdot y \vee (\neg x)) \quad (4)$$

$$(a \vee (\neg a) \cdot (\neg b) \cdot (\neg c)) \cdot x \rightarrow y \quad (5)$$

Формулы (1)–(5) истинны по условию, следовательно, их конъюнкция тоже истинна:

$$(a \vee b \vee c \rightarrow x \vee y \vee z) \cdot (a \cdot (\neg c) \rightarrow y) \cdot (c \cdot (\neg b) \rightarrow y \cdot (\neg x)) \cdot$$

$$(b \cdot (\neg c) \rightarrow x \cdot y \vee (\neg x)) \cdot ((a \vee (\neg a) \cdot (\neg b) \cdot (\neg c)) \cdot x \rightarrow y) = 1$$

Выражая импликацию через дизъюнкцию и отрицание и учитывая, что $(a \rightarrow b) = (\neg a \vee b)$, получаем:

$$\begin{aligned} & (\neg a \cdot \neg b \cdot \neg c \vee x \vee y \vee z) \cdot (\neg a \vee c \vee y) \cdot (\neg c \vee b \vee y \cdot \neg x) \cdot \\ & \cdot (\neg b \vee c \vee x \cdot y \vee \neg x) \cdot (\neg [(a \vee \neg a) \cdot (a \vee \neg b) \cdot (a \vee \neg c) \cdot x] \vee \\ & \vee y) = (\neg a \cdot \neg b \cdot \neg c \vee x \vee y \vee z) \cdot (\neg a \vee c \vee y) \cdot (\neg c \vee b \vee y \cdot \\ & \cdot \neg x) \cdot (\neg b \vee c \vee x \cdot y \vee \neg x) \cdot (\neg a \cdot b \vee \neg a \cdot c \vee \neg x \vee y) = 1. \end{aligned}$$

Подставляя в это тождество конкретные значения истинности $x = 1, y = 0, z = 0$, получаем:

$$\begin{aligned} & (\neg a \vee c) \cdot (\neg c \vee b) \cdot (\neg b \vee c) \cdot (\neg a \cdot b \vee \neg a \cdot c) = \\ & = (\neg a \cdot \neg c \vee \neg a \cdot b \vee c \cdot b) \cdot (\neg b \cdot \neg a \cdot c \vee c \cdot \neg a \cdot b \vee \neg a \cdot c) = \\ & = \neg a \cdot b \cdot c = 1. \end{aligned}$$

Отсюда следует, что $a = 0, b = 1, c = 1$.

Ответ на первый вопрос задачи: нужно заменить блоки b и c ; блока a не требует замены.

Ответ на второй вопрос задачи получите самостоятельно.

2. Решение логических задач табличным способом

При использовании этого способа условия, которые содержит задача, и результаты рассуждений фиксируются с помощью специально составленных таблиц.

Пример 6. В симфонический оркестр приняли на работу трех музыкантов: Брауна, Смита и Вессона, умеющих играть на скрипке, флейте, альте, кларнете, гобое и трубе.

Известно, что:

- 1) Смит самый высокий;
- 2) играющий на скрипке меньше ростом играющего на флейте;
- 3) играющие на скрипке и флейте и Браун любят пиццу;
- 4) когда между альтистом и трубачом возникает ссора, Смит мирит их;
- 5) Браун не умеет играть ни на трубе, ни на гобое.

На каких инструментах играет каждый из музыкантов, если каждый владеет двумя инструментами?

Решение. Составим таблицу и отразим в ней условия задачи, заполнив соответствующие клетки цифрами 0 и 1 в зависимости от того, ложно или истинно соответствующее высказывание.

Так как музыкантов трое, инструментов шесть и каждый владеет только двумя инструментами, получается, что каждый музыкант играет на инструментах, которыми остальные не владеют.

Из условия 4 следует, что Смит не играет ни на альте, ни на трубе, а из условий 3 и 5, что Браун не умеет играть на скрипке, флейте, трубе и гобое. Следовательно, инструменты Брауна – альт и кларнет. Занесем это в таблицу 4.6, а оставшиеся клетки столбцов «альт» и «кларнет» заполним нулями.

Таблица 4.6 – Условия задачи 6

Имя	Скрипка	Флейта	Альт	Кларнет	Гобой	Труба
Браун	0	0	1	1	0	0
Смит			0	0		0
Вессон			0	0		

Из таблицы 4.6 видно, что на трубе может играть только Вессон.

Из условий 1 и 2 следует, что Смит не скрипач. Так как на скрипке не играет ни Браун, ни Смит, то скрипачом является Вессон. Оба инструмента, на которых играет Вессон, теперь определены, поэтому остальные клетки строки «Вессон» можно заполнить нулями в таблице 4.7.

Таблица 4.7 – Инструменты, на которых играет Вессон

Имя	Скрипка	Флейта	Альт	Кларнет	Гобой	Труба
Браун	0	0	1	1	0	0
Смит	0		0	0		0
Вессон	1	0	0	0	0	1

Отсюда видно, что играть на флейте и на гобое может только Смит (табл. 4.8).

Таблица 4.8 – Инструменты, на которых играет Смит

Имя	Скрипка	Флейта	Альт	Кларнет	Гобой	Труба
Браун	0	0	1	1	0	0
Смит	0	1	0	0	1	0
Вессон	1	0	0	0	0	1

Ответ. Браун играет на альте и кларнете, Смит – на флейте и гобое, Вессон – на скрипке и трубе.

Пример 7. Три одноклассника – Влад, Тимур и Юра – встретились спустя 10 лет после окончания школы. Выяснилось, что один из них стал врачом, другой физиком, а третий юристом. Один полюбил туризм, другой бег, страсть третьего – регби.

Юра сказал, что на туризм ему не хватает времени, хотя его сестра – единственный врач в семье, заядлый турист. Врач сказал, что он разделяет увлечение коллеги.

Забавно, но у двоих из друзей в названиях их профессий и увлечений не встречается ни одна буква их имен.

Определите, кто чем любит заниматься в свободное время и у кого какая профессия.

Решение. Здесь исходные данные разбиваются на тройки (имя – профессия – увлечение).

Из слов Юры ясно, что он не увлекается туризмом и он не врач. Из слов врача следует, что он турист (табл. 4.9).

Таблица 4.9 – Условия задачи 7

Имя	Юра		
Профессия		Врач	
Увлечение		Туризм	

Буква «а», присутствующая в слове «врач», указывает на то, что Влад тоже не врач, следовательно, врач – Тимур. В его имени есть буквы «т» и «р», встречающиеся в слове «туризм», следовательно, второй из друзей, в названиях профессии и увлечения которого не встречается ни одна буква его имени, – Юра. Юра не юрист и не регбист, так как в его имени содержатся буквы «ю» и «р». Следовательно, окончательно имеем решение в таблице 4.10.

Ответ. Влад – юрист и регбист, Тимур – врач и турист, Юра – физик и бегун.

Таблица 4.10 – Решение задачи 7

Имя	Юра	Тимур	Влад
Профессия	Физик	Врач	Юрист
Увлечение	Бег	Туризм	Регби

Пример 8. Три дочери писательницы Дорис Кей – Джуди, Айрис и Линда, тоже очень талантливы. Они приобрели известность в разных видах искусств – пении, балете и кино. Все они живут в разных городах, поэтому Дорис часто звонит им в Париж, Рим и Чикаго.

Известно, что:

- 1) Джуди живет не в Париже, а Линда – не в Риме;
- 2) парижанка не снимается в кино;
- 3) та, кто живет в Риме, певица;
- 4) Линда равнодушна к балету.

Где живет Айрис и какова ее профессия?

Решение. Составим таблицу 4.11 и отразим в ней условия 1 и 4, заполнив клетки цифрами 0 и 1 в зависимости от того, ложно или истинно соответствующее высказывание.

Таблица 4.11 – Условия задачи 8

Париж	Рим	Чикаго		Пение	Балет	Кино
0			Джуди			
			Айрис			
	0		Линда		0	

Далее рассуждаем следующим образом. Так как Линда живет не в Риме, то, согласно условию 3, она не певица. В клетку, соответствующую строке «Линда» и столбцу «Пение», ставим 0.

Значит, Линда киноактриса, а Джуди и Айрис не снимаются в кино (табл. 4.12).

Согласно условию 2, парижанка не снимается в кино, следовательно, Линда живет не в Париже. Но она живет и не в Риме. Следовательно, Линда живет в Чикаго. Так как Линда и Джуди живут не в Париже, там живет Айрис. Джуди живет в Риме и, согласно условию 3, является певицей. А так как Линда киноактриса, то Айрис балерина.

Таблица 4.12 – Увлечения Линды

Париж	Рим	Чикаго		Пение	Балет	Кино
0			Джуди			0
			Айрис			0
	0		Линда	0	0	1

В результате постепенного заполнения получаем таблицу 4.13.

Таблица 4.13 – Решение задачи 8

Париж	Рим	Чикаго		Пение	Балет	Кино
0	0	1	Джуди	1	0	0
1	0	0	Айрис	0	1	0
0	0	1	Линда	0	0	1

Ответ. Айрис балерина. Она живет в Париже.

3. Решение логических задач с помощью рассуждений

Этим способом обычно решают несложные логические задачи.

Пример 9. Вадим, Сергей и Михаил изучают различные иностранные языки: китайский, японский и арабский. На вопрос, какой язык изучает каждый из них, один ответил: «Вадим изучает китайский, Сергей не изучает китайский, а Михаил не изучает арабский». Впоследствии выяснилось, что в этом ответе только одно утверждение верно, а два других ложны. Какой язык изучает каждый из молодых людей?

Решение. Имеется три утверждения:

- 1) Вадим изучает китайский;
- 2) Сергей не изучает китайский;
- 3) Михаил не изучает арабский.

Если верно первое утверждение, то верно и второе, так как юноши изучают разные языки. Это противоречит условию задачи, поэтому первое утверждение ложно.

Если верно второе утверждение, то первое и третье должны быть ложны. При этом получается, что никто не изучает китайский. Это противоречит условию, поэтому второе утверждение тоже ложно.

Остается считать верным третье утверждение, а первое и второе – ложными. Следовательно, Вадим не изучает китайский, китайский изучает Сергей.

Ответ. Сергей изучает китайский язык, Михаил – японский, Вадим – арабский.

Пример 10. В поездке пятеро друзей – Антон, Борис, Вадим, Дима и Гриша – познакомились с попутчицей. Они предложили ей отгадать их фамилии, причем каждый из них высказал одно истинное и одно ложное утверждение:

Дима сказал: «Моя фамилия – Мишин, а фамилия Бориса – Хохлов». Антон сказал: «Мишин – это моя фамилия, а фамилия Вадима – Белкин». Борис сказал: «Фамилия Вадима – Тихонов, а моя фамилия – Мишин». Вадим сказал: «Моя фамилия – Белкин, а фамилия Гриши – Чехов». Гриша сказал: «Да, моя фамилия Чехов, а фамилия Антона – Тихонов».

Какую фамилию носит каждый из друзей?

Решение. Обозначим высказывание «юноша по имени А носит фамилию Б» как A_B , где буквы А и Б соответствуют начальным буквам имени и фамилии.

Зафиксируем высказывания каждого из друзей:

D_M и B_X ;

A_M и B_B ;

V_T и B_M ;

V_B и G_C ;

G_C и A_T .

Допустим сначала, что истинно D_M . Но если истинно D_M , то у Антона и у Бориса должны быть другие фамилии, значит A_M и B_M ложно. Но если A_M и B_M ложны, то должны быть истинны V_B и V_T , но V_B и V_T одновременно истинными быть не могут.

Значит, остается другой случай: истинно B_X . Этот случай приводит к цепочке умозаключений:

B_X истинно $\rightarrow B_M$ ложно $\rightarrow V_T$ истинно $\rightarrow A_T$ ложно $\rightarrow G_C$ истинно $\rightarrow V_B$ ложно $\rightarrow A_M$ истинно.

Ответ. Борис – Хохлов, Вадим – Тихонов, Гриша – Чехов, Антон – Мишин, Дима – Белкин.

Пример 11. Министры иностранных дел России, США и Китая обсудили за закрытыми дверями проекты соглашения о полном разоружении, представленные каждой из стран. Отвечая затем на вопрос журналистов: «Чей именно проект был принят?», министры дали такие ответы:

Россия – «Проект не наш, проект не США»;

США – «Проект не России, проект Китая»;

Китай – «Проект не наш, проект России».

Один из них (самый откровенный) оба раза говорил правду; второй (самый скрытный) оба раза говорил неправду, третий (осторожный) один раз сказал правду, а другой раз – неправду.

Определите, представителями каких стран являются откровенный, скрытный и осторожный министры.

Решение. Для удобства записи пронумеруем высказывания дипломатов:

Россия – «Проект не наш» (1), «Проект не США» (2),

США – «Проект не России» (3), «Проект Китая» (4),

Китай – «Проект не наш» (5), «Проект России» (6).

Узнаем, кто из министров самый откровенный.

Если это российский министр, то из справедливости (1) и (2) следует, что победил китайский проект. Но тогда оба утверждения министра США тоже справедливы, чего не может быть по условию.

Если самый откровенный – министр США, то тогда вновь получаем, что победил китайский проект, значит оба утверждения российского министра тоже верны, чего не может быть по условию.

Получается, что наиболее откровенным был китайский министр. Действительно, из того, что (5) и (6) справедливы, следует, что победил российский проект. А тогда получается, что из двух утверждений российского министра первое ложно, а второе верно. Оба же утверждения министра США неверны.

Ответ. Откровеннее был китайский министр, осторожнее – российский, скрытнее – министр США.

4.3 Логические основы персонального компьютера

Логические элементы. Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: «1» и «0».

Из этого следует *два вывода*:

- одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных;
- на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, следовательно, уменьшить число элементарных логических элементов, из десятков тысяч которых состоят основные узлы компьютера.

Данные и команды представляются в виде двоичных последовательностей различной структуры и длины. Существуют различные физические способы кодирования двоичной информации. В электронных устройствах компьютера двоичные единицы чаще

всего кодируются более высоким уровнем напряжения, чем двоичные нули (или наоборот), например, рисунок 4.1.

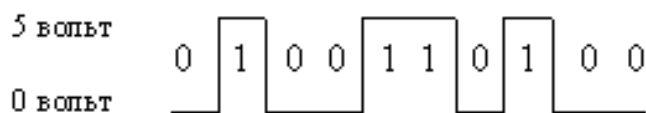


Рисунок 4.1 – Кодирование в электронных устройствах

Логический элемент компьютера – это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы *И*, *ИЛИ*, *НЕ*, *И–НЕ*, *ИЛИ–НЕ* и другие (называемые также *вентиллями*), а также *триггер*.

С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентилей бывает от двух до восьми входов и один или два выхода.

Чтобы представить два логических состояния – «1» и «0» в вентиллях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Например, +5 вольт и 0 вольт.

Высокий уровень обычно соответствует значению «истина» («1»), а низкий – значению «ложь» («0»).

Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем.

Работу логических элементов описывают с помощью таблиц истинности.

Таблица истинности – это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний

Логические операторы в запросах. Существует *четыре вида логических операторов в запросах*:

- Логическое И – оператор AND.
- Логическое ИЛИ – оператор OR.

- Логическое НЕ – оператор NOT.
- Логическое БЛИЗКО – оператор NEAR.

Графическое изображение этих операторов представлено на рисунке 4.2.

Оператор AND: книга AND история (рис. 4.2).

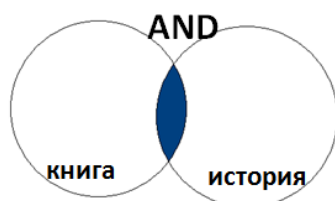


Рисунок 4.2 – Графическое представление логического оператора AND

Оператор OR: книга OR история (рис. 4.3).



Рисунок 4.3 – Графическое представление логического оператора OR

Оператор NOT: книга NOT история (рис. 4.4).



Рисунок 4.4 – Графическое представление логического оператора NOT

Сложный логический оператор: книга AND история NOT смешная (рис. 4.5).



Рисунок 4.5 – Графическое представление сложного логического оператора

Логические схемы. Схема «И» реализует конъюнкцию двух или более логических значений. Условное обозначение на структурных схемах схемы «И» с двумя входами представлено на рисунке 4.6.

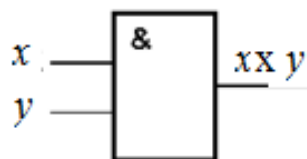


Рисунок 4.6 – Логическая схема «И»

Таблица истинности схемы «И» представлена ниже (табл. 4.14).

Таблица 4.14 – Таблица истинности схемы «И»

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Единица на выходе схемы «И» будет тогда и только тогда, когда на всех входах будут единицы. Когда хотя бы на одном входе будет ноль, на выходе также будет ноль.

Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \wedge y$, (читается, как « x и y »). Операция конъюнкции на структурных схемах обозначается знаком «&» (читается как «амперсэнд»), являющимся сокращенной записью английского слова and.

Схема «ИЛИ» реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном входе схемы «ИЛИ» будет единица, на ее выходе также будет единица.

Условное обозначение на структурных схемах схемы «ИЛИ» с двумя входами представлено на рисунке 4.7. Знак «1» на схеме – от устаревшего обозначения дизъюнкции как « $>=1$ » (т.е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1). Связь между выходом z этой схемы и входами x и y описывается соотношением: $z = x \vee y$ (читается как « x или y ») (рис. 4.7).

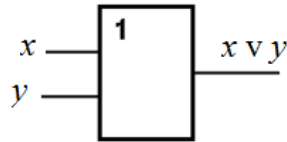


Рисунок 4.7 – Логическая схема «ИЛИ»

Таблица истинности схемы «ИЛИ» представлена ниже (табл. 4.15).

Таблица 4.15 – Таблица истинности схемы «ИЛИ»

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Схема «НЕ» (инвертор) реализует операцию отрицания. Связь между входом x этой схемы и выходом z можно записать соотношением $z = \bar{x}$, где \bar{x} читается как «не x » или «инверсия x ».

Если на входе схемы 0, то на выходе 1. Когда на входе 1, на выходе 0. Условное обозначение на структурных схемах инвертора показано на рисунке 4.8.

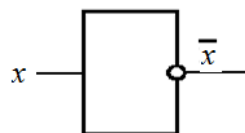


Рисунок 4.8 – Схема инвертора

Таблица истинности схемы «НЕ» представлена ниже (табл. 4.16).

Таблица 4.16 – Таблица истинности схемы «НЕ»

x	\bar{x}
0	1
1	0

Схема «И – НЕ» состоит из элемента «И» и инвертора и осуществляет отрицание результата схемы «И». Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \wedge y}$, где $\overline{x \wedge y}$ читается как «инверсия x и y ». Условное обозначение на структурных схемах схемы «И – НЕ» с двумя входами представлено на рисунке 4.9.

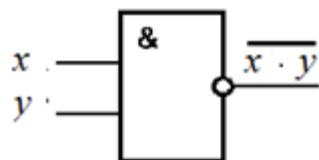


Рисунок 4.9 – Схема «И – НЕ» с двумя входами

Таблица истинности схемы «И – НЕ» представлена ниже (табл. 4.17).

Таблица 4.17 – Таблица истинности схемы «И – НЕ»

x	y	$\overline{x \wedge y}$
0	0	1
0	1	1
1	0	1
1	1	0

Схема «ИЛИ – НЕ» состоит из элемента «ИЛИ» и инвертора и осуществляет отрицание результата схемы «ИЛИ». Связь между выходом z и входами x и y схемы записывают следующим образом: $z = \overline{x \vee y}$, где $\overline{x \vee y}$, читается как «инверсия x или y ». Условное обозначение на структурных схемах схемы «ИЛИ – НЕ» с двумя входами представлено на рисунке 4.10.

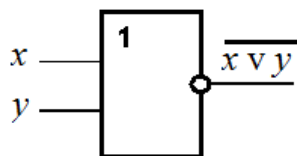


Рисунок 4.10 – Схемы «ИЛИ – НЕ» с двумя входами

Таблица истинности схемы «ИЛИ – НЕ» представлена ниже (табл. 4.18).

Таблица 4.18 – Таблица истинности схемы «ИЛИ – НЕ»

x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

Триггер – это электронная схема, широко применяемая в регистрах компьютера для надежного запоминания одного разряда двоичного кода. Триггер имеет два устойчивых состояния, одно из которых соответствует двоичной единице, а другое – двоичному нулю.

Термин *триггер* происходит от английского слова *trigger* – защелка, спусковой крючок. Для обозначения этой схемы в английском языке чаще употребляется термин *flip-flop*, что в переводе означает *хлопанье*. Это звукоподражательное название электронной схемы указывает на ее способность почти мгновенно переходить («перебрасываться») из одного электрического состояния в другое и наоборот.

Самый распространенный тип триггера – так называемый *RS-триггер* (*S* и *R* соответственно от английских *set* – установка, и *reset* – сброс). Условное обозначение триггера представлено на рисунке 4.11.

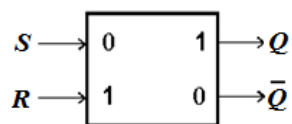


Рисунок 4.11 – Условное обозначение триггера

Он имеет два симметричных входа *S* и *R* и два симметричных выхода *Q* и \bar{Q} , причем выходной сигнал *Q* является логическим отрицанием сигнала \bar{Q} .

На каждый из двух входов *S* и *R* могут подаваться входные сигналы в виде кратковременных импульсов ($\text{—}\overline{\text{1}}\text{—}$).

На рисунке 4.12 представлены реализация триггера с помощью вентиля «ИЛИ – НЕ» и соответствующая таблица истинности.

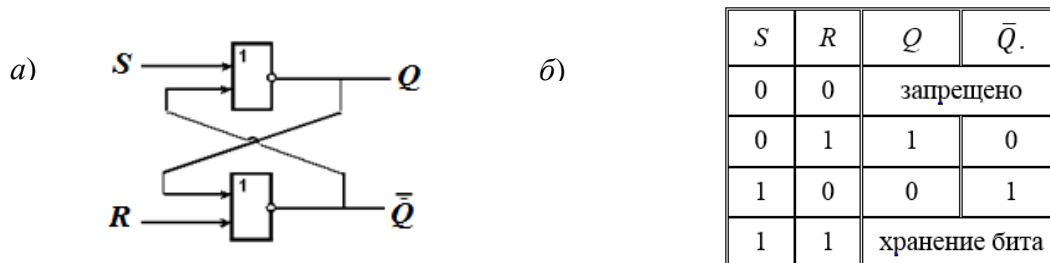


Рисунок 4.12 – Таблица истинности для реализации триггера:
 а – реализация триггера; б – таблица истинности

Поскольку один триггер может запомнить только один разряд двоичного кода, то для запоминания байта нужно 8 триггеров, для запоминания килобайта соответственно $8 \cdot 2^{10} = 8192$ триггера. Современные микросхемы памяти содержат миллионы триггеров.

Сумматор – это электронная логическая схема, выполняющая суммирование двоичных чисел.

Сумматор служит прежде всего центральным узлом арифметико-логического устройства компьютера, однако он находит применение также и в других устройствах машины.

Многоразрядный двоичный сумматор, предназначенный для сложения многоразрядных двоичных чисел, представляет собой комбинацию одноразрядных сумматоров. Условное обозначение одноразрядного сумматора представлено на рисунке 4.13.

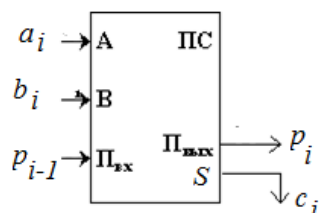


Рисунок 4.13 – Условное обозначение одноразрядного сумматора

При сложении чисел А и В в одном i -м разряде приходится иметь дело с тремя цифрами:

- цифра a_i первого слагаемого;
- цифра b_i второго слагаемого;
- перенос p_{i-1} из младшего разряда.

В результате сложения получаются две цифры:

- цифра c_i для суммы;
- перенос p_i из данного разряда в старший.

Таким образом, одноразрядный двоичный сумматор есть устройство с тремя входами и двумя выходами, работа которого может быть описана таблицей истинности (табл. 4.19).

Таблица 4.19 – Таблица истинности одноразрядного двоичного сумматора

Входы			Выходы	
Первое слагаемое	Второе слагаемое	Перенос	Сумма	Перенос
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Если требуется складывать двоичные слова длиной в два и более бит, то можно использовать последовательное соединение таких сумматоров, причем для двух соседних сумматоров выход переноса одного сумматора является входом для другого.

Вопросы и задания для самоконтроля

1. Как называется логическое умножение?
2. Какое из обозначений: а) НЕ; б) $|$; в) \neg ; г) NOT не применяется для отрицания?
3. У какой из логических функций следующая таблица истинности?

A	B	?
0	0	0
1	0	1
0	1	1
1	1	1

- а) «Солнце есть спутник Земли»;
- б) « $2 + 3 = 4$ »;
- в) «сегодня отличная погода»;
- г) «в романе Л.Н. Толстого «Война и мир» 3 432 536 слов»;
- д) «Санкт-Петербург расположен на Неве»;
- е) «музыка Баха слишком сложна»;
- ж) «первая космическая скорость равна 7,8 км/с»;
- з) «железо – металл»;
- и) «если один угол в треугольнике прямой, то треугольник будет тупоугольным»;
- к) «если сумма квадратов двух сторон треугольника равна квадрату третьей, то он прямоугольный».

15. Укажите, какие из высказываний предыдущего упражнения истинны, какие – ложны, а какие относятся к числу тех, истинность которых трудно или невозможно установить.

16. Приведите примеры истинных и ложных высказываний:

- а) из арифметики;
- б) из физики;
- в) из биологии;
- г) из информатики;
- д) из геометрии;
- е) из жизни.

17. Сформулируйте отрицания следующих высказываний или высказывательных форм:

- а) «Эльбрус – высочайшая горная вершина Европы»;
- б) « $2 > 5$ »;
- в) « $10 < 7$ »;
- г) «все натуральные числа целые»;
- д) «через любые три точки на плоскости можно провести окружность»;
- е) «теннисист Кафельников не проиграл финальную игру»;
- ж) «мишень поражена первым выстрелом»;
- з) «это утро ясное и теплое»;
- и) «число n делится на 2 или на 3»;
- к) «этот треугольник равнобедренный и прямоугольный»;
- л) «на контрольной работе каждый ученик писал своей ручкой».

18. Определите, какие из высказываний в следующих парах являются отрицаниями друг друга, а какие нет:

- а) « $5 < 10$ », « $5 > 10$ »;
- б) « $10 > 9$ », « $10 < = 9$ »;
- в) «мишень поражена первым выстрелом», «мишень поражена вторым выстрелом»;

- г) «машина останавливалась у каждого из двух светофоров», «машина не останавливалась у каждого из двух светофоров»;
- д) «человечеству известны все планеты Солнечной системы», «в Солнечной системе есть планеты, неизвестные человечеству»;
- е) «существуют белые слоны», «все слоны серые»;
- ж) «кит – млекопитающее», «кит – рыба»;
- з) «неверно, что точка A не лежит на прямой a », «точка A лежит на прямой a »;
- и) «прямая a параллельна прямой b », «прямая a перпендикулярна прямой b »;
- к) «этот треугольник равнобедренный и прямоугольный», «этот треугольник не равнобедренный или он не прямоугольный».

19. Алеша, Боря и Гриша нашли в земле сосуд. Каждый из них высказал по два предположения.

Алеша: «Это сосуд греческий, V века».

Боря: « Это сосуд финикийский, III века».

Гриша: « Это сосуд не греческий, IV века».

Преподаватель по истории сказал студентам, что каждый из них прав только в одном из двух своих предположений.

Где и в каком веке был изготовлен сосуд?

20. В нарушении правил обмена валюты подозревают четыре работника банка – A , B , C и D . Известно, что:

если A нарушил, то и B нарушил правила обмена валюты;

если B нарушил, то и C нарушил или A не нарушал;

если D не нарушил, то A нарушил, а C не нарушал;

если D нарушил, то и A нарушил.

Кто из подозреваемых нарушил правила обмена валюты?

21. На вопрос «Какая завтра будет погода?» синоптик ответил:

«если не будет ветра, то будет пасмурная погода без дождя»;

«если будет дождь, то будет пасмурно и без ветра»;

«если будет пасмурная погода, то будет дождь и не будет ветра».

Подумав немного, синоптик уточнил, что его три высказывания можно лаконично записать в виде одного составного высказывания. Сформулируйте его, решив задачу с помощью логических операций.

22. Определите, кто из подозреваемых участвовал в преступлении, если известно:

если Иванов не участвовал или Петров участвовал, то Сидоров участвовал;

если Иванов не участвовал, то Сидоров не участвовал.

23. Виктор, Роман, Леонид и Сергей заняли на олимпиаде по физике четыре первых места. Когда их спросили о распределении мест, они дали три таких ответа:

«Сергей – первый, Роман – второй»;

«Сергей – второй, Виктор – третий»;

«Леонид – второй, Виктор – четвертый».

Известно, что в каждом ответе только одно утверждение истинно. Как распределились места?

24. Андрей, Боря, Женя, Ольга, Роза, Полина, Дима и Серафима – друзья. В это воскресенье Андрей отправился на концерт, Боря провел вечер с Ольгой, Женя так и не встретил Розу, Полина побывала в кино, Роза посмотрела спектакль в театре. Какая-то пара посетила художественную выставку. Мы не знаем, где именно были Дима и Серафима, но известно, что каждый юноша из этой компании был в театре, на выставке, на концерте или в кино с одной из девушек – Ольгой, Розой, Полиной или Серафимой. Определите, где и с кем побывала каждая девушка из этой компании.

25. В финале турнира Российской армии по шахматам встретились представители шести воинских званий (майор, капитан, лейтенант, старшина, сержант и ефрейтор) и разных специальностей (летчик, танкист, артиллерист, минометчик, сапер и связист). Определите специальность и звание каждого из шахматистов по следующим данным:

в первом туре лейтенант выиграл у летчика, майор – у танкиста, а сержант – у минометчика;

во втором туре капитан выиграл у танкиста;

в третьем и четвертом турах минометчик из-за болезни не участвовал в турнире, поэтому свободными от игры оказались капитан и ефрейтор;

в четвертом туре майор выиграл у связиста;

победителями турнира оказались лейтенант и майор, а хуже всех выступил сапер.

5 ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

Кодирование информации – это процесс формирования определенного представления информации. В более узком смысле под термином «кодирование» часто понимают переход от одной формы представления информации к другой, более удобной для хранения, передачи или обработки.

Существует три основных способа кодирования информации.

- *Графический* – с помощью рисунков или значков (рис. 5.1).



Рисунок 5.1 – Пример графического способа кодирования

- *Числовой* – с помощью чисел (систем счисления) (рис. 5.2).



Рисунок 5.2 – Пример числового способа кодирования

- *Символьный* – с помощью символов того же алфавита, что и исходный текст (рис. 5.3).

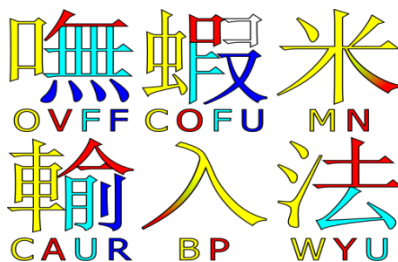


Рисунок 5.3 – Пример символьного способа кодирования

Информация в компьютере представлена в *двоичном коде*, алфавит которого состоит из двух цифр – **0** и **1** (рис.5.4).

Двоичное кодирование информации в компьютере

Вид информации	Двоичный код
Числовая	
Текстовая	
Графическая	
Звуковая	
Видео	

Рисунок 5.4 – Двоичное кодирование информации в компьютере

Эти два символа принято называть *двоичными цифрами*, или *битами*.

С помощью только двух цифр 0 и 1 можно закодировать любое сообщение. Это явилось причиной того, что в компьютере обязательно должно быть организовано два важных процесса: кодирование и декодирование.

Кодирование – это перевод информации в удобную для передачи, обработки или хранения форму с помощью некоторого кода.

Декодирование – это процесс восстановления содержания закодированной информации.

С точки зрения технической реализации использование двоичной системы счисления для кодирования информации оказалось намного более простым, чем применение других способов.

Действительно, удобно кодировать информацию в виде последовательности нулей и единиц, если представить эти значения как два возможных устойчивых состояния электронного элемента:

0 – отсутствие электрического сигнала;

1 – наличие электрического сигнала.

Эти состояния легко различать.

Недостаток двоичного кодирования – *длинные коды*.

Но в технике легче иметь дело с большим количеством простых элементов, чем с небольшим числом сложных.

Способы кодирования и декодирования информации в компьютере в первую очередь зависят от вида информации, а именно от того, что должно кодироваться: числа, текст, графические изображения или звук. Рассмотрим режимы кодирования информации различного вида в компьютере более подробно.

5.1 Представление текстовой информации

Любой текст состоит из последовательности символов. Символами могут быть буквы, цифры, знаки препинания, знаки математических действий, круглые и квадратные скобки и т.д. Обратите особое внимание на символ «пробел», который используется для разделения слов и предложений между собой. Хотя на бумаге или экране монитора «пробел» – это пустое, свободное место, тем не менее это тоже символ. На клавиатуре компьютера символу «пробел» соответствует специальная клавиша.

Текстовая информация, как и любая другая, хранится в памяти компьютера в двоичном виде. Для этого каждому символу ставится в соответствие некоторое неотрицательное число, называемое *кодом* символа, и это число записывается в память ПК в двоичном виде. Конкретное соответствие между символами и их кодами называется *системой кодировки*. В современных ПК, в зависимости от типа операционной системы и конкретных прикладных программ, используются 8-разрядные, 16- и 32-разрядные коды символов. Использование 8-разрядных кодов позволяет закодировать 256 различных знаков, этого вполне достаточно для представления многих символов, используемых на практике. При такой кодировке для кода символа достаточно выделить в памяти один байт.

При восьмиразрядном двоичном кодировании текстовой информации каждому символу ставится в соответствие своя уникальная последовательность из восьми нулей и единиц, свой уникальный код от 00000000 до 11111111 (десятичный код от 0 до 255).

Присвоение символу конкретного двоичного кода – это вопрос соглашения, которое фиксируется в кодовой таблице. Первые 33 кода (от 0 до 32) соответствуют не символам, а операциям (перевод строки, ввод пробела и т.д.). Коды от 33 до 127 являются интернациональными и соответствуют символам латинского алфавита, цифрам, знакам арифметических операций и знакам препинания.

Таблица, в которой всем символам компьютерного алфавита поставлены в соответствие порядковые номера (коды), называется *таблицей кодировки*.

Коды с 128 по 255 являются национальными, т.е. в национальных кодировках одному и тому же коду соответствуют раз-

личные символы. Поскольку в настоящее время существует 5 различных кодовых таблиц для русских букв, поэтому тексты, созданные в одной кодировке, не будут правильно отображаться в другой.

Традиционно для кодирования одного символа используется количество информации, равное 1 байту (1 байт = 8 битов).

Хронологически одним из первых стандартов кодирования русских букв на компьютере был код **КОИ – 8** («*Код обмена информацией – 8-битный*»). Эта кодировка применяется в компьютерах с операционной системой UNIX.

Наиболее распространенная кодировка – это стандартная кириллическая кодировка Microsoft Windows, обозначаемая сокращением **CP 1251** («*CP*» означает «*Code Page*»). Все Windows-приложения, работающие с русским языком, поддерживают эту кодировку.

Для работы в среде операционной системы MS-DOS используется «альтернативная» кодировка, в терминологии фирмы Microsoft – кодировка CP 866.

Фирма Apple разработала для компьютеров Macintosh свою собственную кодировку русских букв (*Mac*).

Международная организация по стандартизации (**International Standards Organization, ISO**) утвердила в качестве стандарта для русского языка еще одну кодировку под названием **ISO 8859-5**.

Эту кодировку поддерживает платформа Microsoft Windows&Office97.

В персональных компьютерах обычно используется система кодировки **ASCII** (*American Standard Code for Information Interchange* – американский стандартный код для обмена информацией). Он ставит в соответствие каждому символу семиразрядный двоичный код. Легко определить, что в коде ASCII можно представить 128 символов.

В системе ASCII закреплены две таблицы кодирования – *базовая* и *расширенная*. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255.

Коды от 0 до 32 – функциональные клавиши, *коды от 33 до 127* – буквы английского алфавита, знаки математических операций и т.д.

На рисунке 5.5 представлена кодовая таблица ASCII (*American Standard Code for Information Interchange*).

32 пробел	48 0	64 @	80 P	96 `	112 p
33 !	49 1	65 A	81 Q	97 a	113 q
34 "	50 2	66 B	82 R	98 b	114 r
35 #	51 3	67 C	83 S	99 c	115 s
36 \$	52 4	68 D	84 T	100 d	116 t
37 %	53 5	69 E	85 U	101 e	117 u
38 &	54 6	70 F	86 V	102 f	118 v
39 ^	55 7	71 G	87 W	103 g	119 w
40 (56 8	72 H	88 X	104 h	120 x
41)	57 9	73 I	89 Y	105 i	121 y
42 *	58 :	74 J	90 Z	106 j	122 z
43 +	59 ;	75 K	91 [107 k	123 {
44 ,	60 <	76 L	92 \	108 l	124
45 -	61 =	77 M	93]	109 m	125 }
46 .	62 >	78 N	94 ^	110 n	126 ~
47 /	63 ?	79 O	95 _	111 o	127

Рисунок 5.5 – Кодовая таблица ASCII

Цифры кодируются по стандарту ASCII в двух случаях – при вводе-выводе и когда они встречаются в тексте. Если цифры участвуют в вычислениях, то они преобразуются в другой двоичный код.

Пример 1. Закодировать число 57.

Решение. При использовании в тексте каждая цифра будет представлена своим кодом в соответствии с таблицей ASCII. В двоичной системе это 0011010100110111.

При использовании в вычислениях код этого числа будет получен по правилам перевода в двоичную систему и получим 00111001.

Пример 2. В таблице 5.1 представлена часть кодовой таблицы ASCII.

Таблица 5.1 – Часть кодовой таблицы ASCII

Символ	1	5	A	B	Q	a	b
Десятичный код	49	53	65	66	81	97	98
Шестнадцатеричный код	31	35	41	42	51	61	62

Определить шестнадцатеричный код символа «q».

Решение. В кодовой таблице ASCII все заглавные латинские буквы A–Z расположены по алфавиту. Следовательно, раз-

ница кодов букв «q» и «a» равна разнице кодов букв «Q» и «A», то есть, $51_{16} - 41_{16} = 10_{16}$. Тогда шестнадцатеричный код символа «q» равен коду буквы «a» плюс 10_{16} . Следовательно, имеем $61_{16} + 10_{16} = 71_{16}$.

По причине ограниченности набора кодов (256) возникла система, основанная на 16-разрядном кодировании символов, которая получила название универсальной – **UNICODE**. Стандарт Unicode отводит на каждый символ не один байт, а два, и поэтому с его помощью можно закодировать не 256 символов, а $2^{16} = 65536$ различных символов. Таким образом, шестнадцать разрядов позволяют обеспечить уникальные коды для 65 536 различных символов – этого поля вполне достаточно для размещения в одной таблице символов большинства различных языков.

Несмотря на тривиальную очевидность такого подхода, простой механический переход на данную систему долгое время сдерживался из-за недостатков ресурсов средств вычислительной техники (в системе кодирования UNICODE все текстовые документы становятся автоматически вдвое длиннее). Во второй половине 90-х годов технические средства достигли необходимого уровня обеспечения ресурсами, поэтому начался постепенный перевод документов и программных средств на универсальную систему кодирования.

Сегодня это самая распространенная текстовая кодировочная система.

5.2 Представление числовой информации в компьютере

Основное отличие числовых данных от символьных заключается в том, что над числами, кроме операции сравнения, производятся разнообразные математические операции: сложение, умножение, извлечение корня, вычисление логарифма.

Целые числа в памяти компьютера могут храниться в двух форматах: с фиксированной запятой и плавающей запятой.

Число в формате с плавающей запятой занимает

4 байта (число обычной точности);

8 байтов (число двойной точности).

Пример целого числа с плавающей запятой: $2, = 0,2 \cdot 10^1 = 200, \cdot 10^{-2}$.

Память компьютера представляет собой совокупность ячеек (табл. 5.2).

Таблица 5.2 – Память компьютера

Байты							ячейка
1	1	1	1	1	1	1	
1	1	1	1	1	1	1	
1	0	1	0	1	1	1	
1	1	1	1	1	1	1	
1	1	1	0	1	1	1	
1	1	1	1	1	1	0	

Ячейка – это часть памяти компьютера, вмещающая в себя информацию, доступную для обработки отдельной командой процессора.

Содержимое ячейки памяти называется *машинным словом*.

Количество информации, хранящейся в ЭВМ, измеряется ее «объемом», который выражается в *битах* (от английского **binary digit** – двоичная цифра).

Ячейка памяти разделяется на разряды, в каждом из которых хранится разряд числа (рис. 5.6). Разряд ячейки памяти также называют битом.

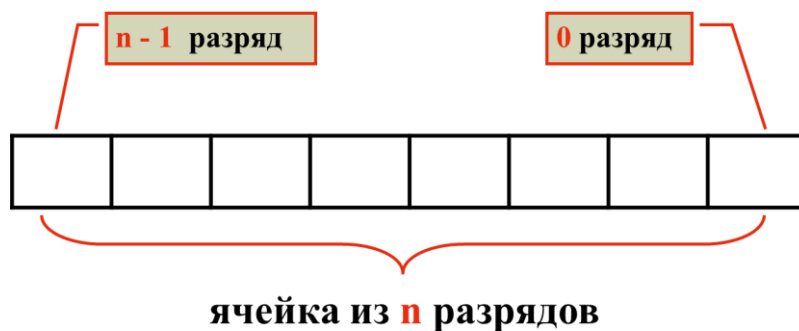


Рисунок 5.6 – Ячейка памяти

8 бит = 1 байт.

Байт – основная единица представления данных. Под байтом (от английского **byte** – *слог*) понимается часть машинного слова, состоящая из 8 бит, обрабатываемая в ЭВМ как одно целое (рис. 5.6).

В ячейке памяти компьютера числа могут быть представлены в следующих форматах.

Байт = 8 бит:

70



Полуслово = 2 байта = 16 бит:

158 70



Слово = 4 байта = 32 бита:

31 24 23 16 15 8 70



Двойное слово = 8 байт = 64 бита:

63 56 55

8 70



Целые числа – это простейшие числовые типы данных, с которыми оперируют ЭВМ.

Специальные типы для целых чисел вводятся:

- для эффективного расходования памяти;
- повышения быстродействия;
- введения операции деления нацело с остатком;
- решения задач экономического характера;
- обозначения даты и времени;
- нумерации различных объектов.

При кодировании целых чисел в компьютере важно указать тип числа: знаковый или беззнаковый.

Для **беззнакового целого типа** числа имеем: минимальное число – 00000000 и максимальное число: 11111111.

Учитывая, что

$$11111111_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 255_{10},$$

можно утверждать, что в байте (8 разрядов) можно представить беззнаковые числа от 0 до 255.

Диапазоны целых чисел без знака в N разрядной сетке представлены на рисунке 5.7.

Числа без знака			
N	8	16	32
MAX	255 ($2^8 - 1$)	65 535 ($2^{16} - 1$)	4 294 967 295 ($2^{32} - 1$)
MIN	0		

Рисунок 5.7 – Диапазоны значений целых чисел без знака

Пример 3. Представление числа $72_{10} = 1001000_2$ в ПК. В **однобайтовом** формате имеет вид:

Номер разрядов	7	6	5	4	3	2	1	0
Биты числа	0	1	0	0	1	0	0	0

Пример 4. Это же число в **двухбайтовом** формате:

Номер разрядов	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты числа	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

Пример 5. Число 65535 в **двубайтовом** формате:

Номер разрядов	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты числа	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Знаковый тип целых чисел. Целые числа со знаком обычно занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа. Знак «плюс» кодируется нулем (рис. 5.8), а «минус» – единицей.

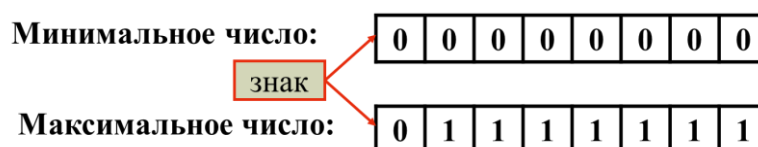


Рисунок 5.8 – Представление знаковых положительных чисел в компьютере

Так как $1111111_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 127_{10}$, то в байте (8 разрядов) можно представить знаковые положительные числа от 0 до 127.

Диапазон допустимых значений для знаковых типов от -2^{k-1} до $2^{k-1}-1$, где k – количество разрядов в ячейке.

На рисунке 5.9 представлены максимальные и минимальные значения для целых N -разрядных чисел со знаком.

Числа со знаком			
N	8	16	32
MAX	127	32 767	2 147 483 647
MIN	- 128	- 32 768	- 2 147 483 648

Рисунок 5.9 – Диапазон представления целых чисел со знаком

Алгоритм представления целого числа со знаком плюс в памяти компьютера:

1. Перевести число в двоичную систему счисления.
2. Нарисовать k -разрядную сетку.
3. Указать код знака «+» в старшем разряде.
4. Записать число в разрядную сетку, начиная с младшего разряда.
5. Заполнить оставшиеся разряды нулями.

Пример 6. Представление числа 54_{10} в памяти компьютера.

$$54_{10} = 110110_2.$$

$$k = 8 \quad \boxed{0 \mid 0 \mid 1 \mid 1 \mid 0 \mid 1 \mid 1 \mid 0}$$

$$k = 16 \quad \boxed{0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 1 \mid 1 \mid 0 \mid 1 \mid 1 \mid 0}$$

Пример 7. Представление числа 200 в памяти компьютера.
 $200 = 11001000_2.$

$$k = 8 \quad \boxed{1 \mid 1 \mid 0 \mid 0 \mid 1 \mid 0 \mid 0 \mid 0}$$

$$k = 16 \quad \boxed{0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 0 \mid 1 \mid 1 \mid 0 \mid 0 \mid 1 \mid 0 \mid 0 \mid 0}$$

Алгоритм представления целого числа со знаком минус в памяти компьютера:

1. Перевести модуль числа в двоичную систему счисления.
2. Записать число в прямом коде в n -двоичных разрядах.
3. Получить обратный код числа, для этого значения всех битов инвертировать (все единицы заменить на нули и все нули заменить на единицы).
4. Найти дополнительный код числа, прибавив к обратному коду единицу.
5. Нарисовать k -разрядную сетку.
6. Записать число в разрядную сетку.

Арифметические действия над целыми числами. В ПК в целях упрощения выполнения арифметических операций применяются специальные коды для представления целых чисел:

- прямой код числа;
- обратный код числа;
- дополнительный код числа.

Разряды числа в коде жестко связаны с разрядной сеткой (8, 16, 32, 64 разряда). Для записи *кода знака* числа в разрядной сетке отводится фиксированный разряд.

Знаковым разрядом является *старший разряд* в разрядной сетке (рис. 5.10)

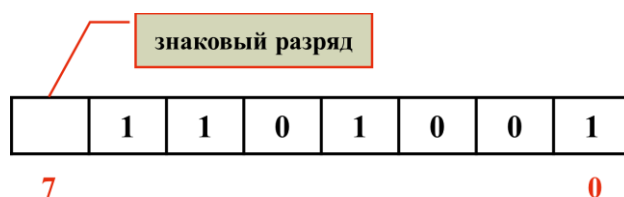


Рисунок 5.10 – Знаковый разряд в разрядной сетке

Прямой код двоичного числа совпадает по изображению с записью самого числа.

Значение *знакового разряда* для положительных чисел равно 0, а для отрицательных чисел равно 1.

Пример 8. Значения знакового разряда для положительных и отрицательных чисел (прямой код)

+ 1101	0	0	0	0	1	1	0	1
– 1101	1	0	0	0	1	1	0	1

Обратный код для положительного числа совпадает с прямым кодом.

Для отрицательного числа все цифры числа заменяются на противоположные (1 на 0, 0 на 1), а в знаковый разряд заносится единица.

Пример 9. Значения знакового разряда для положительных и отрицательных чисел (прямой и обратный код).

+ 1101	0	0	0	0	1	1	0	1	- прямой код
	0	0	0	0	1	1	0	1	- обратный код

- 1101	1	0	0	0	1	1	0	1	- прямой код
	1	1	1	1	0	0	1	0	- обратный код

Дополнительный код для положительного числа совпадает с прямым кодом (рис. 5.11).

+1101

Прямой код	Обратный код	Дополнительный код
00001101	00001101	00001101

Рисунок 5.11 – Пример прямого, обратного и дополнительного кодов для положительного числа

Для отрицательного числа дополнительный код образуется путем получения обратного кода и добавлением к младшему разряду единицы (рис. 5.12).

-1101

Прямой код	Обратный код	Дополнительный код
10001101	11110010	11110011

Рисунок 5.12 – Пример прямого, обратного и дополнительного кодов для отрицательного числа

Пример 10. Дополнительный код числа (-117) для 8-разрядной ячейки.

Решение. Однобайтовое представление числа:

Прямой код	11110101
Обратный код	10001010
Дополнительный код	10001011

Пример 11. Дополнительный код числа (-117) для 16-разрядной ячейки.

Решение. Двухбайтовое представление:

Прямой код	10000000 01110101
Обратный код	11111111 10001010
Дополнительный код	11111111 10001011

Все *целые отрицательные числа* в компьютере представляются дополнительным кодом.

Представление арифметических операций в компьютере. В большинстве компьютеров операция *вычитания* не используется. Вместо нее производится сложение уменьшаемого с обратным или дополнительным кодом вычитаемого. Это позволяет существенно упростить конструкцию арифметико-логического устройства (АЛУ).

При суммировании складываются все разряды, включая разряд знака. Так как знаковые разряды положительных слагаемых равны нулю, разряд знака суммы тоже равен нулю.

Пример 12. Суммирование чисел в компьютере.

Накапливающий сумматор	Множитель
000000000000	101101
+ 110011	101100
110011	Сдвиг на две позиции влево
+ 110011	101000
11111111	Сдвиг на одну позицию влево
+ 110011	100000
1010010111	Сдвиг на одну позицию влево
+ 110011	000000
100011110111	

При сложении может возникнуть ситуация, когда старшие разряды результата операции не помещаются в отведенной для него области памяти. Такая ситуация называется *переполнением*

разрядной сетки формата числа. Для обнаружения переполнения и оповещения о возникшей ошибке в компьютере используются специальные средства.

Пример 13. Сложение чисел в компьютере.

Десятичная запись

Двоичные коды

$$\begin{array}{r} 3 \\ + 7 \\ \hline 10 \end{array}$$

$$\begin{array}{r} 0\ 000011 \\ + 0\ 000011 \\ \hline 0\ 0001010 \end{array}$$

Десятичная: $15_{10} + 6_{10}$

Двоичная: $1111_2 + 110_2$

$$\begin{array}{r} 15 \\ + 6 \\ \hline 21 \end{array}$$

$$\begin{array}{r} 0\ 100011 \\ + 0\ 110001 \\ \hline 1\ 010010 \end{array} \text{ переполнение}$$

Во многих компьютерах *умножение* производится как последовательность сложений и сдвигов. Для этого в АЛУ имеется регистр, называемый *накапливающим сумматором*, который до начала выполнения операции содержит число 0. В процессе выполнения операции в нем поочередно размещаются множимое и результаты промежуточных сложений, а по завершении операции – окончательный результат.

Другой регистр АЛУ, участвующий в выполнении этой операции, вначале содержит множитель. Затем по мере выполнения сложений содержащееся в нем число уменьшается, пока не достигнет нулевого значения.

Деление для компьютера является трудной операцией. Обычно оно реализуется путем многократного прибавления к делимому дополнительного кода делителя.

5.3 Машинное представление вещественных чисел

Вещественными числами (в отличие от целых) в компьютерной технике называются числа, имеющие дробную часть. При их написании вместо запятой принято писать точку. Так, например, число 5 – целое, а числа 5,1 и 5,0 – вещественные. Для удобства отображения чисел, принимающих значения из достаточно широкого диапазона (то есть, как очень маленьких, так и очень

больших), используется форма записи чисел с порядком основания системы счисления. Например, десятичное число 1,25 можно в этой форме представить так: $1,25 \cdot 10^0 = 0,125 \cdot 10^1 = 0,0125 \cdot 10^2 = 12,5 \cdot 10^{-1} = 125,0 \cdot 10^{-2} = 1250,0 \cdot 10^{-3} = \dots$

Любое число N в системе счисления с основанием q можно записать в виде $N = M \cdot q^p$, где M называется мантиссой числа, а p – порядком. Такой способ записи чисел называется представлением с плавающей точкой.

Если плавающая точка расположена в мантиссе перед первой значащей цифрой, то при фиксированном количестве разрядов, отведенных под мантиссу, обеспечивается запись максимального количества значащих цифр числа, то есть максимальная точность представления числа в машине. Из этого следует: мантисса должна быть правильной дробью, первая цифра которой отлична от нуля: $M \in [0,1; 1)$. Такое наиболее выгодное для компьютера представление вещественных чисел называется нормализованным. Мантиссу и порядок q -го числа принято записывать в системе с основанием q , а само основание – в десятичной системе.

Примеры нормализованного представления:

Десятичная система

Двоичная система

$$3,15 = 0,00315 \cdot 10^3,$$

$$-101,01 = -0,10101 \cdot 2^{11}$$

(порядок $11_2 = 3_{10}$),

$$-0,000034 = -0,34 \cdot 10^{-4},$$

$$-0,000011 = 0,11 \cdot 2^{-100}$$

(порядок $-100_2 = -4_{10}$).

При хранении числа с плавающей точкой отводятся разряды для мантиссы, порядка, знака числа и знака порядка.

Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа. Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего отличного от нуля числа до наибольшего числа, представимого в машине при заданном формате.

5.4 Кодирование графической информации

Важным этапом кодирования графического изображения является разбиение его на дискретные элементы (дискретизация).

Дискретизация – преобразование непрерывной функции в дискретную.

Графический объект в компьютере может быть представлен как растровое или векторное изображение. От этого зависит и способ кодирования.

Растровое изображение представляет собой совокупность точек (пикселей), полученных в результате дискретизации изображения в соответствии с матричным принципом.

Матричный принцип кодирования графических изображений заключается в том, что изображение разбивается на заданное количество строк и столбцов. Затем каждый элемент полученной сетки кодируется по выбранному правилу.

Растр (итал. *gastro*, от лат. *gastrum* «грабли») – сетка, решетка. В полиграфии – прозрачная пленка с нанесенными на нее тонкими линиями в прямую или косую клетку (рис. 5.13).

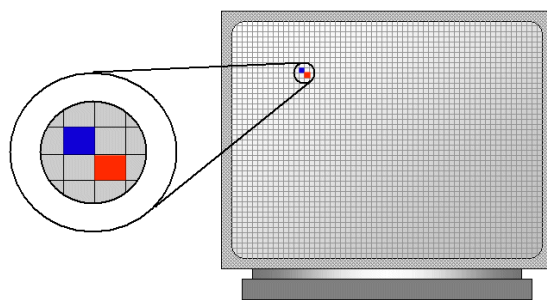


Рисунок 5.13 – Растр

Пиксель – наименьший логический элемент двумерного цифрового изображения в растровой графике. Кодировка растра в ЭВМ происходит за счет создания *растрового файла*. Растровый файл представляет собой прямоугольную таблицу, в каждой ячейке которой установлен *пиксель* (рис. 5.14).

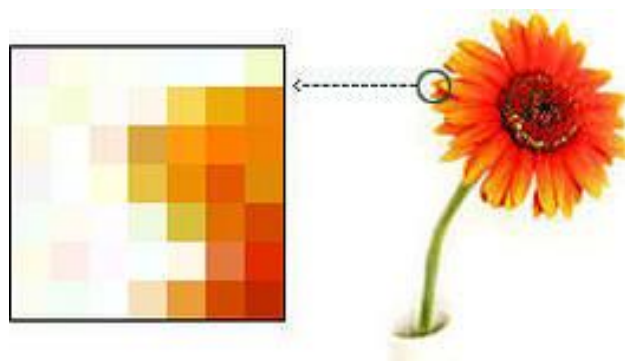


Рисунок 5.14 – Растровое изображение

Размер растрового изображения программируется по ширине и высоте экрана монитора 1024×768, 640×480, Чем больше пикселей, тем выше качество изображения.

Если говорить о черно-белых иллюстрациях и не использовать полутона, то пиксель будет принимать одно из двух состояний:

- светится (белый);
- не светится (черный).

А так как информация о цвете пикселя называется *кодом пикселя*, то для его кодирования достаточно одного бита памяти: 0 – черный, 1 – белый.

Сам компьютер представляет пиксель в виде матрицы и так же он распознает цвет и текстуру (рис. 5.15).

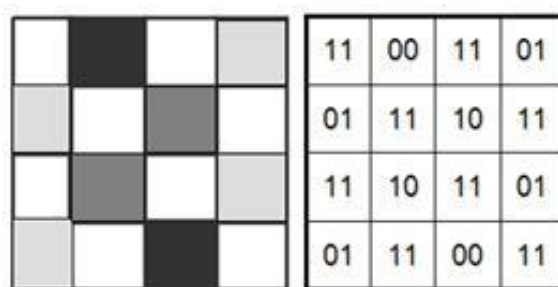


Рисунок 5.15 – Представление пикселя в компьютере

В отличие от однобитного монохромного изображения, в цветном RGB-изображении каждый пиксель кодируется 24-битовым числом.

Сначала определяется размер изображения в виде произведения числа пикселей по горизонтали на число пикселей по вертикали. Затем размер пикселя. И, наконец, *битовая глубина*, характеризующая информационную емкость пикселя в битах или цветовую разрешающую способность изображения.

Векторное изображение представляет собой графический объект, состоящий из элементарных геометрических фигур (чаще всего отрезков и дуг). Положение этих элементарных отрезков определяется координатами точек и величиной радиуса. Для каждой линии указываются двоичные коды типа линии (сплошная, пунктирная, штрихпунктирная), толщины и цвета.

Информация о векторном изображении кодируется как обычная буквенно-цифровая и обрабатывается специальными программами.

5.5 Кодирование звуковой информации

Звуковая информация может быть представлена последовательностью элементарных звуков (фонем) и пауз между ними. Каждый звук кодируется и хранится в памяти. Вывод звуков из компьютера осуществляется синтезатором речи, который считывает из памяти хранящийся код звука.

Гораздо сложнее преобразовать речь человека в код, так как живая речь имеет большое разнообразие оттенков. Каждое произнесенное слово должно сравнивать с предварительно занесенным в память компьютера эталоном, и при их совпадении происходит его распознавание и запись.

Развитие цифровой техники и применение компьютерной обработки и хранения информации привело к широкому использованию импульсных методов модуляции или кодирования, среди которых можно выделить два основных направления.

Метод FM (Frequency Modulation) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый из которых представляет собой правильную синусоиду, а следовательно, может быть описан числовыми параметрами, т.е. кодом.

В природе звуковые сигналы имеют непрерывный спектр, т.е. являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальное устройство – аналогово-цифровые преобразователи (АЦП). Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифро-аналоговые преобразователи (ЦАП). При таких преобразованиях неизбежны потери информации, связанные с методом кодирования, поэтому качество звукозаписи обычно получается не вполне удовлетворительным и соответствует качеству звучания простейших электромузыкальных инструментов с окрасом, характерным для электронной музыки. В то же время данный метод кодирования обеспечивает весьма компактный код, поэтому он нашел применение еще в те годы, когда ресурсы средств вычислительной техники были явно недостаточны.

Метод таблично-волнового (Wave-Table) синтеза лучше соответствует современному уровню развития техники. Суть его заключается в том, что в заранее подготовленных таблицах хра-

няются образцы звуков для множества различных музыкальных инструментах. В технике такие образцы называют *сэмплами*. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звучания. Поскольку в качестве образцов исполняются реальные звуки, то его качество получается очень высоким и приближается к качеству звучания реальных музыкальных инструментов.

Развитие аппаратной базы современных компьютеров параллельно с развитием программного обеспечения позволяет сегодня записывать и воспроизводить на компьютерах музыку и человеческую речь. Существуют два способа звукозаписи.

1. *Цифровая запись*, когда реальные звуковые волны преобразуются в цифровую информацию путем измерения звука тысячи раз в секунду.

2. *MIDI-запись*, которая, вообще говоря, является не реальным звуком, а записью определенных команд – указаний (какие клавиши надо нажимать, например, на синтезаторе). MIDI-запись является электронным эквивалентом записи игры на фортепиано.

Для того чтобы воспользоваться первым указанным способом, в компьютере должна быть звуковая карта (плата). Реальные звуковые волны имеют весьма сложную форму, и для получения их высококачественного цифрового представления требуется высокая частота квантования. Звуковая плата преобразует звук в цифровую информацию путем измерения характеристики звука (уровень сигнала) несколько тысяч раз в секунду. То есть аналоговый (непрерывный) сигнал измеряется в тысячах точек, и получившиеся значения записываются в виде 0 и 1 в память компьютера. При воспроизведении звука специальное устройство на звуковой карте преобразует цифры в аналог звуковой волны. Хранение звука в виде цифровой записи занимает достаточно много места в памяти компьютера. Число разрядов, используемое для создания цифрового звука, определяет качество звучания.

MIDI-запись была разработана в начале 80-х годов (MIDI – Musical Instrument Digital Interface – интерфейс цифровых музыкальных инструментов). MIDI-информация представляет собой команды, а не звуковую волну. Эти команды – инструкции синте-

затому. MIDI-команды гораздо удобнее для хранения музыкальной информации, чем цифровая запись. Однако для записи MIDI-команд необходимо устройство, имитирующее клавишный синтезатор, которое воспринимает MIDI-команды и при их получении может генерировать соответствующие звуки.

5.6 Представление видеоинформации в компьютере

В последнее время использование видеоинформации становится все более популярным: просмотр кинофильмов и видеоклипов, многочисленные видеоигры.

С точки зрения информатики фильм представляет собой сочетание звуковой и графической информации. Для создания на экране эффекта движения используется технология быстрой смены статических картинок. Известно, что если за одну секунду сменяется более 10–12 кадров, то человеческий глаз воспринимает изменения на них как непрерывные.

Однако если использовать традиционные методы кодирования статической информации, электронная версия фильма получится слишком большой. Предложенное усовершенствование состоит в том, чтобы первый кадр запомнить целиком (его обычно называют *ключевым*), а в следующих сохранять лишь отличия от начального кадра (*разностные кадры*).

Принцип формирования разностного кадра представлен на рисунке 5.16, где показано небольшое горизонтальное смещение прямоугольного объекта. Из рисунка 5.16 видно, что на всей площади кадра изменились всего 2 небольшие зоны: первая сзади объекта возвратилась к цвету фона, а на второй – перед ним, фон перекрасился в цвет объекта. Для разноцветных предметов произвольной формы эффект сохранится, хотя изобразить его будет труднее.

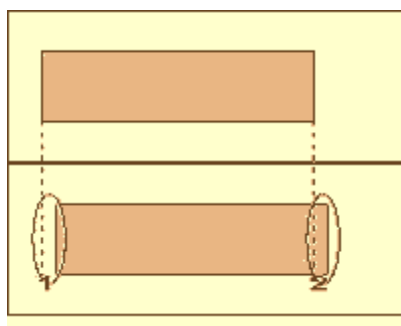


Рисунок 5.16 – Формирование разностного кадра

Конечно, в фильме существует много ситуаций, связанных со сменой действия, когда первый кадр новой сцены настолько отличается от предыдущего, что его проще сделать ключевым, чем разностным. Поэтому ключевых кадров в фильме достаточно много. Регулярное расположение их в потоке позволяет пользователю оперативно начинать просмотр с любого места фильма.

Более того, создание множества ключевых кадров позволяет эффективно восстановить изображение при любых сбоях или при «потере темпа» и пропуске отдельных кадров на медленных компьютерных системах.

Следует отметить, что в современных методах сохранения движущихся видеоизображений используются и другие типы кадров.

Существует множество различных форматов представления видеоданных. В среде Windows, например, применяется формат Video for Windows, базирующийся на универсальных файлах с расширением .avi (Audio Video Interleave – чередование аудио и видео). Суть AVI файлов состоит в хранении структур произвольных мультимедийных данных, каждая из которых имеет простой вид (рис. 5.17). Файл представляет собой единый блок, причем в него, как и в любой другой, могут быть вложены новые блоки. Заметим, что идентификатор блока определяет тип информации, которая хранится в блоке.



Рисунок 5.17 – Структура мультимедийных данных

Внутри такого блока (своеобразного контейнера информации) могут храниться абсолютно произвольные данные, в том числе, например, блоки, сжатые разными методами. Таким образом, все AVI-файлы только внешне выглядят одинаково, а внутри могут различаться очень существенно.

Еще более универсальным является мультимедийный формат **QuickTime**, первоначально возникший на компьютерах Apple. По сравнению с описанным выше он позволяет хранить независимые фрагменты данных, причем даже не имеющие общей временной синхронизации, как этого требует AVI. В резуль-

тате в одном файле может, например, храниться песня, текст с ее словами, нотная запись в MIDI-формате, способная управлять синтезатором, и т.п. Мощной особенностью Quick Time является возможность формировать изображение на новой дорожке путем ссылок на кадры, имеющиеся на других дорожках. Полученная таким способом дорожка оказывается несоизмеримо меньше, чем если бы на нее были скопированы требуемые кадры. Благодаря описанной возможности файл подобного типа легко может содержать не только полную высококачественную версию видеофильма, но и специальным образом «упрощенную» копию для медленных компьютеров, а также рекламный ролик, представляющий собой «выжимку» из полной версии. И все это без особого увеличения объема по сравнению с полной копией.

Все большее распространение в последнее время получают системы сжатия видеоизображений, допускающие некоторые незаметные для глаза искажения изображения с целью повышения степени сжатия. Наиболее известным стандартом подобного класса служит **MPEG** (Motion Picture Expert Group).

Методы, применяемые в MPEG, непросты для понимания и опираются на достаточно сложную математику. Опишем наиболее общие приемы, за счет которых достигается сжатие. Прежде всего, обрабатываемый сигнал из RGB-представления с равноправными компонентами преобразуется в яркость и две «координаты» цветности. Как показывают эксперименты, цветовые компоненты менее важны для восприятия и их можно уменьшить вдвое. Кроме того, производятся специальные математические преобразования (DCT – дискретно-косинусное преобразование), несколько загроубляющие изображение в мелких деталях. Известно, что на субъективном восприятии изображения это практически не сказывается. Наконец, специальными методами ликвидируется сильная избыточность информации, связанная со слабыми отличиями между соседними кадрами. Полученные в результате всех описанных процедур данные дополнительно сжимаются общепринятыми методами, подобно тому, как это делается при архивации файлов.

В последнее время все большее распространение получает технология под названием **DivX** (происходит от сокращения слов **D**igital **V**ideo **E**xpress, обозначающих название видеосистемы). Благодаря DivX удалось достигнуть степени сжатия, позволив-

шей вместить качественную запись полнометражного фильма на один компакт-диск – сжать 4,7 Гб DVD-фильма до 650 Мб. И хотя это достижение, к сожалению, чаще всего используется для пиратского копирования, сам по себе этот факт не умаляет достоинств новой технологии. Наиболее популярные программы проигрывания видеофайлов позволяют использовать замещаемые подсистемы сжатия и восстановления видеоданных – *кодеки* (от англ. *compression/decompression – codec*).

Такой подход позволяет легко адаптировать новые технологии, как только те становятся доступными. Замещаемые кодеки хороши как для пользователей, так и для разработчиков программного обеспечения. Тем не менее большое разнообразие кодеков создает определенные трудности для производителей видеопродукции. Часто в качестве выхода из создавшегося положения необходимые кодеки помещают на компакт-диск с фильмами или даже поставляют видеоматериалы в нескольких вариантах, предоставляя тем самым возможность выбрать подходящий. Все больше распространяется автоматизация распознавания, когда плеер, обнаружив информацию об отсутствующем кодеке, загружает его из Интернета.

Таким образом, рассмотрев принципы кодирования различных видов информации в персональном компьютере, можно сделать важный вывод о том, что все они так или иначе преобразуются в числовую форму и кодируются набором нулей и единиц. Благодаря такой универсальности представления данных, если из памяти наудачу извлечь содержимое какой-нибудь ячейки, то принципиально невозможно определить, какая именно информация там закодирована: текст, число или картинка.

Вопросы и задания для самоконтроля

1. В каком виде представлена информация в памяти компьютера?
2. Нарисуйте разрядную сетку. Покажите, как представлены в памяти компьютера числа 25 и (–25).
3. В каком формате хранятся целые числа в памяти компьютера?
4. Что называют ячейкой памяти компьютера?
5. Что такое машинное слово?

6. Что понимается под байтом?
7. Объясните необходимость использования целочисленных типов данных.
8. Можно ли ограничиться представлением целых чисел как вещественных, но с нулевой дробной частью?
9. Найдите значения верхних границ диапазонов для беззнаковых типов в 16- и 32-разрядном представлении.
10. Что является знаковым разрядом?
11. Какие специальные коды используются для представления числа в компьютере?
12. Чему равно значение знакового разряда для положительных чисел? Для отрицательных?
13. Чему равен дополнительный код для положительного числа? Для отрицательного?
14. Запишите дополнительный код двоичного числа (-1000_2) для восьмиразрядной ячейки.

6 КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

6.1 Основные принципы работы с текстовым процессором MicrosoftOfficeWord

Работа с текстом

К основным приемам работы с документами в текстовом процессоре MicrosoftWord относятся:

- ввод текста;
- редактирование текста;
- рецензирование текста;
- форматирование текста;
- сохранение и печать документа.

Учитывая, что базовые знания по работе с текстами в этом процессоре даются в школьном курсе информатики, здесь мы не будем рассматривать все приемы, а остановимся только на основных моментах, которые нужны и важны в образовательной деятельности.

Ввод, редактирование и рецензирование текста

Символы. Ввод текста с использованием символов, имеющих на клавиатуре, как правило, не вызывает трудностей. В том случае, если необходимо вставить буквы или символы, которых нет на клавиатуре ПК, используется команда *Вставка/Символ*. Откроется окно, в котором представлены используемые недавно символы. Если нужного символа не оказалось, следует кликнуть на кнопке *Другие символы*, в появившемся окне открыть нужную вкладку, выбрать категорию шрифта и нужный набор символов, выделить символ и нажать на кнопку *Вставить* (рис. 6.1).

Тире и дефис. Важно помнить, что дефис и тире – это разные знаки с различным значением. *Дефис* – орфографический знак, который ставится внутри слова (например, все-таки, во-первых). Он короткий, с обеих сторон от него нет пробелов, на клавиатуре ПК дефис набирается как знак «минус» (-).

Тире – это пунктуационный знак. Он используется между словами, пишется как длинный дефис, отделяется пробелами с обеих сторон. На клавиатуре ПК он набирается одновременным нажатием клавиш *Ctrl* и *минус* (-) на цифровом блоке клавиатуры. В результате получаем знак тире (–).

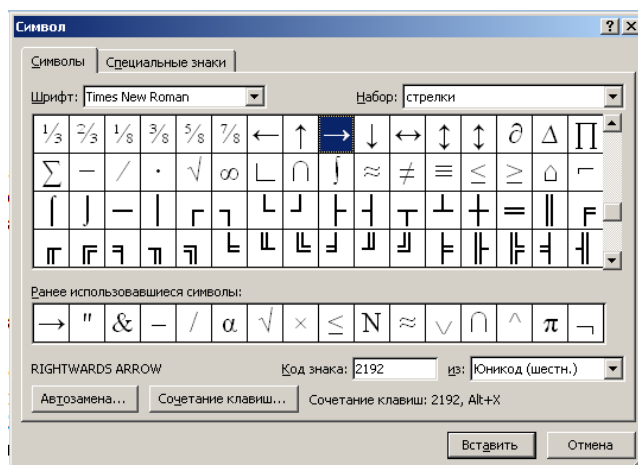



Рисунок 6.1 – Окно Другие символы

Неразрывный пробел. При выравнивании текста по ширине программа распределяет все слова строки так, чтобы левый и правый края текста были ровными, по ширине страницы. При этом часто бывает, что между словами, которые должны быть напечатаны рядом, появляется большой пробел (например, при наборе инициалов и фамилии – И.И. Иванов – между ними не должно быть значительного пробела). Такая же ситуация возможна, когда программа переносит слова на другую строку, разрывая словосочетание (например, при введении номера рисунка мы пишем рис. 3). В таких случаях ставится *неразрывный пробел* – его можно получить сочетанием трех клавиш: *Ctrl*, *Shift* и *Пробел*.

Непечатаемые символы. *Непечатаемые символы* – это символы, которые не видны при наборе и распечатке текста. Однако они играют большую роль при редактировании и форматировании текста, особенно если текст скопирован из сети Интернет. Включение режима просмотра непечатаемых символов позволяет увидеть лишние пробелы, неразрывные пробелы, принудительный перенос текста, разрыв страницы или абзаца, скрытый текст (и это возможно в MicrosoftWord) и т.п. Для включения этого режима следует нажать на клавишу  в разделе *Абзац* вкладки *Главная*.

Настройка отображения скрытых символов осуществляется в строке *Экран меню* *Параметры Word*.

Сочетания клавиш. Использование сочетания клавиш предполагает одновременное их нажатие для выполнения определенной запрограммированной команды. Часто вместо этого названия используют синонимы «горячая клавиша», «клавиша

быстрого вызова». Обычно сочетания клавиш дублируют команды меню, чем значительно ускоряют работу. В таблице 6.1 приведены наиболее часто употребляемые сочетания клавиш.

Таблица 6.1 – Горячие сочетания клавиш

Сочетание клавиш	Команда
<i>Перемещение курсора</i>	
Home	Перемещение курсора в начало текущей строки
End	Перемещение курсора в конец текущей строки
Ctrl+Home	Перемещение курсора в начало поля ввода текста
Ctrl+End	Перемещение курсора в конец поля ввода текста
PageUp	Перемещение курсора на начало экрана
PageDown	Перемещение курсора в конец экрана
<i>Выделение текста</i>	
Shift+Home	Выделение текста между текущим положением курсора и началом строки
Shift+End	Выделение текста между текущим положением курсора и концом строки
Shift+Ctrl+Home	Выделение текста между текущим положением курсора и началом текста
Shift+Ctrl+End	Выделение текста между текущим положением курсора и концом текста
Shift+PageDown	Выделение текста между текущим положением курсора и началом экрана
Shift+PageUp	Выделение текста между текущим положением курсора и концом экрана
Ctrl+A	Выделение всего текста
<i>Редактирование</i>	
Ctrl+C, Ctrl+Insert	Копировать выделенный текст
Ctrl+X, Shift+Delete	Вырезать выделенный текст
Ctrl+V, Shift+Insert	Вставить выделенный текст
Ctrl+Z	Отменить
Ctrl+Y	Повторить

Редактирование текста. Под *редактированием текста* (лат. *redactus* – приведенный в порядок) обычно понимают проверку литературного стиля, логики изложения, изменение его структуры, а также проверку орфографии и грамматики. Как правило, проверкой стиля и логики изложения занимаются специалисты – редакторы в издательстве, а вот изменить структуру и

провести проверку правописания можно самим в текстовом процессоре.

В программе Microsoft Word заложены два режима проверки – *командный* и *автоматический*.

Для включения (или отключения) автоматического режима проверки необходимо открыть вкладку *Параметры* в меню программы Microsoft Word (кнопка *Office* в правом верхнем углу окна программы) (рис. 6.2).

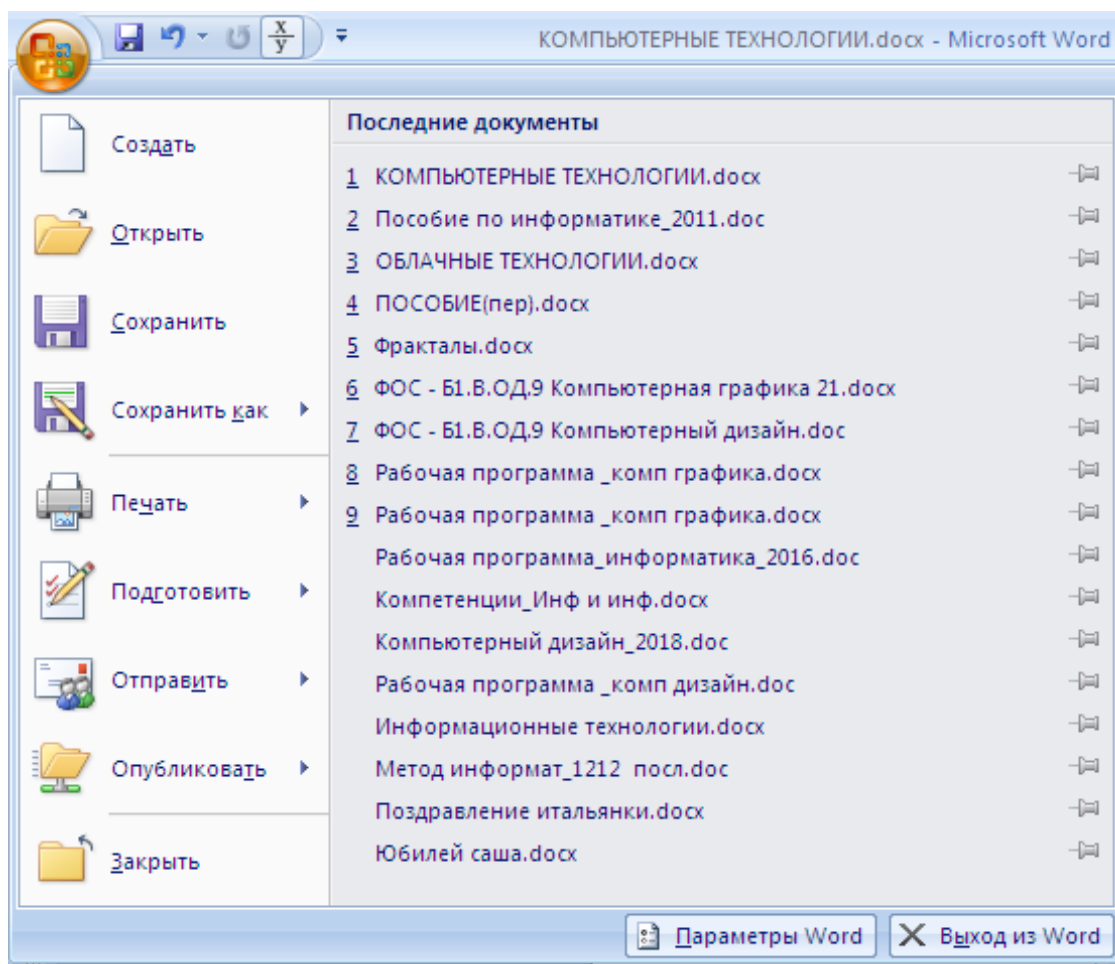


Рисунок 6.2 – Кнопка *Office*

В разделе *Правописание* включите (или отключите) необходимые функции правописания в Word (рис. 6.3).

Сохраните заданные параметры, кликнув мышью на кнопке *OK*. Обратите внимание, что в зависимости от версии программы Microsoft Word названия функций могут отличаться.

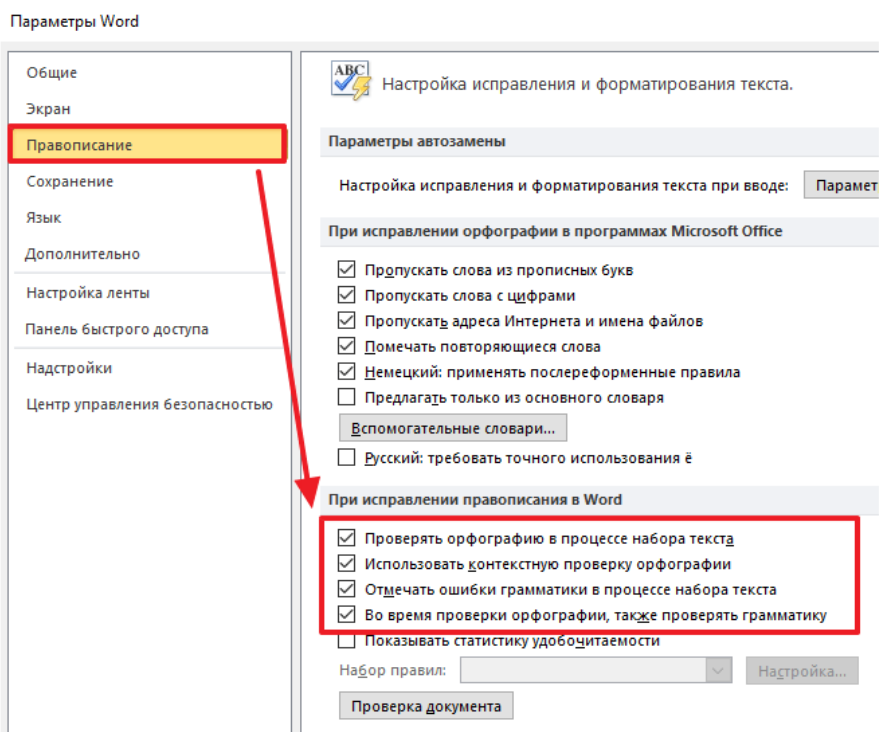


Рисунок 6.3 – Вкладка Правописание

Рецензирование. Для проверки правописания в командном режиме достаточно выбрать команду *Рецензирование/Правописание* (рис. 6.4).

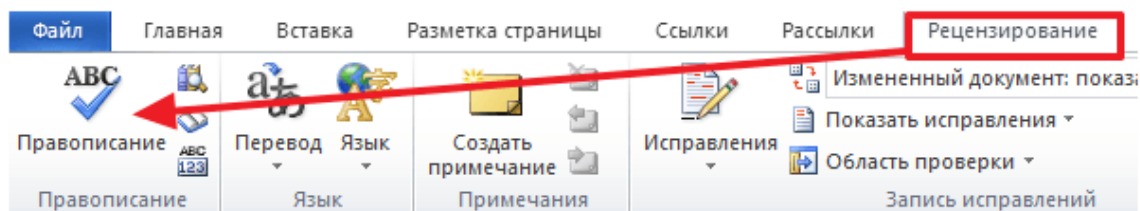


Рисунок 6.4 – Меню Рецензирование

Слова, написанные с ошибками, будут выделены красным цветом, при этом программа предложит несколько вариантов замены. Выбрав нужный вариант замены, следует нажать на кнопку *Заменить*. Если, с вашей точки зрения, слово написано правильно, нажмите на кнопку *Пропустить*.

Если среди предложенных вариантов замены нет нужного слова, его можно добавить в словарь с помощью кнопки *Добавить*.

В том случае, если обнаружена ошибка в слове, которое часто повторяется в набранном тексте, можно включить функцию автозамены в разделе *Правописание* (рис. 6.5).

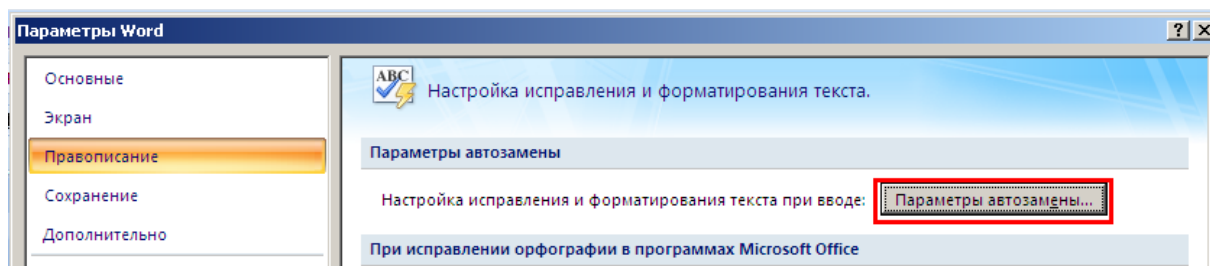


Рисунок 6.5 – Параметры автозамены

Во вкладке *Автозамена* в поле *Заменить* нужно ввести слово с ошибкой, а в поле *На* – то слово, на которое следует заменить (рис. 6.6).

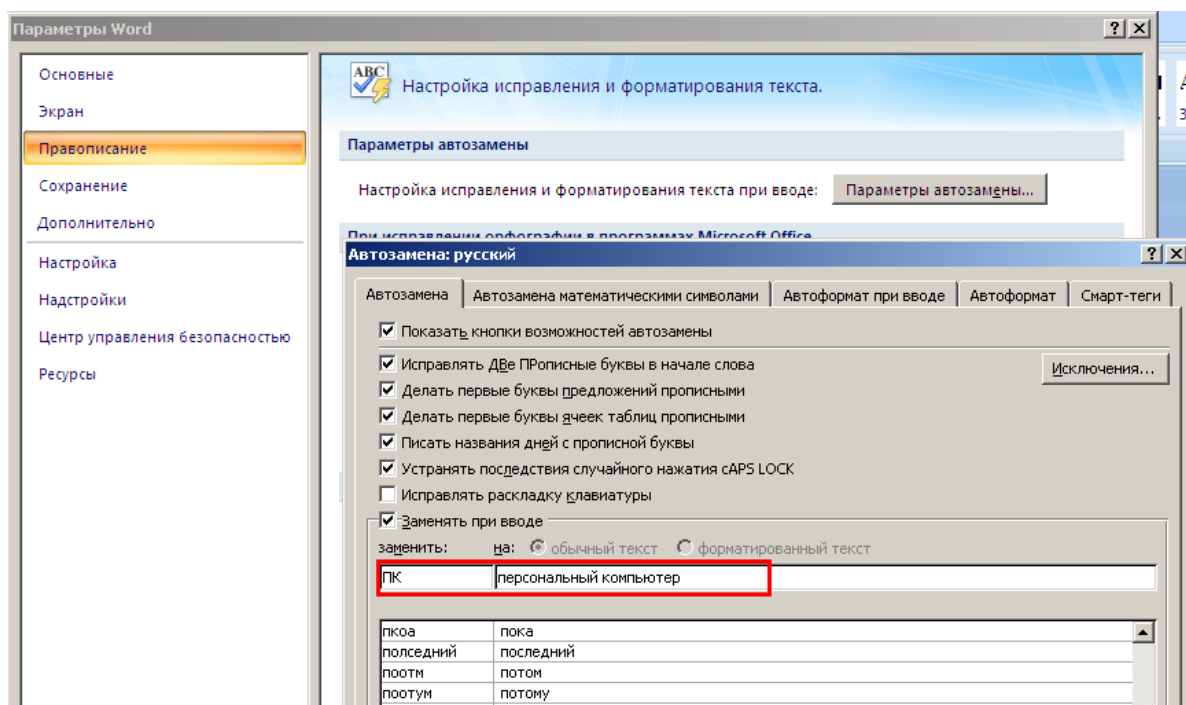


Рисунок 6.6 – Пример автозамены

При изменении структуры документа часто появляется необходимость одновременного просмотра его разных частей. В этом случае удобно отобразить один и тот же документ в двух разных окнах, воспользовавшись командой *Новое окно* во вкладке *Вид*.

Форматирование текста. *Форматирование текста* представляет собой процедуру приведения его к требуемому виду.

К основным параметрам форматирования относятся:

- настройка параметров страницы (поля, ориентация, переносы и т.п.);
- выбор параметров шрифта (гарнитура, размер, начертание, цвет, выравнивание);
- настройка параметров абзаца (отступы, интервалы);
- задание стилей;
- вставка колонтитулов и нумерация страниц;
- создание маркированных и нумерованных списков (в том числе многоуровневых);
- вставка оглавления.

Настройка параметров страницы. Настройка параметров страницы осуществляется во вкладке *Разметка страницы* (рис. 6.7).

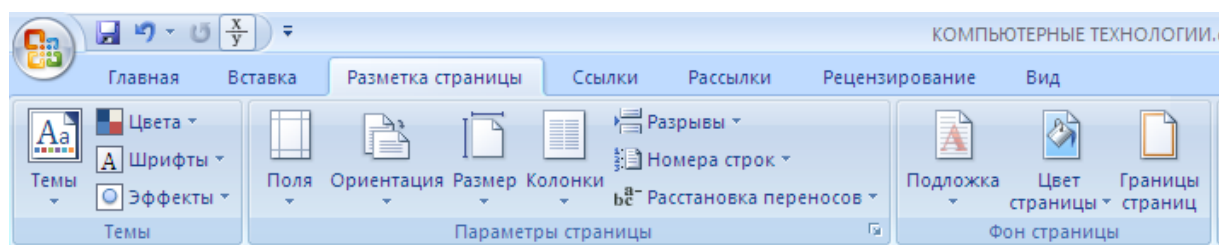


Рисунок 6.7 – Окно настройки параметров страницы

Для выбора параметров шрифта необходимо открыть вкладку *Шрифт* в меню *Главная* (рис. 6.8).

Размер шрифта измеряется в полиграфических пунктах. Один пункт равен 1/72 дюйма (введен компанией Adobe), стандартный типографский пункт равен 0,376 мм. Для разных категорий шрифта при одинаковом значении размера шрифта высота букв может отличаться.

Во вкладке *Интервал* окна настройки параметров шрифта можно изменить расстояние между символами в словах. Для этого следует выбрать один из возможных режимов: *Обычный*, *Разреженный* и *Уплотненный*.

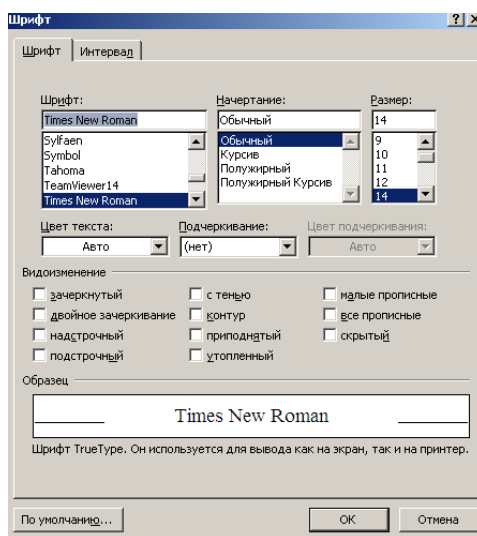


Рисунок 6.8 – Окно настройки параметров шрифта

Настройка параметров абзаца. Параметры настройки для абзаца размещены во вкладке *Абзац* в меню *Главная* (рис. 6.9).

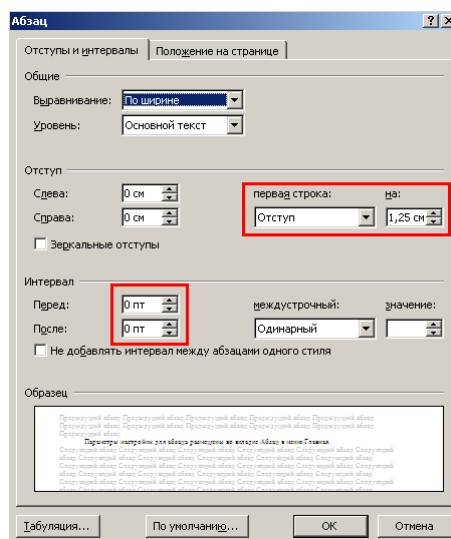


Рисунок 6.9 – Настройка параметров абзаца

Обратите внимание (выделено красным), где следует задавать отступ для первой строки абзаца и дополнительные интервалы к междустрочному, если это необходимо (например, отступы перед рисунком и после него).

Задание стилей. Под стилем в программе Microsoft Word понимают набор определенных форматов, которые применяются к части текста для формирования его структуры. Чаще всего весь текст делится на разделы разного уровня – например, глава, параграф и т.п. При этом одна глава, как правило, содержит несколько

параграфов, а параграф – несколько абзацев. Поэтому названиям разделов задают стили разного уровня: названию главы – стиль *Заголовок 1* (первого уровня), названию параграфа – *Заголовок 2* (второго уровня) и т.д. Эти стили отражают структуру документа и отображаются в оглавлении.

Задание и просмотр стилей осуществляются в разделе *Стили* вкладки *Главная*. Стиль *Обычный* не имеет уровня и предназначен для набора обычного текста. Для изменения параметров стиля необходимо нажать на кнопку *Параметры* в окне стилей (рис. 6.10).



Рисунок 6.10 – Окно стилей

Вставка колонтитулов и нумерация страниц. *Колонтитулом* (фр. *colonne* – столбец, лат. *titulus* – надпись, заголовок) называют строку, расположенную на краю полосы набора (в верхнем или нижнем поле страницы), в которой размещают текст или изображение, не относящиеся к основному тексту документа. Колонтитулы носят информационный или оформительский характер, в них указывают такую информацию, как, например, номера страниц, название книги или главы, имя автора, логотип организации и т.п.

Для вставки колонтитула откройте вкладку *Вставка*, в разделе *Колонтитулы* выберите необходимый вариант размещения колонтитула (верхний или нижний), в открывшемся окне выделите макет колонтитула и введите нужную информацию. Колонтитул будет отделен от основного текста пунктирной линией. Выход из режима колонтитула осуществляется с помощью двойного клика мышью в любом месте документа. В этом же окне находится команда *Удалить колонтитул*. При необходимости можно создать свой колонтитул, для этого следует выбрать место размещения колонтитула и выбрать команду *Изменить колонтитул* (рис. 6.11).

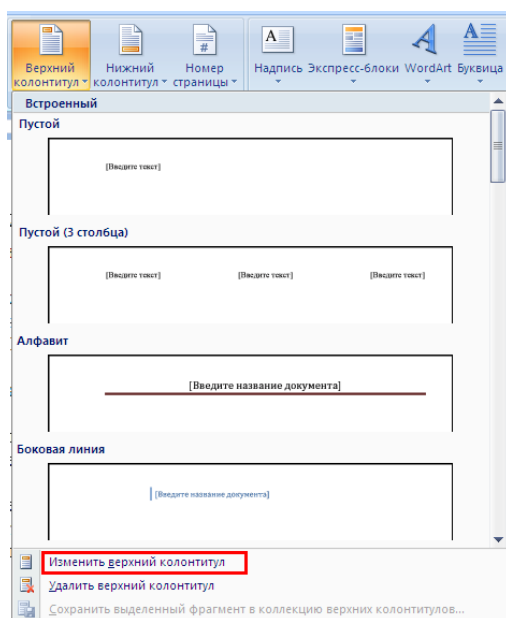


Рисунок 6.11 – Ниспадающее окно раздела Колонтитулы

Колонтитулы на разных страницах (титальной, четной, нечетной и т.п.) могут различаться. Для изменения настроек колонтитула выберите команду *Изменить колонтитул* и в открывшейся дополнительной вкладке *Работа с колонтитулами* установите флажок рядом с нужным параметром (рис. 6.12).

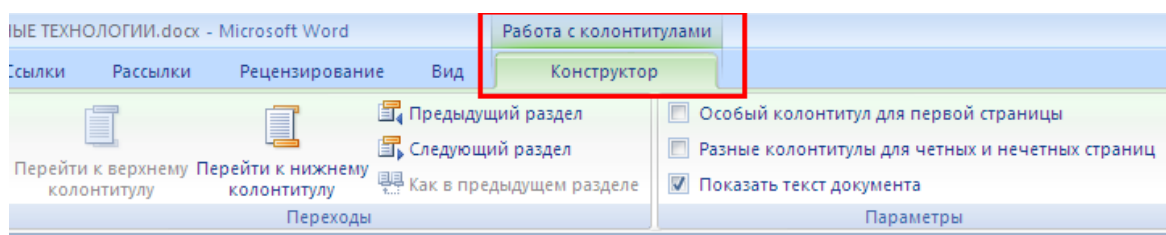


Рисунок 6.12 – Дополнительная вкладка

Работа с колонтитулами

Для вставки номеров страниц используется команда *Номер страницы*, которая находится во вкладке *Вставка* в разделе *Колонтитулы*. В программе заложена возможность изменять формат номера страницы, убирать номер страницы с титульного листа и т.п. (рис. 6.13).

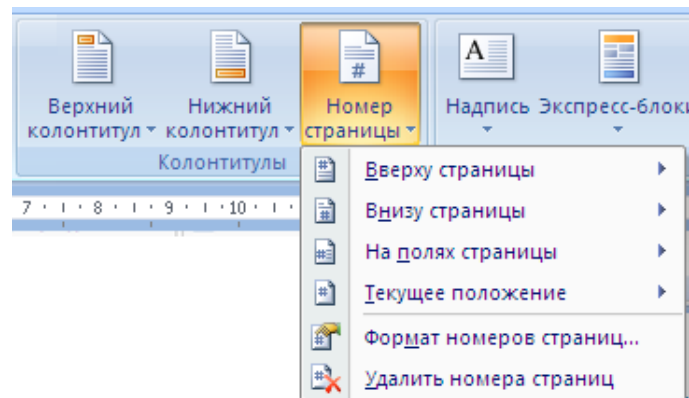


Рисунок 6.13 – Окно изменения параметров номера страницы

При необходимости можно изменить вид нумерации страниц, т.е. изменить категорию шрифта, его размер, начертание, цвет и т.д. Для этого необходимо выделить номер страницы и во вкладке *Главная* задать необходимые параметры. Это можно сделать и на мини-вкладке, которая появится при выделении номера страницы.

Создание списков. По сути, *список* – это перечисление чего-нибудь в тексте. В программе MicrosoftWord доступны три вида списков:

- нумерованный;
- маркированный;
- многоуровневый.

Нумерованный список представляет собой перечисление, в котором каждый пункт начинается с цифры. Его используют обычно тогда, когда необходимо выделить количество перечисляемого.

В *маркированном списке* в начале каждого пункта стоит символ – *маркер*.

Многоуровневый список – это структура, состоящая из одного или нескольких подчиненных списков. Примером многоуров-

нового списка может быть оглавление книги, включающее название главы, нескольких параграфов, абзацев.

Существуют два способа создания списков.

Первый заключается в том, что сначала нужно набрать текст, состоящий из нескольких пунктов перечислений (причем начинать каждое новое значение списка нужно с новой строки), а затем – выбрать *Маркеры* или *Нумерация* в разделе *Абзац* вкладки *Главная* (рис. 6.14).

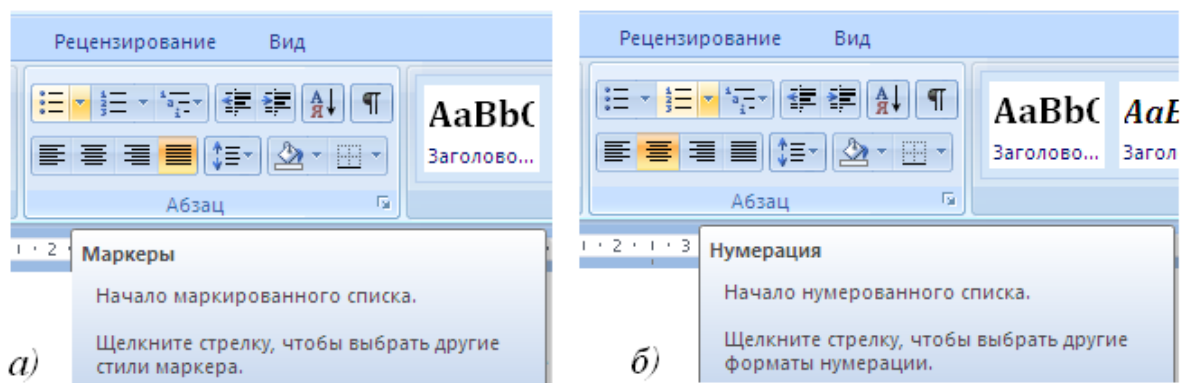


Рисунок 6.14 – Создание списков:
а – маркированный; б – нумерованный

При втором способе для создания маркированного списка нужно перейти на новую строку и нажать клавишу (*), а затем пробел – появится первый маркер списка. Введенное слово списка будет располагаться на некотором расстоянии от маркера, которое можно менять с помощью линейки (она включается в разделе *Показать* или *скрыть* вкладки *Вид*). Для этого потяните указателем мыши при нажатой левой клавише за черный уголок, находящийся в начале линейки. Следующую запись списка нужно начинать с новой строки.

Для создания нумерованного списка следует ввести цифру 1. (с точкой) и нажать пробел. Далее поступать так же, как и при формировании маркированного списка.

При необходимости можно изменить вид маркера или формат цифры, если кликнуть мышью на стрелке рядом с кнопками *Маркеры* или *Нумерация* (рис. 6.15).

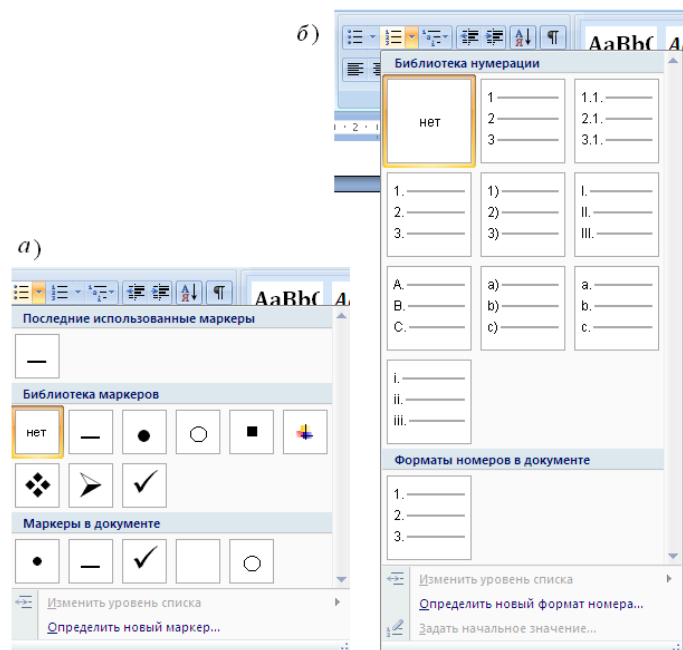


Рисунок 6.15 – Ниспадающее меню кнопок:
а – маркеры; б – нумерация

Для создания многоуровневого списка кликните на кнопке *Многоуровневый список* в разделе *Абзац* в открывшемся окне и выберите формат списка (рис. 6.16).

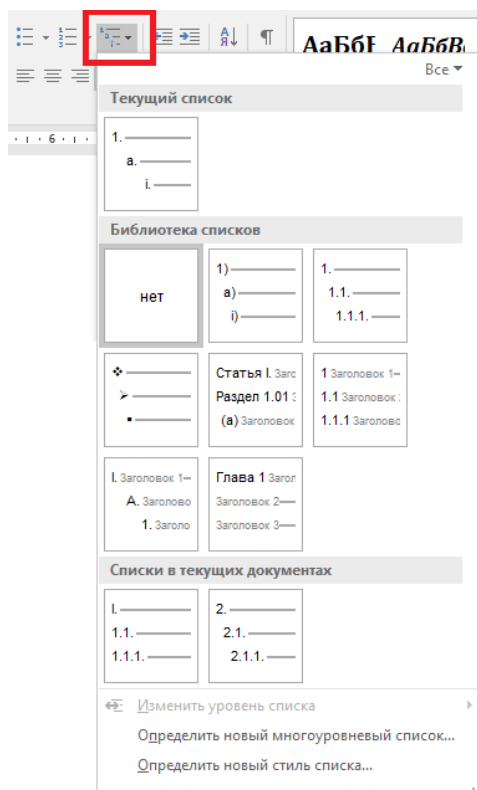


Рисунок 6.16 – Создание многоуровневого списка

Кнопки *Уменьшить отступ* и *Увеличить отступ* в разделе *Абзац* позволяют понижать и повышать уровни в списке (рис. 6.17).

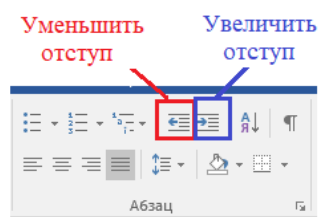


Рисунок 6.17 – Кнопки изменения уровня в списке

Работа с таблицами. Таблицы нередко становятся необходимым элементом в тексте документа. Программа *MicrosoftWord* предоставляет широкие возможности для работы с таблицами в документе. К основным приемам работы с таблицами относятся:

- создание;
- редактирование;
- форматирование;
- работа с диаграммами.

Создание таблиц. Существует два способа создания таблицы в текстовом документе: вставить таблицу с указанием количества строк и столбцов, нарисовать таблицу. Выбрать тот или иной способ построения можно в разделе *Таблицы* во вкладке *Вставка* (рис. 6.18).

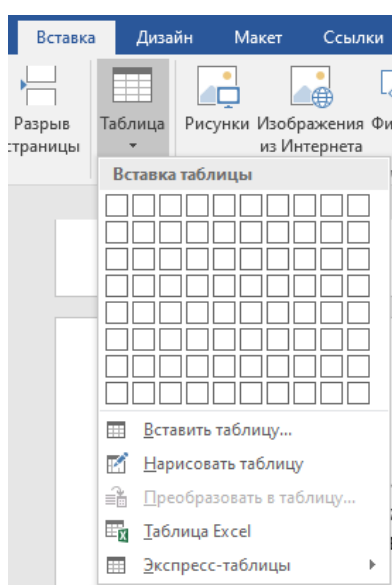


Рисунок 6.18 – Раздел *Таблица*

Для вставки таблицы следует выделить мышью необходимое количество ячеек либо выбрать команду *Вставить таблицу*, в открывшемся окне задать количество строк и столбцов и нажать на кнопку *ОК*. В этом же окне можно задать ширину столбцов.

При выборе команды *Нарисовать таблицу* указатель мыши принимает вид карандаша и таблица рисуется вручную.

Редактирование таблиц. К основным действиям при редактировании таблиц относятся:

- изменение размеров ячеек;
- изменение направления текста в ячейках;
- добавление строк и столбцов;
- удаление выделенных ячеек, строк и столбцов;
- объединение выделенных ячеек;
- разбиение выделенных ячеек.

Все команды для выполнения этих действий сгруппированы в дополнительной вкладке *Работа с таблицей* в разделе *Макет* (рис. 6.19).

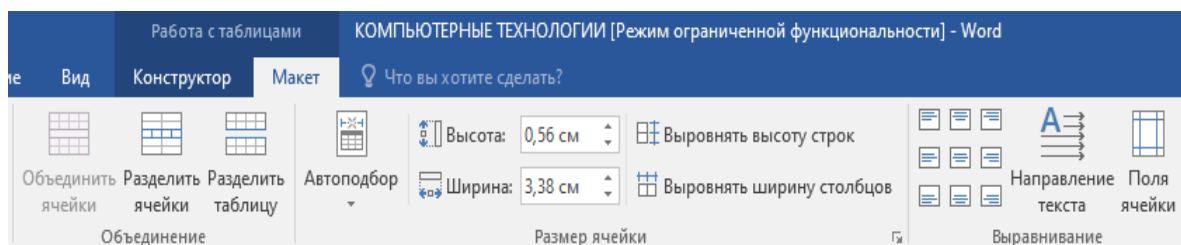


Рисунок 6.19 – Раздел *Макет*

Форматирование таблиц. Под форматированием таблицы, как правило, понимают изменение внешнего вида таблицы, а именно:

- изменение стиля;
- добавление, удаление и редактирование границ таблицы;
- добавление фона как отдельным ячейкам, так и всей таблице.

Команды для форматирования таблицы объединены в разделе *Конструктор* дополнительной вкладки *Работа с таблицами* (рис. 6.20).

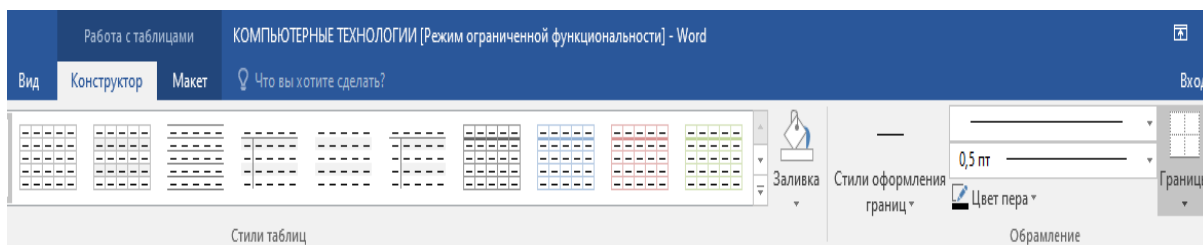


Рисунок 6.20 – Раздел Конструктор

Как видно из рисунка, придать красивый внешний вид таблице можно двумя способами: выбрать готовый стиль и применить к таблице либо вручную выбрать толщину, тип и цвет границ таблицы, а также цвет заливки (для отдельных ячеек можно задать разные цвета).

Вычисления в таблице. Как правило, таблицы используются для представления числовой информации. Однако при необходимости и в текстовом процессоре можно производить несложные вычисления, не обращаясь к табличному процессору.

Для вставки формулы в таблицу необходимо:

- 1) поставить указатель мыши в ячейку таблицы, в которую следует вставить формулу;
- 2) в разделе *Макет* дополнительной вкладки *Работа с таблицами* кликнуть на кнопке *Формула* (рис. 6.21);
- 3) в появившемся окне ввести нужную формулу.

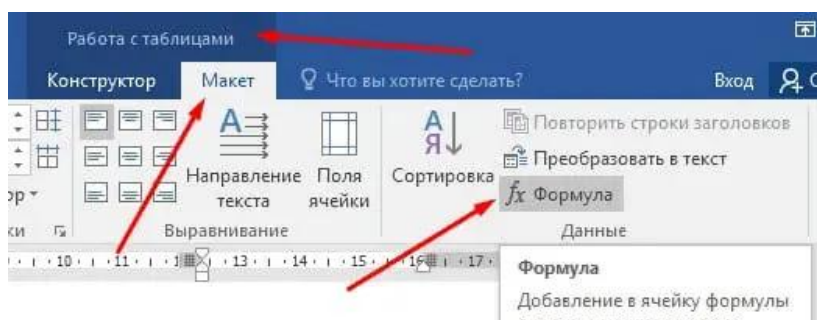


Рисунок 6.21 – Вставка формулы

По умолчанию окно открывается с уже введенной формулой для вычисления суммы: `=SUM(ABOVE)`. Знак «`=`» в записи показывает, что это формула, его всегда нужно набирать. В скобках записывается параметр, показывающий расположение чисел относительно курсора, которые нужно просуммировать:

- ABOVE – выше;
- BELOW – ниже;

LEFT – слева;

RIGHT – справа.

Имеется 18 допустимых операций. Наиболее часто используются такие функции, как:

AVERAGE () – среднее арифметическое;

COUNT () – количество элементов;

MAX () – нахождение наибольшего числа;

MIN () – нахождение наименьшего числа;

PRODUCT () – вычисление произведения чисел.

В скобках, как и для суммирования, указывается расположение чисел относительно курсора, для которых применяется функция.

Если требуется применить формулу не ко всему ряду данных, можно в скобках указать адреса ячеек (как в Excel) только нужных данных (рис. 6.22).

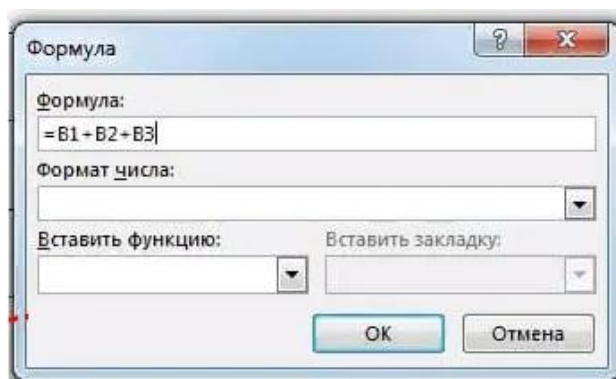


Рисунок 6.22 – Формула с адресами ячеек

Адрес ячейки состоит из имени столбца и номера строки. Столбцы именуются латинскими буквами: *A, B, C, D, E, F* и т.д., строки нумеруются по порядку сверху вниз: *1, 2, 3* и т.д.

Работа с диаграммами. Диаграмма представляет собой удобное средство визуального представления информации и позволяет показать отношения между различными рядами данных. Текстовый редактор Word может создавать диаграммы двумя способами: внедрить ее в документ или вставить диаграмму из табличного процессора Excel. Выбор способа вставки диаграммы зависит от того, где располагаются данные для диаграммы. Если данные находятся на листе табличного процессора, то при включении диаграммы в текстовый документ она теряет связь с данными и обновляться при обновлении данных не будет.

Рассмотрим *первый способ*. Для вставки диаграммы необходимо выбрать команду *Вставка/Диаграмма*. Откроется окно, представляющее все возможные типы диаграмм (рис. 6.23).

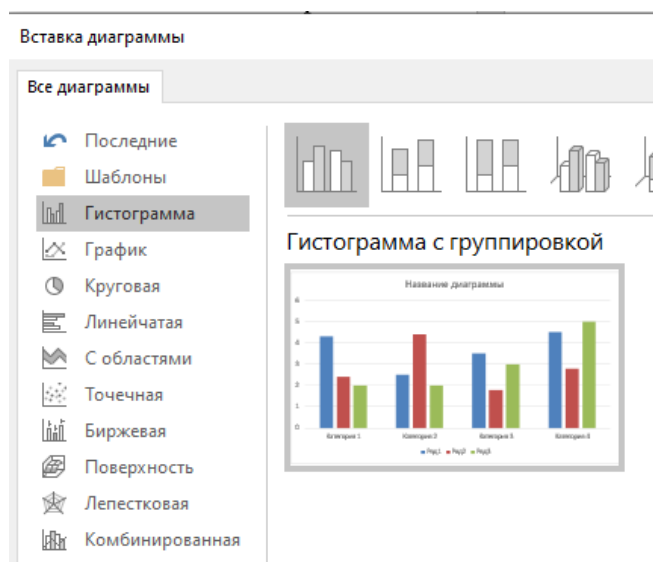


Рисунок 6.23 – Типы диаграмм

Выберите нужный вид и нажмите на кнопку *ОК*. На странице документа появится диаграмма и чуть ниже в отдельном окне заполненная таблица, похожая на таблицу Excel. Необходимо заменить данные таблицы на свои, при этом можно изменить подписи осей (в таблице это первый столбец) и имя легенды (первая строка) (рис. 6.24).

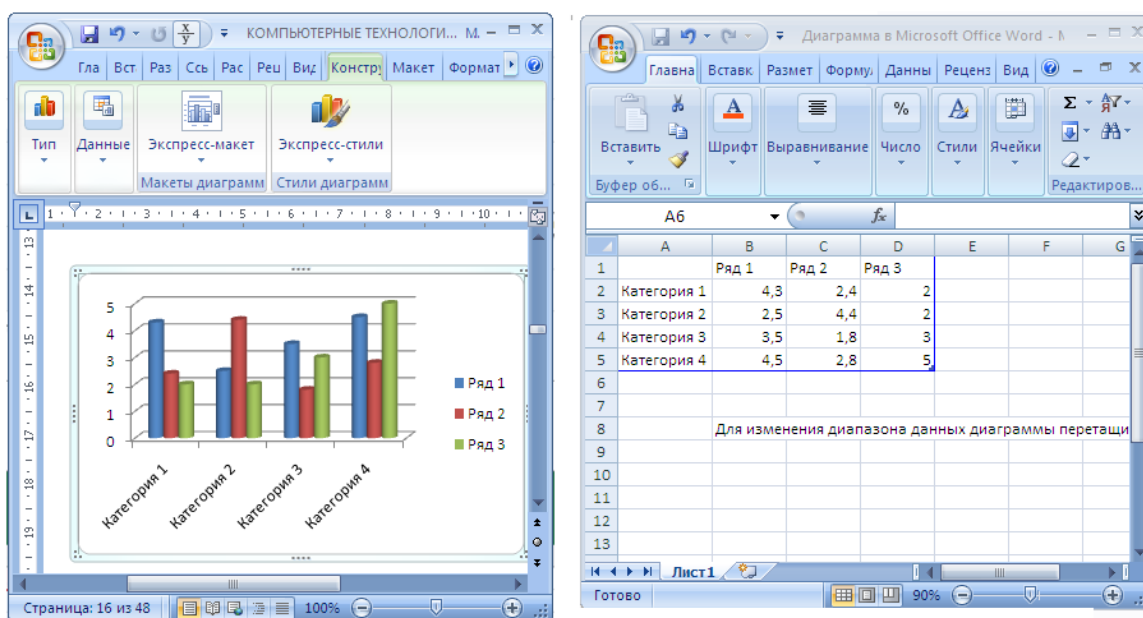


Рисунок 6.24 – Построение диаграммы

После внесения правок в таблицу их следует подтвердить нажатием кнопки *Изменить данные* в разделе *Конструктор* дополнительной вкладки *Работа с диаграммами* (рис 6.25).

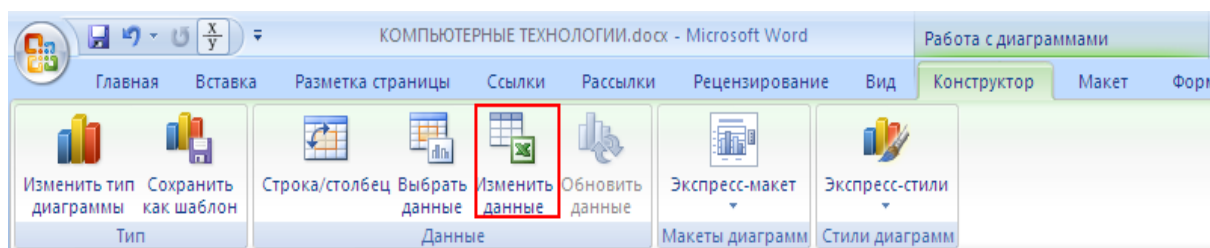


Рисунок 6.25 – Раздел Конструктор

Второй способ создания диаграмм связан с работой в табличном процессоре Excel. Суть его заключается в том, чтобы набрать данные в таблицу и построить диаграмму по этим данным. Далее диаграмма копируется (или вырезается) и вставляется в текстовый документ. Более подробно процесс построения диаграммы в Excel мы рассмотрим в следующем параграфе.

Работа с формулами в тексте. Для набора уравнений в тексте служит *Редактор формул*, который является стандартным элементом текстового редактора Word. Открывается он с помощью команды *Вставка/Формула* либо *Вставка/Символы/Уравнение* (для Word 2013 и Word 2016) (рис. 6.26).

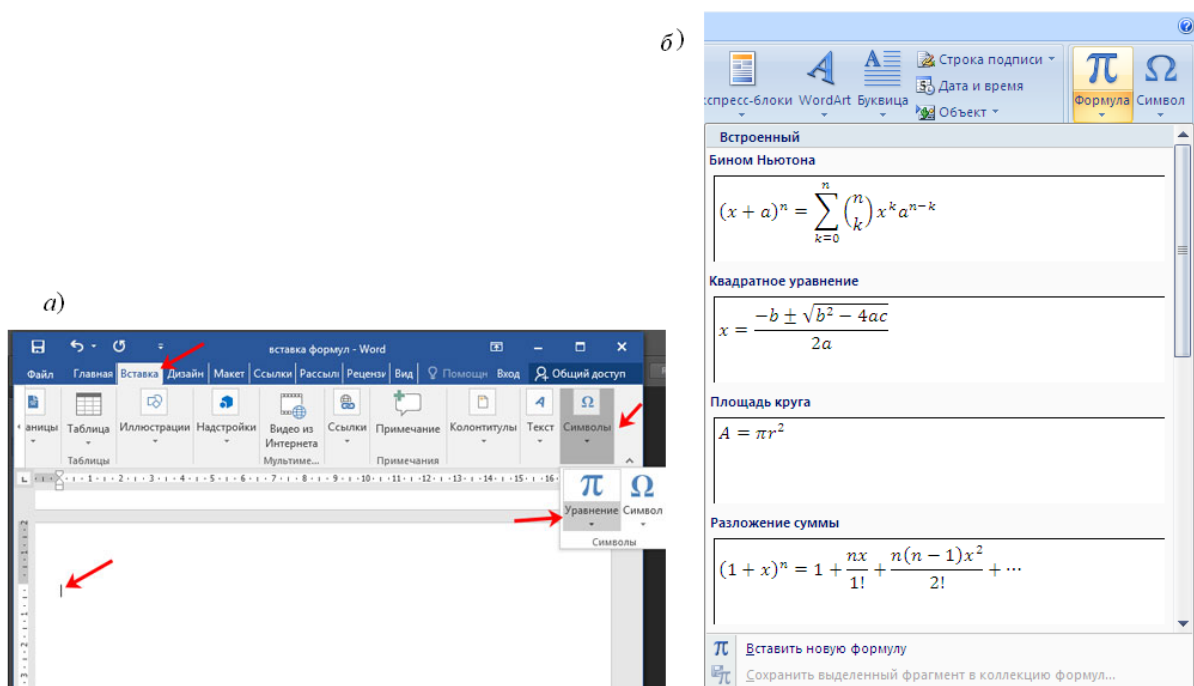


Рисунок 6.26 – Редактор формул:
 а – для Word 2013 и Word 2016; б – для Word 2007 и Word 2010

В появившемся окне можно выбрать готовую формулу из предложенного списка либо ввести свою, кликнув на кнопку *Вставить новую формулу*.

При вставке новой формулы в месте, где стоит курсор, появится блок уравнения, а в меню откроется дополнительная вкладка *Работа с формулами* с разделом *Конструктор*. В *Конструкторе* представлены два поля: одно с символами, другое со структурами. Структуры используются в тех случаях, когда нужно набрать дробь, степенное выражение, корень, интеграл и т.п. Например, при наборе корня выбираем одну из предложенных структур и вводим необходимые символы в пустые квадраты (рис. 6.27).

Примечание. Важно отметить, что в этом случае положение курсора играет большую роль. Пока вы набираете подкоренное выражение, курсор не должен выходить за пределы выделенного цветом блока, он должен мигать в блоке. При продолжении уравнения за корнем необходимо курсор поставить за блок, воспользовавшись стрелкой вправо на клавиатуре (блок не будет выделен). Аналогична ситуация для уравнений, содержащих дроби, систем уравнений и т.п.

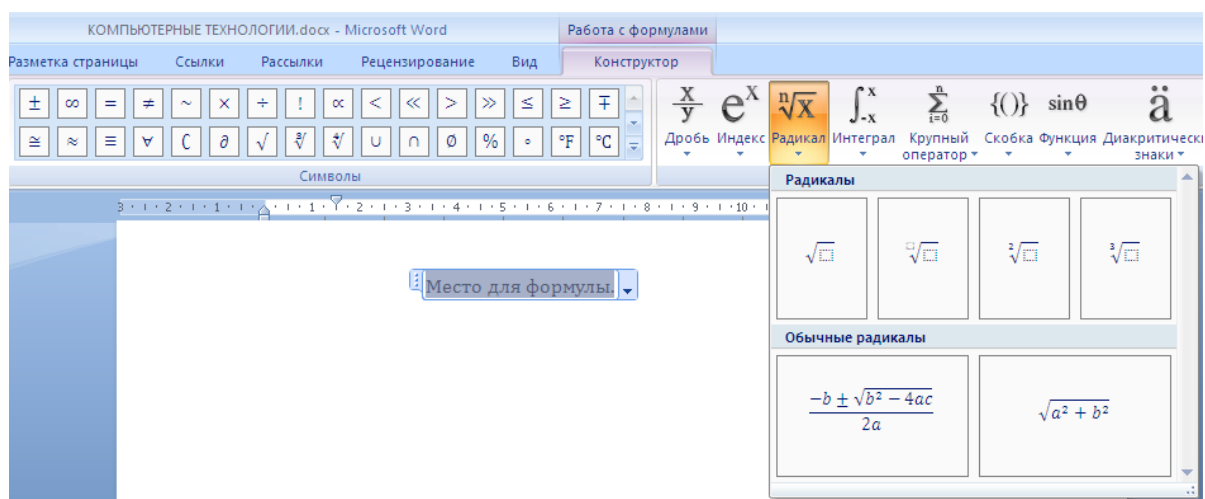


Рисунок 6.27 – Символы и структура Редактора формул

Для редактирования формулы достаточно дважды кликнуть мышью в поле формулы.

Работа с графическими объектами и рисунками. В документах Microsoft Word можно использовать три типа графических объектов: фигуры, объекты *Надпись* и объекты *WordArt*. *Фигуры* – это векторные графические объекты (линии, стрелки, схемы, гео-

метрические фигуры). Простейшие средства для их создания и редактирования есть в самом текстовом процессоре.

Объекты *Надпись* позволяют добавлять текст с привязкой к рисункам и фигурам. Объекты *WordArt* – это художественный текст, который можно вставлять в документ.

Рисунки относятся к растровым объектам. Они вставляются в документ из других файлов. К рисункам в Word относятся сканированные изображения, фотографии, картинки и т.п.

Работа с фигурами. Все средства для создания и редактирования рисунков размещены в разделе *Иллюстрации* вкладки *Вставка*. При нажатии кнопки *Фигуры* откроется окно, в котором представлены различные группы фигур.

Для рисования в тексте следует выделить фигуру мышью и при нажатой левой клавише перемещать мышь по столу.

Редактирование фигур включает в себя такие действия, как:

- изменение размера – для этого необходимо выделить фигуру и перетянуть маркер в нужном направлении;

- изменение толщины, типа и цвета контура, цвета заливки фигуры;

- придание эффекта объема фигуре – можно изменять глубину, направление, цвет объемной фигуры, настраивать освещение;

- построение тени – предлагаются различные стили теней и возможность построения перспективы;

- поворот и отражение фигуры;

- размещение фигуры относительно текста – предлагается несколько вариантов обтекания фигуры текстом: в тексте, за текстом, перед текстом вокруг рамки, по контуру и т.д.

Все необходимые команды для проведения этих операций размещены в разделе *Формат* дополнительной вкладки *Средства рисования*, которая открывается при выделении фигуры.

При создании рисунка, состоящего из отдельных фигур, необходимо сгруппировать фигуры, чтобы при перемещении рисунка сохранил свою целостность. Для группировки фигур следует выделить поочередно все фигуры при нажатой клавише *Ctrl* на клавиатуре и в контекстном меню (открывается с помощью щелчка правой клавишей мыши в выделенной области) выбрать команду *Группировка/Группировать*.

Объекты *Надпись*. Нарисованные объекты могут содержать текстовые элементы, такие как заголовки, обозначения на схемах

и чертежах в виде букв или цифр. Если воспользоваться основными средствами текстового процессора для создания надписей, то надписи не будут связаны с рисунками и при редактировании документа их местоположение может измениться. Чтобы присоединить надпись к рисунку, лучше воспользоваться кнопкой *Надпись* вкладки *Вставка*.

При нажатии данной кнопки откроется окно, в котором представлены различные макеты надписи. Выберите нужный макет и в появившейся рамке наберите текст. Рамку надписи можно редактировать аналогично редактированию фигуры. Для этого следует выделить надпись, навести указатель мыши на ее границу и, открыв контекстное меню надписи нажатием правой клавиши мыши, выбрать команду *Формат надписи*. Здесь размещены все элементы управления форматом надписи: фоновый цвет; цвет, тип и толщина рамки надписи; размеры внутренних полей между текстом и внешней рамкой поля *Надпись*. Здесь же можно убрать границу поля рамки, если она не нужна.

Объекты WordArt. *WordArt* – это графический текст, который используется для оформления документа. Чтобы добавить объект *WordArt* в текст, необходимо выбрать пункт *WordArt* во вкладке *Вставить* (рис. 6.28).

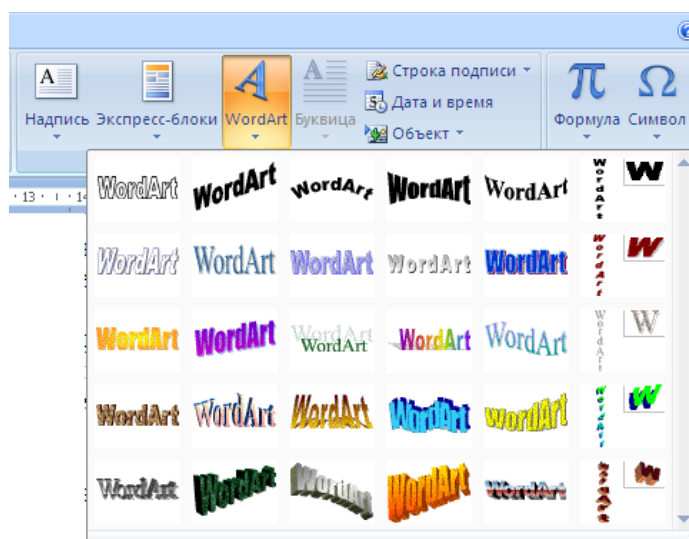


Рисунок 6.28 – Объекты WordArt

Поскольку объекты *WordArt* относятся к графическим объектам текстового процессора, к ним применимы все инструменты редактирования рисунков, размещенные в разделе *Формат* до-

полнительной вкладки *Работа с рисунками*, которая появляется при выделении объекта (рис. 6.29).

Работа с рисунками. Для вставки готового рисунка в документ необходимо нажать кнопку *Рисунок* в разделе *Иллюстрации* вкладки *Вставка*, в открывшемся окне выбрать файл с рисунком и нажать кнопку *Вставить*.

В том случае, если не оказалось нужного рисунка, можно воспользоваться библиотекой рисунков текстового процессора Clip Gallery – она находится по адресу: Диск C / ProgramFiles / MicrosoftOffice / ClipGallery.

В зависимости от версии текстового процессора название библиотеки рисунков может различаться (например, Clipart, Cag-Cat и т.п.).

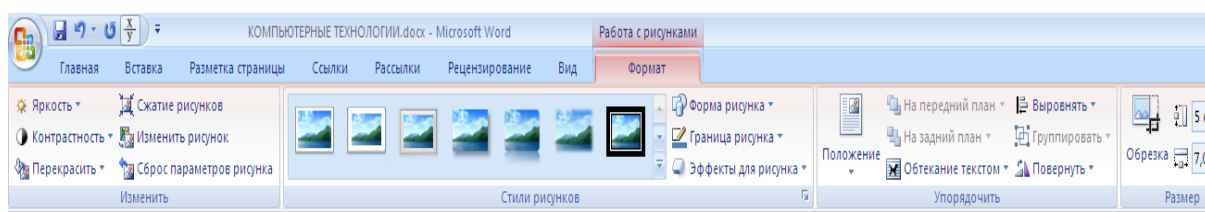


Рисунок 6.29 – Средства редактирования рисунка

Редактирование рисунка позволяет изменить яркость и контрастность, стиль и размеры (при этом можно просто изменить размеры и обрезать ненужное), произвести сжатие и поворот. Средства для редактирования рисунка размещены в разделе *Формат* дополнительной вкладки *Работа с рисунками* (рис. 6.29).

6.2 Основные принципы работы с табличным процессором MicrosoftOfficeExcel

Основные понятия

Электронные таблицы – это программа для создания и использования документов с автоматическим расчетом вносимых данных и построением диаграмм на основе полученных или введенных данных.

Документ программы Excel называется *рабочей книгой*. *Рабочая книга* – это многостраничный документ, страница которого называется *рабочим листом*. Каждая рабочая книга содержит 16 рабочих листов. При загрузке программы открывается книга с

тремя листами, при необходимости листы можно добавить. Имя листа и его номер указаны на ярлыке листа в нижней части окна программы. Если имя листу не было присвоено, он называется просто Лист 1, Лист 2 и т.д.

Рабочий лист представляет собой таблицу, состоящую из ячеек, которые образуются при пересечении строк и столбцов. Каждый лист рабочей книги содержит 256 столбцов и 65 536 строк.

Столбцы называются латинскими буквами A, B, ..., Z, но так как в латинском языке всего 26 букв, то первые 26 столбцов называются этими буквами, в названиях следующих 26 столбцов сначала ставится буква A (AA, AB, ..., AZ), к названиям последующих 26 добавляется буква B (BA, BB, ..., BZ) и т.д.

Строки нумеруются по порядку сверху вниз.

Каждая ячейка имеет собственный адрес. *Адрес ячейки* состоит из имени столбца и номера строки, например D32, AM14.

Диапазоном называется две и более ячейки. Адрес диапазона состоит из адреса левой верхней и правой нижней ячеек, разделенных двоеточием (например, A1:C4).

В ячейки рабочего листа могут быть введены *данные четырех типов*:

- числа;
- тексты;
- дата и время;
- формулы.

Числа делятся на положительные и отрицательные, целые и дробные. В дробных числах целая часть от дробной отделяется запятой. При вводе обыкновенной дроби (например, $\frac{3}{4}$) программа делит числитель на знаменатель и выводит результат в ячейку.

Текст представляет собой набор символов, включая цифры. Максимальная длина текста в одной ячейке составляет 268 435 456 символов. Если длина текста больше ширины ячейки, то он накладывается поверх соседних ячеек, хотя хранится в одной ячейке.

Тип *Дата и время* позволяет указывать на листе дату и время как отдельно, так и вместе, а также проводить расчеты с датами и временем. При вычислениях этот тип данных принимает сутки за единицу. И относится это не только к датам, но и ко времени. Например, время 12:30 рассматривается программой,

как 0,52083 суток, а уже потом выводится в ячейку в привычном для пользователя виде.

Формулы состоят из данных и операторов. Сама формула отражается только в строке формул, а в ячейке виден результат выполнения формулы (рис. 6.30).

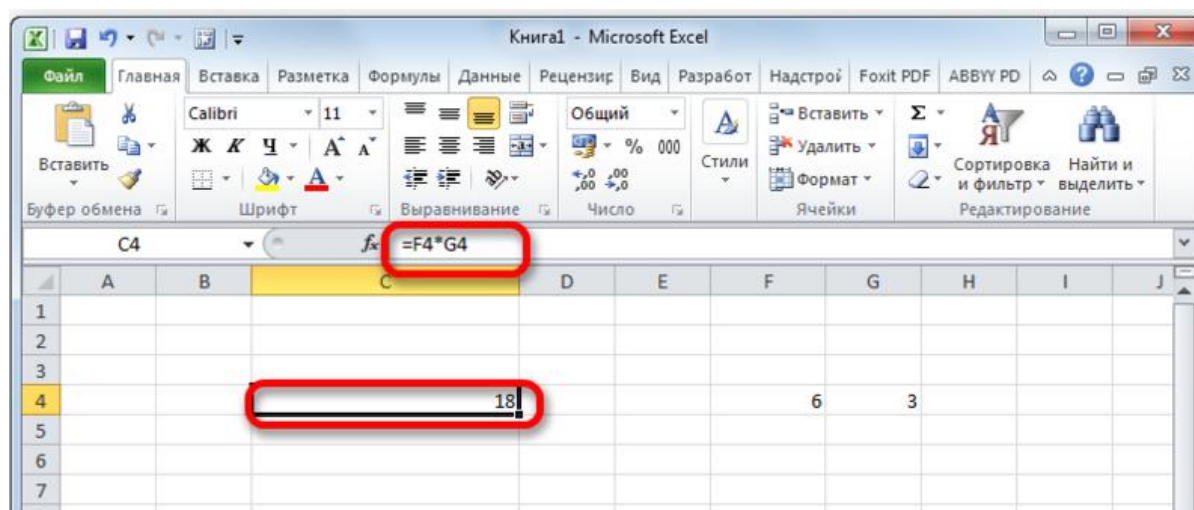



Рисунок 6.30 – Ввод формул

Формула начинается с символа «=» (знак равенства). Ввод формулы завершается нажатием клавиши Enter на клавиатуре, после этого в ячейке отображается результат вычислений. В качестве данных формулы могут выступать тексты, числа, адреса ячеек, а также функции. Приложение Microsoft Excel поддерживает четыре типа операторов: арифметические, текстовые, операторы сравнения и операторы ссылок.

Для *выделения* одной ячейки достаточно кликнуть на ней левой клавишей мыши. Чтобы выделить диапазон ячеек, необходимо провести мышью от одной угловой ячейки до противоположной (угловой) при нажатой левой клавише мыши. Для выделения всего столбца (строки) следует кликнуть мышью на названии столбца (номере строки). При необходимости выделить всю таблицу кликните мышью по ячейке, находящейся на пересечении названий столбцов и номеров строк .

Ввод, редактирование и форматирование данных

Ввод данных. Для ввода данных существует два способа: в строку формул (строка выше названий столбцов) либо в текущую ячейку. Для указания места ввода необходимо поставить курсор. Вводимые данные отображаются как в ячейке, так и в строке формул.

Для редактирования введенных данных необходимо выделить ячейку с данными и кликнуть в строке формул или дважды на текущей ячейке.

Форматирование ячеек. Все вводимые данные в ячейках электронной таблицы размещаются в соответствии с заданным форматом. При создании документа для всех ячеек устанавливается формат *Общий*, перечислим его основные свойства:

- текстовые данные выравниваются по левому краю, а числовые – по правому. Если в вводимых данных кроме цифр встречается символ (например, при вводе дробного числа дробную часть от целой отделили не запятой, а точкой), программа воспринимает запись как текст, и применить арифметические операции к этим данным будет уже нельзя;

- вводимые или полученные данные, длина которых оказалась больше ширины столбца, заменяются на символ «#». В этом случае нужно просто увеличить ширину столбца;

- очень большие числа, полученные в результате вычислений, программа автоматически записывает в экспоненциальном виде (например, $6.e+10$);

- при уменьшении ширины столбца десятичные дроби округляются. Например, если в ячейке записано число 3,1415 и при уменьшении ширины столбца цифра 5 не помещается в ячейку, программа округлит это число до 3,142;

- при вводе данных, обозначающих денежные суммы или проценты, не стоит вручную дописывать символ процентов % или единицы измерения денежных сумм. Чтобы программа могла использовать эти данные в формулах, т.е. производить какие-то операции с ними, необходимо задать ячейкам с этими данными соответствующий формат.

Для изменения формата ячейки следует кликнуть по кнопке *Формат* в разделе *Ячейки* вкладки *Главная* и выбрать команду *Формат ячеек*. Откроется окно с несколькими вкладками (рис. 6.31).

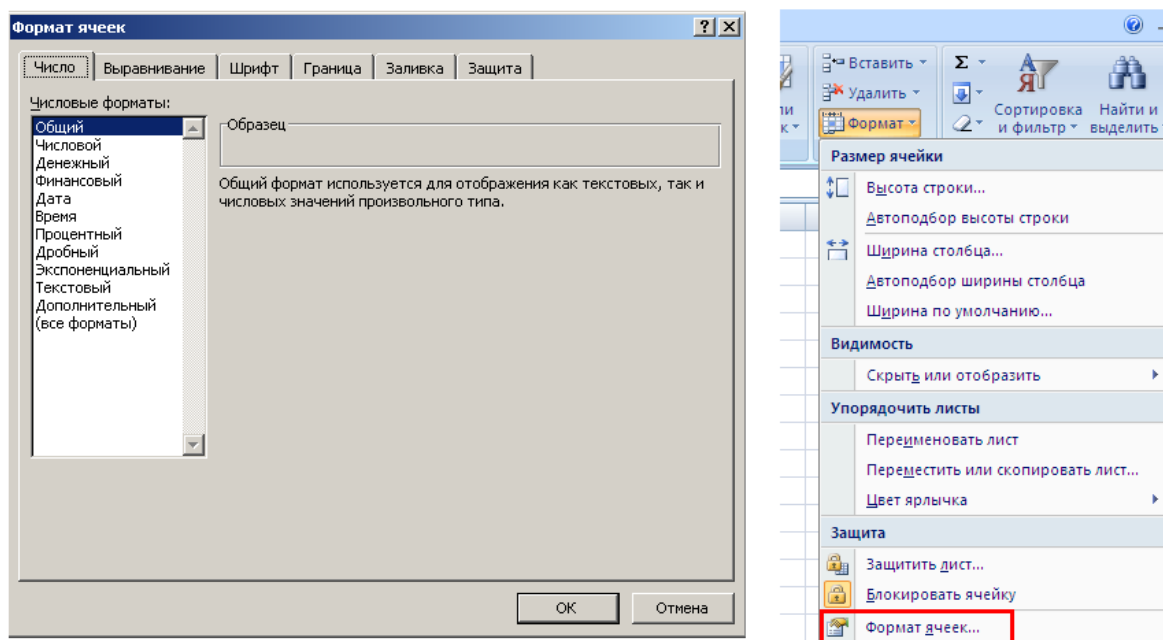


Рисунок 6.31 – Изменение формата ячеек

Кратко опишем каждую из этих вкладок.

Вкладка *Число* позволяет изменить формат представления данных. В зависимости от того, какого типа данные вы вводите в ячейки, необходимо задать ячейкам соответствующий формат, чтобы программа правильно выполняла действия с этими данными и не выдавала ошибок. Как видно из рисунка 6.31, для каждого типа данных предлагаются определенные форматы. Задав ячейкам *Процентный* формат, уже не надо ставить символ процентов. Аналогично для *Денежного* формата программа автоматически выставит единицы измерения денег. При выборе *Числового* формата можно сразу задать формат для целых чисел или для дробных, при этом для дробных определить количество знаков после запятой.

Во вкладке *Выравнивание* размещены параметры для расположения данных в ячейке (по ширине, по центру и т.п.), а также параметры переноса данных на другую строку и объединения ячеек.

В отличие от правил работы текстового процессора при нажатии клавиши *Enter* курсор не переводится на новую строку. С помощью этой клавиши происходит ввод информации и переход на другую ячейку, поэтому, чтобы разместить текст в одной ячейке в несколько строк, необходимо активировать параметр *переносить по словам*.

Для ввода заголовков и размещения их по центру таблицы или какого-то диапазона удобно ввести название и активировать параметр *объединить ячейки* (это можно сделать в любом порядке). Здесь же можно изменить направление текста и его ориентацию (ввести текст под любым углом).

Вкладка *Шрифт* очень похожа на соответствующую вкладку текстового процессора и позволяет менять параметры шрифта.

Для задания границ таблицы и отдельных ячеек служит вкладка *Границы*. Обратите внимание на то, что границы ячеек видны нам только на экране. Если вы их не задали, то при распечатке данные будут размещены на чистом белом листе. Поэтому, чтобы таблица выглядела, как таблица, необходимо задать границы. Здесь вы можете выбрать тип, толщину, цвет линий для границ и место их размещения (только внешние границы или внутренние, линию по диагонали и т.д.).

Во вкладке *Заливка* можно выбрать не только цвет однородной заливки ячеек, но и способы заливки (аналогично соответствующей вкладке в текстовом редакторе).

Вкладка *Защита* позволяет скрыть формулы и защитить ячейки. Однако защита ячеек возможна только при установке защиты листа.

Вычисления в таблицах

Для вычислений в табличном процессоре используются формулы. В качестве данных, как было сказано выше, могут применяться тексты, числа, адреса ячеек и функции. В таблице 6.2 приведены примеры операторов для разных типов данных.

Результатом выполнения арифметического оператора всегда является число, результатом выполнения текстового – текст. В качестве результата оператора сравнения могут быть только два значения: *Ложь* или *Истина*.

Порядок выполнения арифметических операторов стандартный – как в арифметике.

Приоритеты выполнения операторов сравнения:

$=$, $><$, $<=$, $>=$, $<>$.

Таблица 6.2 – Типы операторов

Тип оператора	Обозначение	Значение
Арифметический	+	Сложение
	-	Вычитание
	*	Умножение
	/	Деление
	%	Доля
	^	Возведение в степень
Операторы сравнения	=	Равно
	>	Больше
	<	Меньше
	<=	Меньше или равно
	>=	больше или равно
	<>	Не равно
Текстовый	&	Объединение

Абсолютные и относительные ссылки. В приложении Excel в формулах могут использоваться адреса ячеек двух типов: *абсолютные* и *относительные*. По умолчанию ссылки на ячейки в формулах рассматриваются как относительные, но приложение построено так, что при копировании формулы адреса в ссылках автоматически изменяются в соответствии с относительным расположением исходной ячейки и создаваемой копии.

Чтобы запретить изменение адреса, необходимо использовать *абсолютные ссылки*. Если необходимо запретить изменение названия столбца, перед названием ставят символ «\$» (знак доллара): например, вместо G4 следует написать \$G4. Для запрета изменения номера строки знак доллара ставится перед номером строки: например, N\$12. При необходимости запрета изменения и названия столбца, и номера строки ставится два знака доллара: \$V\$6.

Способы копирования данных. Для ускорения работы при вводе одинаковых данных в несколько ячеек рациональнее их копировать. Приложение Excel допускает несколько способов копирования.

Первый способ – стандартный, когда данные копируются в буфер обмена, а затем вставляются в нужный диапазон. В этом случае важно помнить, что перед вставкой данных необходимо выделить ровно столько строк и столбцов, сколько занимают данные для копирования, а затем выбрать команду *Вставить*.

Второй способ, его называют *автозаполнение*, осуществляется следующим образом:

- выделить ячейку с данными для копирования;

- подвести указатель мыши к нижнему правому углу рамки ячейки на черный маленький квадрат (указатель изменит вид на +);
- при нажатой левой клавише мыши выделить диапазон ячеек, в который планируется скопировать данные.

Обратите внимание, что при выделении ячеек для копирования они не подсвечиваются голубым цветом, как при обычном выделении (рис. 6.32).

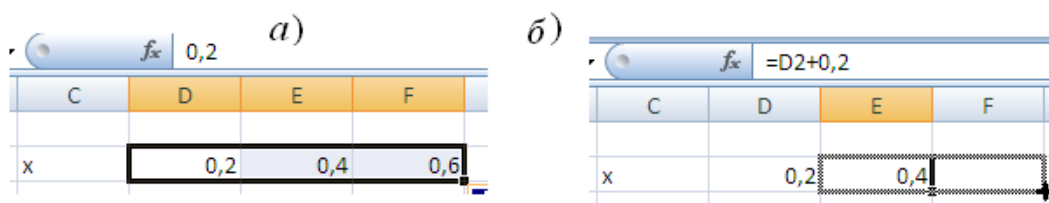


Рисунок 6.32 – Выделение ячеек:

a – стандартное выделение; *б* – выделение при копировании

Если копируется формула, содержащая адрес ячейки, следует помнить, что при автозаполнении относительные ссылки изменяются в соответствии с относительным расположением копии и оригинала, абсолютные остаются без изменений (табл. 6.3).

В том случае, когда необходимо заполнить ячейки числами, буквами, днями недели или месяцами, можно также воспользоваться автозаполнением. Только теперь необходимо ввести данные в две ячейки подряд, чтобы программа могла определить, с чего начинать и с каким шагом копировать. А далее также выделить, только теперь уже эти две ячейки с введенными данными, навести указатель мыши на черный квадрат в правом нижнем углу второй ячейки и выделить диапазон для вставки данных. Например, если в первые две ячейки ввести 1 и 2, то программа вставит числа по порядку. Если ввести 1 и 3, то будут введены только нечетные числа. Аналогично для дней недели и месяцев.

Таблица 6.3 – Правила изменения ссылок при автозаполнении

Ссылка в исходной ячейке	Ссылка в следующей ячейке			
	При заполнении вправо	При заполнении влево	При заполнении вниз	При заполнении вверх
C4 (относительная)	D4	B4	C5	C3
\$B2 (абсолютная по столбцу)	\$B2	\$B2	\$B3	\$B1
D\$3 (абсолютная по строке)	E\$3	C\$3	D\$3	D\$3
\$F\$5 (абсолютная)	\$F\$5	\$F\$5	\$F\$5	\$F\$5

Третий способ – перетягивание мышью. Выделите диапазон, данные которого следует скопировать, наведите указатель мыши на границу диапазона в любом месте (указатель мыши изменит вид на четырехглавую стрелку) и перетяните ячейки при нажатой левой клавише мыши в то место, куда необходимо скопировать (точнее перенести) данные.

Использование стандартных функций. Под *стандартными функциями* в приложении Excel понимают заранее созданные формулы, использование которых упрощает выполнение сложных вычислений. Обычно стандартные функции используются в тех случаях, когда необходимо написать длинную формулу, например вычислить сумму данных большого диапазона ячеек, или когда невозможно провести вычисления с помощью формул, например логическая функция *Если*.

Стандартные функции делятся на несколько категорий: арифметические, статистические, логические, финансовые, дата и время, инженерные и т.п.

Функция состоит из двух частей: *имени* и *аргументов*, записанных в круглых скобках. В качестве аргументов могут выступать имена, числа, массивы, ссылки на ячейки и даже формулы. При использовании *нескольких аргументов* они отделяются друг от друга точкой с запятой.

Некоторые функции не содержат аргументов, например ПИ, ИСТИНА. Но даже если аргументов нет, круглые скобки обязательно записываются.

Примеры функций для вычисления:

=СУММ(A1:C4) – сумма данных диапазона A1:C4;

=СРЗНАЧ(14;18;-32;78;-54) – среднее арифметическое чисел;

=СЧЕТ(4;6) – количество указанных чисел в выделенном диапазоне;

=ПРОИЗВЕД(A1;C4;E3:H5) – произведения данных, в качестве аргументов указаны ссылки на отдельные ячейки и диапазон ячеек;

=СУММ(SIN(A1*ПИ()); 2*COS(A2*ПИ())) – суммы, в качестве аргументов выступают формулы;

=C2*ПИ() – использование функций без аргументов;

=СЧЕТЕСЛИ(C2:C100;"=студент") – количества студентов в диапазоне данных C2:C100;

$=\text{СУММ}(\text{ЕСЛИ}(A2:A100=\text{МАКС}(A2:A100);1;0))$ – вычисление максимума в ячейках $A2–A100$; затем сравнение всех значений в перечисленных ячейках с ним; в случае равенства функция ЕСЛИ принимает значение 1, в противном случае – 0; в заключение суммирование единиц.

Для ввода функции можно воспользоваться двумя способами: кнопкой *Вставить функцию* в разделе *Библиотека функций* вкладки *Формулы* либо аналогичной кнопкой слева от строки формул (рис. 6.33).

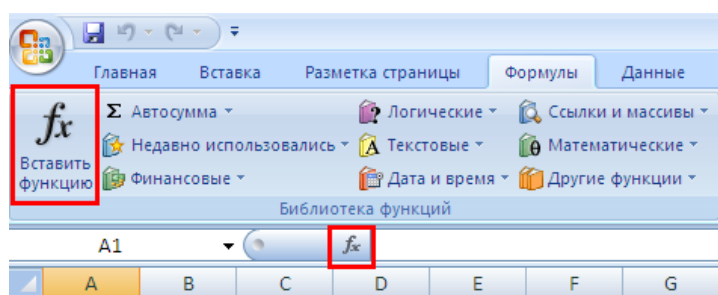


Рисунок 6.33 – Вставка функции

Откроется окно *Мастера функций*, в котором следует найти нужную категорию, а затем выбрать функцию. Ниже списка функций появится подсказка, как правильно записать аргументы функции и какое действие выполняет данная функция (рис. 6.34).

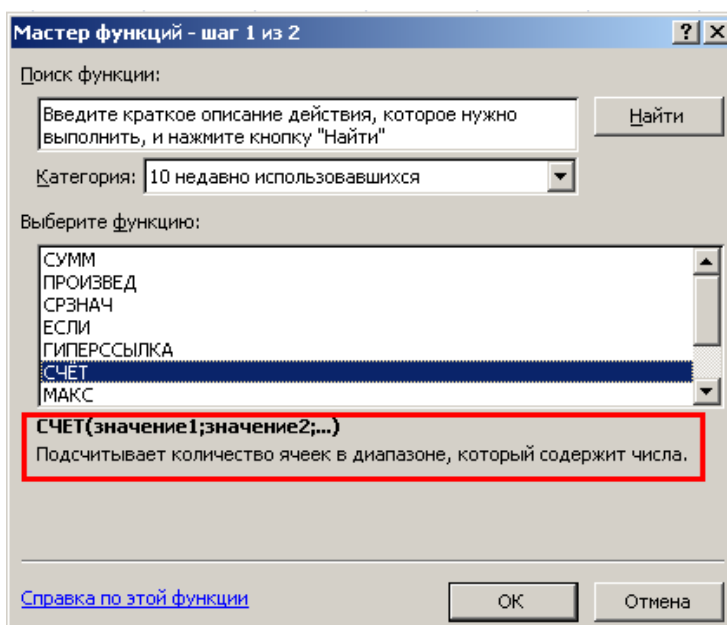


Рисунок 6.34 – Окно Мастера функций

Работа с диаграммами

Построение диаграмм. *Диаграмма* – это графическое представление числовых данных. Она представляет собой вставной объект, внедренный на один из листов рабочей книги. Допустимы два варианта размещения диаграммы: *на том же листе*, на котором находятся данные, или *на другом листе* (программа автоматически создает новый чистый лист без деления на ячейки в рабочей книге). Диаграмма сохраняет связь с данными, на основе которых она построена, и при обновлении этих данных изменяет свой вид.

Для построения диаграммы необходимо открыть раздел *Диаграммы* вкладки *Вставка*. Откроется окно, в левой части которого перечислены все возможные шаблоны, в правой – варианты представления данных для каждого шаблона. Приложение Excel предлагает 11 различных шаблонов: гистограмма, график, круговая и т.д. Остается выбрать подходящий шаблон и нажать кнопку ОК.

Область диаграммы. Рассмотрим основные элементы области диаграммы (рис. 6.35).



Рисунок 6.35 – Область диаграммы

Заголовок диаграммы (1) показывает, что за данные представлены на диаграмме.

Вертикальная ось (ось значений Y) и ее название на оси Y отображаются значения столбцов (2).

Горизонтальная ось (ось X) и ее название, ось X представляет категории (3).

Легенда (4) связана с графическим отображением данных в области графика. Каждая конкретная запись в легенде содержит ключ легенды для ссылки на данные. *Ключ легенды* – это одиноч-

ный цветной или узорчатый маркер в легенде. Справа от каждого ключа легенды находится имя, идентифицирующее данные, представленные конкретным ключом.

Область графика (5), где расположен *Ряд данных*, состоящий из связанных точек (значений) на диаграмме и отражающий зависимость данных друг от друга.

Выбор данных. Существует два способа выбора данных для построения диаграммы.

Выделить мышью диапазон с данными. Если данные не образуют единый прямоугольный диапазон, то после выделения первой части данных следует выделить вторую при нажатой клавише Ctrl. Затем вставить диаграмму.

Сначала вставить диаграмму, затем в контекстном меню диаграммы (открывается щелчком правой клавиши мыши в области диаграммы, не на графике) выбрать команду *Выбрать данные*.

Оформление и размещение диаграммы. Как только диаграмма будет построена, откроется дополнительная вкладка *Работа с диаграммами* с тремя вложенными вкладками: *Конструктор*, *Макет* и *Формат*.

Вкладка *Конструктор* позволяет выбрать или изменить тип диаграммы и макет, стиль диаграммы и ее расположение. Особое внимание необходимо уделить кнопке *Переместить диаграмму* в разделе *Расположение*. Именно здесь указывается, будет ли диаграмма размещена на одном листе вместе с данными либо будет автоматически создан дополнительный новый лист в рабочей книге для ее размещения (рис. 6.36).

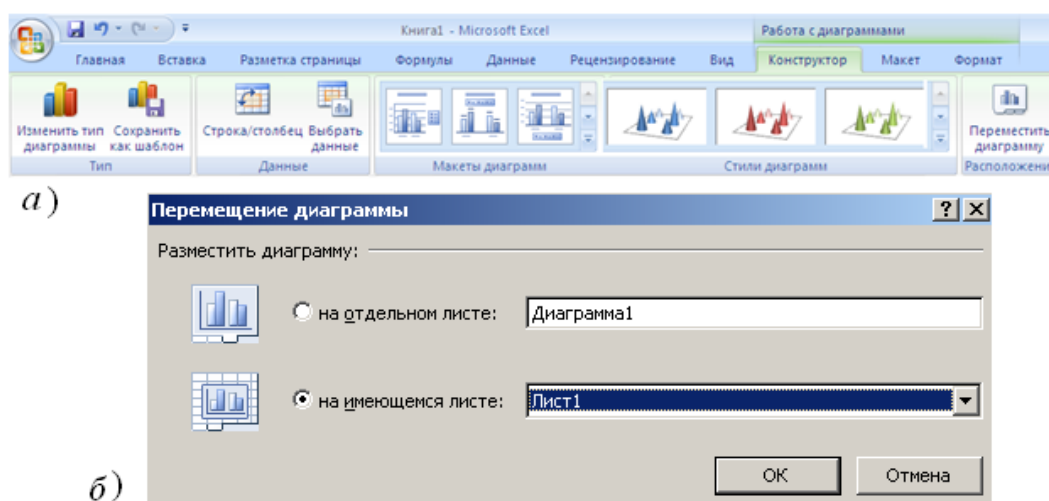


Рисунок 6.36 – Вкладка *Работа с диаграммами*:
а – дополнительная вкладка *Конструктор*; б – раздел *Расположение*

Вкладка *Макет*, пожалуй, одна из основных вкладок, предназначенная для оформления области диаграммы (рис. 6.37).

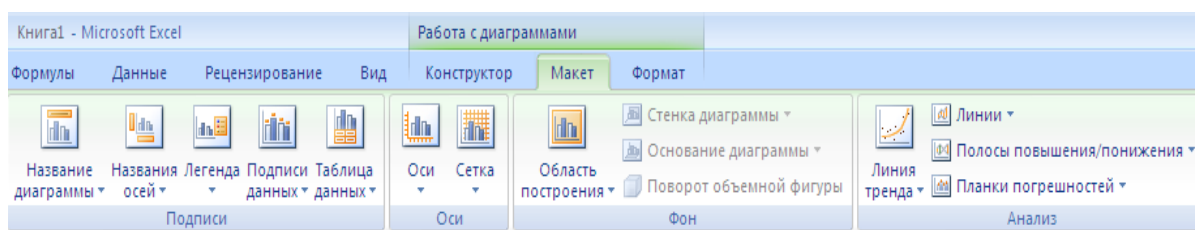


Рисунок 6.37 – Вкладка *Макет*

Правильно и полностью оформленная область диаграммы должна включать:

- название диаграммы;
- названия осей (вертикальной и горизонтальной);
- легенду;
- линии сетки (основные и дополнительные), при необходимости;
- подписи данных; они ставятся над графическими объектами (например, маркерами) в области графика.

Все эти параметры задаются или отменяются во вкладке *Макет*.

Во вкладке *Формат* размещены команды, позволяющие изменить цвет текста, контура и фона области диаграммы и области графика (рис. 6.38).

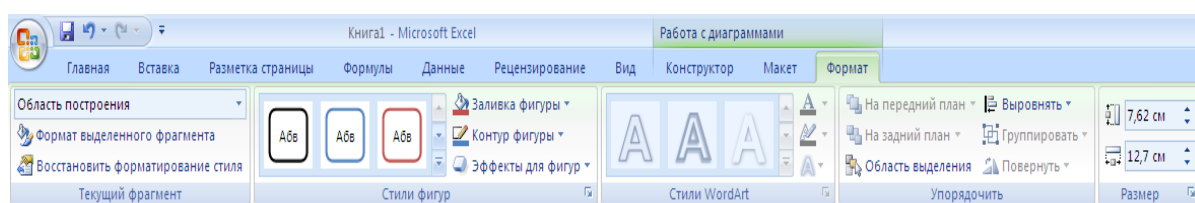


Рисунок 6.38 – Вкладка *Формат*

Редактирование диаграммы. Построенную диаграмму можно менять. Поскольку она состоит из набора отдельных элементов, то для их редактирования необходимо выделить нужный элемент щелчком мыши и выбрать нужный параметр либо в нужной вкладке, либо в контекстном меню этого элемента.

Для удаления диаграммы можно удалить рабочий лист, на котором она расположена (раздел *Ячейки* вкладки *Главная*). В этом случае будет удалена не только диаграмма, но и все данные этого листа. Для удаления только диаграммы ее следует выделить и нажать клавишу DELETE.

Вопросы и задания для самоконтроля

1. Как набрать символ, которого нет на клавиатуре?
2. Что такое неразрывный пробел? Как его поставить?
3. Какие символы относятся к непечатаемым? Как их увидеть?
4. Что такое стиль и где он используется?
5. Что называют колонтитулом? Какую информацию записывают в колонтитулы?
6. Какие виды списков можно создать в приложении MS OfficeWord?
7. Как можно вставить таблицу в документ?
8. Что понимается под форматированием таблицы?
9. Какие типы графических объектов доступны в приложении MS OfficeWord?
10. Как называется документ MS OfficeExcel?
11. Какие данные может содержать ячейка?
12. Как увидеть формулу, записанную в ячейку?
13. Как сделать так, чтобы в ячейке отображался не результат вычислений по формуле, а сама формула?
14. Какие виды ссылок на ячейки существуют в программе электронных таблиц?
15. Что означает символ # в ячейке?
16. Перечислите типы операторов, которые вы знаете.
17. Какие способы копирования вы знаете?
18. Опишите процесс создания диаграммы.
19. Перечислите элементы области диаграммы.
20. Наберите и отформатируйте текст по приведенному в конце задания образцу «Задания» в MS OfficeWord.
21. Создайте шаблон «Тест».
22. В шаблоне установите следующие параметры полей: верхнее – 1,3 см, остальные – 1 см.
23. Создайте стиль первого заголовка: шрифт – *Arial*, жирный, размер шрифта – 13, выравнивание – по центру.
24. Создайте стиль второго заголовка: шрифт – *Arial*, жирный, курсив, размер шрифта – 12, выравнивание – по левому краю с отступом 1 см.
25. Стиль абзаца определите следующим образом: измените размер шрифта на 11.
26. Вставьте таблицу.

27. В ячейки таблицы введите текст. Для ввода формул воспользуйтесь редактором формул. В редакторе формул измените размер шрифта на 11.

28. Отформатируйте текст, используя стили и список для нумерации задач.

29. Создайте документ MSOfficeExcel. Добавьте на лист рисунок с формулами для нахождения длины окружности и площади круга.

30. В ячейке A3 запишите значение радиуса $R=5,6$. В ячейки A4, A5 введите формулы расчета длины окружности с этим радиусом и площади круга.

31. Вычислите и сравните между собой значения выражений $2 \cdot 16 + \frac{64}{8}$ и $\frac{2 \cdot (16+64)}{8}$.

32. Вычислите значения функции $y=x^2$, изменяющейся на отрезке $[-1;1]$ с шагом 0,2, постройте ее график.

33. Измените тип линии графика, ее цвет, задайте цветной фон для области диаграммы.

34. Вычислите значения функции $y = x^2 + 2$, изменяющейся на отрезке $[-1;1]$ с шагом 0,2, постройте графики предыдущей и этой функций в одной координатной плоскости.

Образец «Задания»

A) Отметьте номер правильного ответа в бланке ответов

Таблица 1 – Тест

Задание	Вариант ответов	
1А. Если 20% числа равны $(\sqrt{3} - \sqrt{2}) : (\sqrt{3} + \sqrt{2}) + 2\sqrt{6}$, то это число равно	1) 15; 2) 20; 3) 25;	4) 30; 5) 35
2А. Упростите выражение: $\frac{a^2 - b^2}{a + b - 2\sqrt{ab}} \cdot \frac{\sqrt{a} - \sqrt{b}}{\sqrt{a} + \sqrt{b}}$	1) $a + b$; 2) $\sqrt{a} + \sqrt{b}$; 3) $a - b$;	4) $\sqrt{a} - \sqrt{b}$; 5) $2a$
3А. Количество целых значений параметра а, при которых абсцисса и ордината вершины параболы $y = (x - 2a)^2 - 2a^2 - 8a - 15$ положительны, равно	1) 0; 2) 1; 3) 2;	4) 3; 5) 4
4А. Чему равна сумма корней уравнения $\frac{2x^2 + 3x + 2}{x + 1} = x^2 - 3x - 5$?	1) 4; 2) -2; 3) 5;	4) 3; 5) -5
5А. Все решения неравенства $x^2 + \sqrt{x^2} < \frac{1}{4}$ заполняют на числовой оси промежутки, длина которого равна	1) $\sqrt{2} - 1$; 2) $\sqrt{2}$; 3) 1;	4) $\frac{\sqrt{2} - 1}{2}$; 5) $\frac{\sqrt{2}}{2}$

Б) Напишите правильный ответ в нижнем правом углу бланка ответов:

1Б. Найдите число решений неравенств:

$$1. \frac{8x+2}{(x^2+2x+1)(x^2+x-6)} \geq \frac{1}{x^2-x-2},$$

$$2. \sqrt[3]{2x^2 - 11x - 21} \cdot \sqrt{4 - x} \leq 0,$$

$$3. (x^2 + 25) \cdot \log_{0,2}(x - 3) + \frac{10x}{5} \geq 0.$$

6.3 Математический пакет программ MathCad

Математический пакет программ MathCad – это интегрированная система, включающая связанные между собой текстовый редактор, вычислительный и символьный процессоры. Он относится к универсальным системам, так как кроме проведения вычислений он обеспечивает возможность создания качественных текстов, а также сохранения документа в формате web-страниц.

Пакет *MathCad* построен на основе принципа *WYSIWYG* (*What You See Is What You Get* – «то, что видите, то и получаете»), что обеспечивает простоту его использования. Основное его назначение – выполнение инженерных расчетов.

Среди главных возможностей данной системы можно выделить:

- проведение расчетов с числами, матрицами и векторами;
- вычисление пределов, производных и интегралов;
- решение линейных и нелинейных систем как алгебраических, так и дифференциальных уравнений;
- построение и редактирование диаграмм;
- элементы программирования.

Основные понятия

Документ программы *MathCad* называется *рабочим листом*. Его часто называют «живым документом», основываясь на том, что все введенные на листе выражения и диаграммы доступны для изменений и внесенные в одном месте листа изменения распространяются на весь лист в направлении сверху вниз и слева направо.

Окно программы

Окно программы имеет характерный для многих программ, в том числе офисных приложений, вид. На рисунке 6.39 представлены основные элементы окна MathCad.

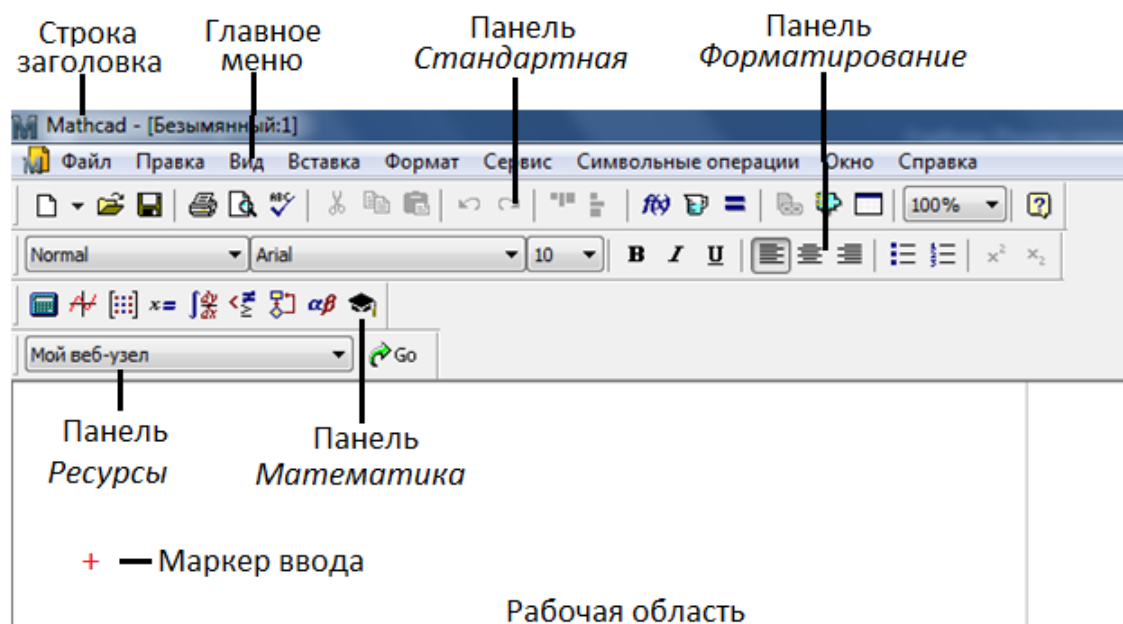


Рисунок 6.39 – Окно программы MathCad

Как правило, по умолчанию при загрузке программы открыты 4 основные панели инструментов, показанные на рисунке 6.39. Для открытия новых панелей (или закрытия ненужных) следует воспользоваться командой *Вид / Панели / Название панели*.

Остановимся более подробно на панели *Математика*, поскольку основные действия выполняются именно с ее использованием (рис. 6.40). Данная панель может располагаться как в верхней части окна программы, как показано на рисунке 6.39, так и снизу, слева или справа, а также может отображаться в отдельном окне. Панель Математика содержит 9 палитр, с помощью которых вводятся данные, производятся расчеты, строятся графики. Для открытия нужной палитры необходимо кликнуть левой клавишей мыши на соответствующей кнопке. Открывают не все палитры, а лишь те, которые в данный момент необходимы для работы. С помощью стандартного приема можно перемещать палитры по рабочей области. Далее опишем кратко каждую из этих палитр.

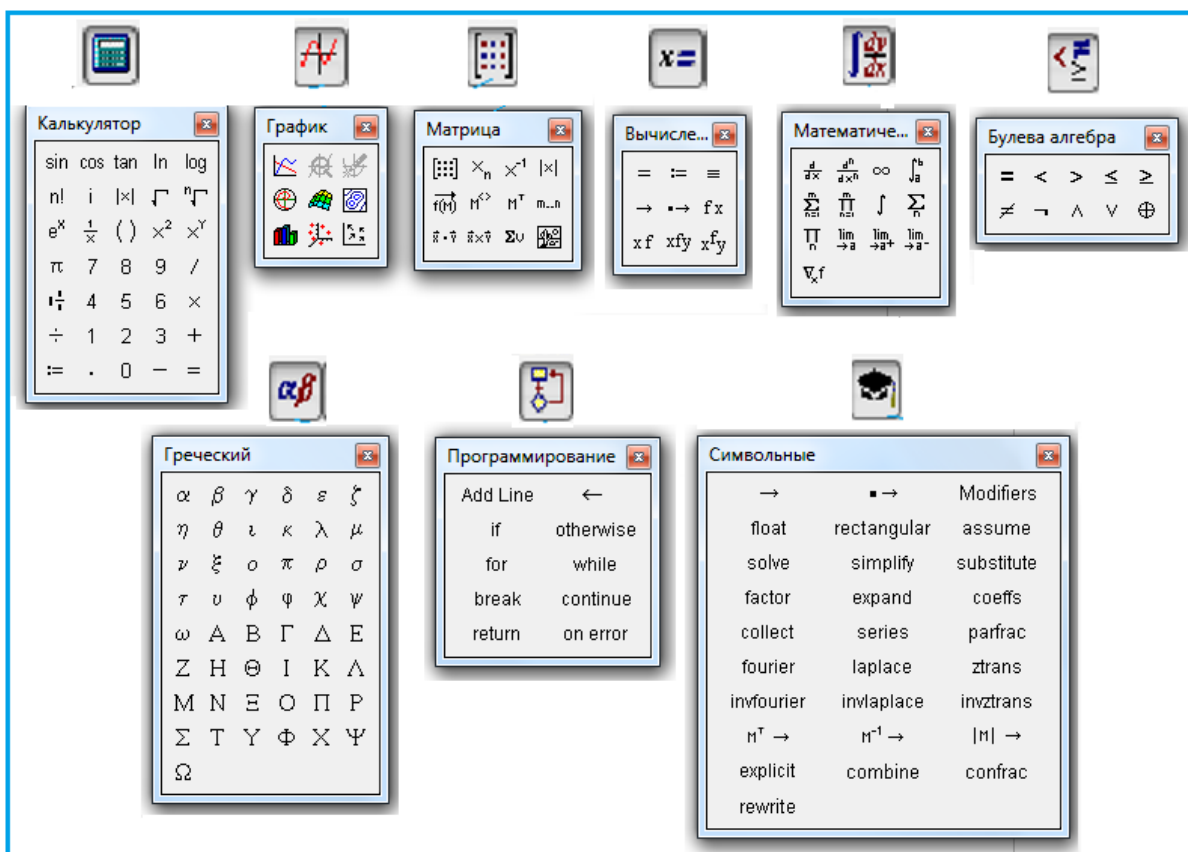


Рисунок 6.40 – Кнопки панели инструментов Математика

Кнопка *Калькулятор* открывает палитру *Калькулятор*, которая позволяет вводить математические символы и операции. Она дублирует действия обычного калькулятора, поэтому назначение символов палитры вполне понятно. Если знаки арифметических действий есть на клавишах клавиатуры (например, знаки сложения и вычитания), можно воспользоваться ими.

Палитра *График* открывается с помощью одноименной кнопки и служит для построения графиков. На палитре представлены все имеющиеся виды диаграмм, которые можно построить в программе MathCad. Для построения графика необходимо установить маркер ввода в нужное место рабочей области и выбрать на панели его тип.

Палитра *Матрица* используется для ввода векторов и матриц и проведения операций над ними.

Доступ к командам управления вычислением и вставка пользовательских операторов осуществляются с помощью палитры *Вычисление*.

На палитре *Математический анализ* размещены знак бесконечности и операции высшей математики, такие как вычисле-

ние пределов, производных, интегралов, алгебраической суммы и алгебраического произведения.

Знаки логических операций и команды, задающие уравнения и неравенства расположены на палитре *Булева алгебра*.

Поскольку на клавиатуре нет букв греческого алфавита, в большинстве программ используются дополнительные возможности для вставки этих символов. Так и в программе MathCad палитра *Греческий* обеспечивает возможность ввода греческого алфавита.

Несмотря на то, что пакет программ называют математическим, он будет более полезен инженерам, нежели математикам и программистам в силу своих возможностей. Палитра *Программирование* содержит операторы, необходимые для создания программных модулей. Можно, например, использовать оператор ветвления в вычислениях. Однако для полноценного программирования необходимо гораздо больше возможностей.


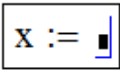

Палитра *Символьные* предназначена для вставки символьных выражений и проведения вычислений над ними.

Большую часть окна программы занимает *Рабочая область*, в которой вводят данные, выполняют вычисления и строят графики.

Рабочий лист программы может содержать формулы, графические области и текстовые блоки. Установлен строгий порядок вычислений: слева направо и сверху вниз. Поэтому нельзя располагать блоки в произвольном порядке.

Ввод данных производится в месте расположения курсора. В зависимости от типа вводимых данных курсор может иметь три различных вида (табл. 6.4).

Таблица 6.4 – Виды курсора в зависимости от типа данных

Вид курсора	Описание
	Крестообразный курсор красного цвета означает, что объект не выделен и указывает место ввода данных.
	Синий курсор в виде угла появляется при вводе формулы. Он может иметь правую или левую ориентацию, указывая направление и место ввода. Для изменения ориентации угла используется клавиша Insert. Если уголок левый, данные будут набираться слева от него, если правый – справа. Клавиши курсоров позволяют перемещать уголок вдоль выражения, а клавиша Пробел – изменять размер сторон угла.
	Курсор красного цвета в виде вертикальной черты появляется при вводе текста.

Следует учесть при вводе данных, что программа различает заглавные и прописные буквы.

Операции с областями документа

В зависимости от типа вводимых данных рабочая область окна программы делится на соответствующие области: текстовые (при вводе текста) или математические (при вводе значений или формул).

Для *выделения всей области* следует поставить указатель вне крайней области и при нажатой левой клавише переместить мышь, заключая нужную область в пунктирную рамку.

С помощью команды *Вид / Регионы* можно выделить границы областей. В этом случае фон окрашивается в серый фон, а области будут выделены белым цветом.

Поскольку ввод данных начинается с установки курсора вручную, часто области ввода располагаются на экране в беспорядке. Чтобы *выровнять области*, необходимо выбрать команду *Формат / Выровнять регионы*.

Компоненты математических выражений

К основным компонентам математических выражений программы MathCad относятся: *операторы, константы, переменные, массивы, функции*.

Операторы. Для ввода данных используется *оператор присвоения*. Он может иметь три различных вида:

$:=$ – *локальный оператор присвоения*. Переменная считается неопределенной и ее нельзя использовать, пока не применен оператор локального присвоения.

\equiv – *глобальный оператор присвоения*. Этот оператор можно использовать в любом месте документа. Даже если переменной было присвоено значение с помощью этого оператора в конце документа, она будет иметь это значение в любом его месте.

$=$ – *оператор приближенного равенства*. Его применяют при решении систем уравнений.

Оператор вывода имеет вид знака равенства « $=$ ». Его используют в тех случаях, когда необходимо показать значение переменной или получить результат вычислений (рис. 6.41).

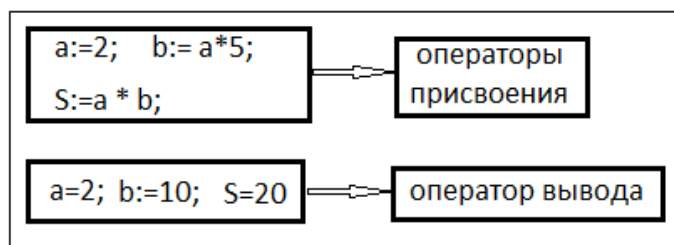


Рисунок 6.41 – Примеры использования операторов присвоения и вывода

При наборе выражений важно отслеживать изменение размеров сторон уголка. Опишем в качестве примера порядок ввода дроби:

$$\frac{123+13,5}{345-132}$$

Набираем числитель:

$$123 + 13.5$$

С помощью нажатия клавиши Пробел охватываем уголком все выражение:

$$123 + 13.5$$

Вводим знак деления – косая черта « / » (ее можно набрать на клавиатуре в английской раскладке, либо на палитре *Калькулятор*).

Набираем знаменатель:

$$\frac{123 + 13.5}{345 - 132}$$

Для получения результата вычисления охватываем уголком все выражение и набираем оператор вывода (знак « = »):

$$\frac{123 + 13.5}{345 - 132} = 0.641$$

Константы. *Константой* в математике называют величину, которая не меняет своего значения. В пакете программ MathCad константа – это объект, имеющий свое оригинальное имя и не меняющий своего значения в ходе вычислений. Имя константы может состоять из букв, цифр и символов, но начинается обязательно с буквы.

Различают следующие *типы констант*:

- *целые* (например, 5, -8);
- *вещественные* (при этом целая часть от дробной отделяется точкой); этот тип констант может быть записан в двух формах – либо с фиксированной точкой (например, 0.5), либо в десятичной форме (например, $2 \cdot 10^4$, для набора показателя степени используется оператор возведения в степень – комбинация клавиш [Shift-6] в английской раскладке);
- *комплексные* (например, $2 + 3i$), при этом между 3 и i знак умножения не ставится;
- *строковые* – последовательность символов, заключенная в кавычки (например, «строковая константа»);
- *системные* – они хранят значения определенных параметров системы.

Кроме этого константы могут иметь размерность или нет. При необходимости вставить размерность необходимо открыть меню *Вставка / Единицы измерения* и выбрать нужную категорию.

Переменные. Переменными называют поименованные объекты с заданными значениями, которые могут изменяться при выполнении вычислений.

Переменные могут быть числовыми, символьными, строковыми и системными, а также глобальными, ранжированными (дискретными), индексными (переменная с индексом) и размерными.

Имена переменных (идентификаторы), также как и имена констант, могут иметь любую длину, содержать латинские и греческие буквы, цифры, символы процента (%), апострофа (') и подчеркивания (_), начинаться с буквы и не могут содержать пробел.

Числовые, символьные и строковые переменные аналогичны соответствующим типам константам.

Системные переменные – это особый класс объектов программы, который по сути не относится ни к константам, ни к переменным. Они определяются при запуске программы, но при необходимости им можно присвоить другие значения. К системным переменным относятся, например:

$$\pi = 3,14159;$$

$$e = 2,7127.$$

ORIGIN[0] – показывает нижнюю границу индекса для векторов и матриц;

TOL[0.01] – задает погрешность числовых расчетов;

PRNPRECISION – устанавливает формат данных при выводе в файл;

PRNCOLWIDTH – устанавливает формат столбца при выводе в файл;

CWD – представляет строковый путь к текущей рабочей папке.

Глобальная переменная – это переменная, значение которой присвоено с помощью глобального оператора присвоения, и ее значение может использоваться в любом месте документа.

Ранжированными или *дискретными* называют переменные, которые принимают ряд значений, меняющихся с определенным шагом от начального до конечного значения. Если шаг не задан, он считается равным 1 или -1.

Для создания ранжированной переменной следует ввести выражение:

Name = $N_1, N_1 + \text{шаг}, \dots, N_n$,

здесь Name – это имя переменной;

N_1 – начальное значение переменной;

шаг – шаг изменения переменной;

N_n – конечное значение переменной;

символ двоеточия «..» вводится с клавиатуры с помощью клавиши «;» в английской раскладке клавиатуры либо кнопкой *Переменная – диапазон (m..n)* палитры *Матрица*.

Если шаг не задан, выражение имеет вид:

Name = N_1, \dots, N_n .

В этом случае при $N_1 < N_n$ шаг равен 1, если $N_1 > N_n$ шаг равен -1.

Чтобы записать ранжированную переменную, нужно ввести:

1) имя переменной;

2) знак присвоения (:=);

3) первое значение диапазона;

4) запятую;

5) второе значение диапазона, которое является суммой первого значения и шага;

6) двоеточие (..) изменение переменной в заданных пределах;

7) последнее значение диапазона.

Любое выражение с ранжированными переменными после знака равенства инициирует появление таблицы вывода, в которую можно и вставлять числовые значения, и корректировать их (рис. 6.42).

$x := 1..5$	$i := 1,2..7$
$x =$	$i =$
1	1
2	3
3	5
4	7
5	

Рисунок 6.42 – Пример ввода и вывода ранжированных переменных

Индексная переменная (или *переменная с индексом*) – это переменная, которой присвоена совокупность чисел, не связанных друг с другом и имеющих свой номер (индекс).

Для ввода индексной переменной следует ввести:

- 1) значения индекса (ранжированной переменной);
 - 2) имя индексной переменной;
 - 3) знак присвоения;
 - 4) значения индексной переменной через запятую
- (рис. 6.43).

$i := 1..5$	
$y_i :=$	
0	
-4	
6	
-8	
0	
$y_1 = 0$	$y_4 = -8$

Рисунок 6.43 – Пример ввода индексной переменной и вывода ее некоторых значений

Размерная переменная – переменная с единицами измерения (скорость, вес, объем и т.п.).

Процедура создания размерной переменной заключается в следующем (на примере объема 25 л):

- 1) присвоить переменной V значение 25;
- 2) выбрать на палитре *Калькулятор* знак умножения « \times » или на клавиатуре знак « $*$ »;
- 3) в меню *Вставка / Единицы измерения* выбрать нужную размерность (окно *Размерность*) и единицу измерения (окно *Единицы*);
- 4) кликнуть на кнопке ОК.

Функции. В системе MathCad можно использовать функции двух типов: *пользовательские* и *встроенные*.

К *пользовательским* относятся функции, определяемые самим пользователем. Требования к именам пользовательских функций точно такие же, как и к именам переменным. Определять пользовательские функции можно как с помощью локального оператора присвоения, так и глобального.

Синтаксис определения пользовательской функции имеет вид:

$\langle \text{имя функции (аргумент}_1, \text{ аргумент}_2, \dots, \text{ аргумент}_n) \rangle := \langle \text{выражение} \rangle$.

Примеры функций:

$$f(x) := \sin(x) + \cos(x);$$

$$g(x,y) := 2x^2 + 3xy - y^2 .$$

Все переменные правой части уравнения должны быть определены выше или являться аргументами функции. Аргументы функции записываются в круглых скобках и отделяются друг от друга запятой.

Аргументы тригонометрических функций указываются в радианах. Перевод градусов осуществляется с помощью переменной *deg*, на пример, чтобы вычислить синус 30° , следует записать так:

$$\sin(30 \cdot \text{deg}) = 0.5$$

Для возведения функции в степень знак показателя степени ставится после имени функции, а не за аргументами:

$$\cos^2(x,y), \log^3(x).$$

Для удобства пользователей программа имеет ряд *встроенных функций*, которые можно разделить на три группы:

- элементарные математические функции;
- специальные математические функции;
- функции с условиями сравнения.

Встроенными называют функции, для которых заранее записана процедура преобразования данных и которые интегрированы сразу в вызывающий код. Система MathCad предоставляет довольно большой объем встроенных функций. Их можно разделить на элементарные и специальные.

К *элементарным* относятся привычные и часто употребляемые пользователем функции, например, тригонометрические, логарифмические и т.п. Здесь нужно учесть, что обозначение некоторых функций в программе отличается от общепринятых, примеры некоторых из них приведены в таблице 6.5.

Особое внимание среди элементарных функций заслуживают функции для работы с комплексными числами и функции комбинаторики.

К основным *функциям комплексного аргумента* относятся:

$arg(z)$ – вычисляет аргумент функции;

$csgn(z)$ – определяет знак комплексного числа:

$$csgn(z) = \begin{cases} 0, & \text{если } z = 0; \\ 1, & \text{если } Re(z) > 0 \text{ или } Re(z) = 0 \text{ и } Im(z) > 0; \\ -1 & \text{в других случаях} \end{cases}$$

$signum(z)$ – определяет знак действительного числа:

$$signum(z) = \begin{cases} 0, & \text{если } z = 0, \\ \frac{z}{|z|}, & \text{в остальных случаях;} \end{cases}$$

$Im(z)$ – определяет значение мнимой части комплексного числа z ;

$Re(z)$ – определяет значение действительной части комплексного числа z .

Таблица 6.5 – Примеры обозначений элементарных встроенных функций

Математическое обозначение	Обозначение в MathCad	Вычисляет
$tg(x)$	$\tan(x)$	Тангенс аргумента x
$arctg(x)$	$\text{atan}(x)$	Арктангенс аргумента x
$\arcsin(x)$	$\text{asin}(x)$	Арксинус аргумента x
$\arccos(x)$	$\text{acos}(x)$	Арккосинус аргумента x
$sh(x)$	$\sinh(x)$	Гиперболический синус аргумента x
$ch(x)$	$\cosh(x)$	Гиперболический косинус аргумента x
$th(x)$	$\tanh(x)$	Гиперболический тангенс аргумента x
$\log_b(x)$	$\log(x,b)$	Логарифм числа x по основанию b
$lg(x)$	$\log(x)$	Десятичный логарифм числа x

Среди функций комбинаторики наиболее часто употребляются:

combin (n, k) – вычисляет число сочетаний из n элементов по $k < n$;

mod (x, y) – остаток от деления x/y со знаком x ;

permut (n, k) – возвращает число размещений из n элементов по k , вычисляя его по формуле: $\text{permut}(n, k) = n!/(n-k)!$, где n и k – целые числа, большие или равные 0.

Специальные функции в системе MathCad предназначены для решения определенных задач. На рисунке 6.44 представлены наиболее часто употребляемые специальные функции [2].

Ввести функцию в MathCad можно одним из трех способов:

- 1) с помощью палитры *Калькулятор*;
- 2) ввести с клавиатуры;
- 3) с помощью команды *Вставка / Функция* (в этом случае в открывшемся окне следует выбрать категорию и найти функцию, здесь же отображается краткая справка по выбранной функции).


название в MathCad	Назначение
Группа "Решение уравнений и систем"	
<i>root</i>	Нахождение корней функции одной переменной
<i>find</i>	Решение систем уравнений
<i>minerr</i>	Приближенное решение систем уравнений
Группа "Решение дифференциальных уравнений "	
<i>rkfixed</i>	Решение систем обыкн. дифф. ур-й 1-го порядка
<i>odesolve</i>	Решение обыкновенного дифф. ур-я n-го порядка
Группа "Оптимизация"	
<i>minimize</i>	Нахождение минимума функции n переменных
<i>maximize</i>	Нахождение максимума функции n переменных
Группа "Вектора и матрицы"	
<i>cols</i>	Число столбцов матрицы
<i>rows</i>	Число строк матрицы
<i>min</i>	Нахождение минимума в массиве (матрице)
<i>max</i>	Нахождение максимума в массиве (матрице)
<i>mean</i>	Среднее арифметическое массива
<i>sort</i>	Сортировка массива
<i>reverse</i>	Обращение (переворачивание) массива

Рисунок 6.44 – Специальные функции MathCad

Массивы. *Массивом* называется поименованная совокупность конечного числа элементов (числовых или символьных), упорядоченных определенным образом и имеющая определенные адреса. По сути массив – это матрица. Массивы бывают одномерные – это векторы (в математике матрица-строка или матрица-столбец) и двумерные (матрица). При использовании вектора-строки или вектора-столбца, имеющих одинаковые значения элементов, в формулах результаты будут разные.


Адресом элемента называется его порядковый номер (индекс) в последовательности элементов. Прежде чем вводить элементы массива необходимо задать нижнюю границу индексации. Это делается с помощью системной переменной *ORIGIN*. Она может принимать только два значения *1* или *0*. Например, ввод *ORIGIN := 1* будет означать, что индексация элементов массива начинается с номера 1.

Ввести элементы массива можно несколькими способами:

С помощью кнопки  на палитре *Матрица*. В этом случае на экране появится шаблон матрицы, в который нужно ввести значения элементов (рис. 6.45).

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad \begin{pmatrix} 2 & 0 & -3 & 7 \\ 0 & -6 & 4 & -1 \end{pmatrix}$$

Рисунок 6.45 – Пример ввода массива с помощью кнопки

Ручной ввод значений поэлементно. Для набора индекса используется кнопка  палитры *Матрица* или фигурная скобка « [» на клавиатуре в английской раскладке. Если необходимо ввести двойной индекс, индексы вводятся через запятую (рис. 6.46).

$$\begin{array}{l} a_1 := 0 \quad a_2 := -1 \quad a_3 := 7 \\ a = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 7 \end{pmatrix} \end{array} \quad \begin{array}{l} b_{1,1} := 2 \quad b_{1,2} := 0 \\ b_{2,1} := -1 \quad b_{2,2} := 4 \\ b = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 4 \end{pmatrix} \end{array}$$

Рисунок 6.46 – Примеры ввода массивов вручную

Использование ранжированных переменных и функций пользователя (рис. 6.47).

$$\begin{array}{l}
 \text{ORIGIN} := 1 \\
 i := 0..1 \quad j := 0..3 \quad i := 1..2 \quad j := 1..4 \\
 D_{i,j} := i - j \quad D_{i,j} := i - j \\
 D = \begin{pmatrix} 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \end{pmatrix} \quad D = \begin{pmatrix} 0 & -1 & -2 & -3 \\ 1 & 0 & -1 & -2 \end{pmatrix}
 \end{array}$$

Рисунок 6.47 – Пример использования ранжированных переменных и функций пользователя

Для работы с векторами и матрицами в системе MathCad также имеется ряд встроенных функций.

К основным векторным функциям относятся:

$\max(X)$ – вычисление наибольшего по значению элемента вектора X (или матрицы);

$\min(X)$ – вычисление наименьшего по значению элемента вектора X (или матрицы);

$\text{length}(X)$ – определение количества элементов вектора X ;

$\text{last}(X)$ – определение номера последнего элемента вектора X ;

$\text{Re}(X)$ – вывод вектора действительных частей вектора X с комплексными элементами;

$\text{Im}(V)$ – вывод вектора мнимых частей вектора X с комплексными элементами.

Примеры векторных функций приведены на рисунке 6.48.

$$\begin{array}{l}
 X := (-1 \ 6 \ -4 \ 5) \\
 \max(X) = 6 \\
 \min(X) = -4
 \end{array}$$

Рисунок 6.48 – Примеры векторных функций

Для работы с матрицами также существует ряд встроенных функций. Их можно разделить на две группы: функции, выполняющие определенные действия с матрицами, и функции, определяющие характеристики матрицы.

К первой группе относятся такие функции, как:

$augment(M1, M2)$ – объединение матриц $M1$ и $M2$, имеющих одинаковое число строк, в одну («бок о бок»);

$stack(M1, M2)$ – объединение матриц $M1$ и $M2$, имеющих одинаковое число строк, в одну, при этом $M1$ располагается над $M2$;

$submatrix(M, r_1, r_2, c_1, c_2)$ – выделяет подматрицу из матрицы M , здесь r_1, r_2 – нижний и верхний номера строки матрицы M ; c_1, c_2 – нижний и верхний номера столбца матрицы M , включаемые в итоговую подматрицу;

$identity(n)$ – создание единичной квадратной матрицы размерности $(n \times n)$;

$diag(V)$ – создание диагональной матрицы, элементы главной диагонали которой равны элементам вектора V ;

$matrix(m, n, f)$ – создание матрицы, у которой (i, j) -й элемент равен $f(i, j)$, где $f(i, j)$ – некоторая функция, $i = 0, 1, \dots, m$ и $j = 0, 1, \dots, n$;

$match(z, M)$ – поиск заданного элемента z и вывод его местоположения (индексов) в матрице M .

Примеры использования функций матриц приведены на рисунке 6.49.

$$M1 := \begin{pmatrix} -1 & 2 \\ -3 & 4 \end{pmatrix} M2 := \begin{pmatrix} 1 & -3 \\ -2 & 5 \end{pmatrix}$$
$$augment(M1, M2) = \begin{pmatrix} -1 & 2 & 1 & -3 \\ -3 & 4 & -2 & 5 \end{pmatrix}$$
$$M := stack(M1, M2) = \begin{pmatrix} -1 & 2 \\ -3 & 4 \\ 1 & -3 \\ -2 & 5 \end{pmatrix}$$
$$submatrix(M, 1, 2, 0, 1) = \begin{pmatrix} -3 & 4 \\ 1 & -3 \end{pmatrix}$$

Рисунок 6.49 – Примеры использования функций матриц

Среди функций, выводящих характеристики матриц, наиболее часто используются следующие:

$cols(M)$ – определяет число столбцов;

$rows(M)$ – возвращает число строк;
 $rank(M)$ – вычисляет ранг матрицы;
 $tr(M)$ – определяет след матрицы (сумму диагональных элементов);
 $mean(M)$ – возвращает среднее значение элементов матрицы;
 $median(M)$ – вычисляет медиану элементов матрицы;
 $sort(V)$ – сортирует элементы вектора V в порядке возрастания их значений;
 $reverse(V)$ – переставляет элементы вектора V в обратном порядке.

Построение графиков

Построение графиков является важным этапом исследования функций, поскольку только график может наглядно отобразить поведение функции на определенном промежутке значений. Система MathCad содержит графический процессор, позволяющий довольно легко и быстро строить различные типы графиков функций, которые условно можно разбить на две группы: двумерные и трехмерные.

К *двумерным* относятся:

- декартовый XY график (XYPlot);
- полярный график (PolarPlot).

Трехмерные графики:

- график трехмерного множества точек (3DScatterPlot);
- трехмерная гистограмма (3DBarPlot);
- график поверхности (SurfacePlot);
- график линий уровня (ContourPlot);
- график векторного поля (VectorFieldPlot).

Как было сказано выше, деление графиков на типы условное, так как система предоставляет возможность не только строить комбинации типов графиков, но и создавать новые типы.

Построение графиков осуществляется с помощью инструментов палитры *График* панели *Математика* (рис. 6.50).

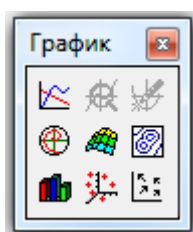


Рисунок 6.50 – Палитра инструментов *График*

Процедура построения одинакова для различных типов графиков: выбирается соответствующий шаблон, затем заполняются поля шаблона.

Кроме шаблонов графиков на палитре имеются две кнопки: *Масштаб* и *Трассировка* (на рис. 6.50 они не активированы). Эти кнопки служат для проведения анализа графика и становятся активными при выделении графика функции одной переменной в декартовой и полярной системе координат.

Существует два способа построения графиков:

- а) по умолчанию (QuickPlot – быстрый график);
- б) график на заданном интервале по точкам (по массиву данных).


Построение двумерных графиков

Система MathCad позволяет строить двумерные графики в декартовой и полярной системе координат. Важно отметить, что построенный график одного типа нельзя переделать в график другого типа.

Построение графика *в декартовой системе координат*

- а) быстрый график

Для построения графика необходимо выполнить следующие действия:

- установить курсор в место построения графика в рабочей области;
- активировать палитру *График*;
- выбрать шаблон двумерного графика в декартовых координатах *График X-Y* с помощью кнопки , при этом левый верхний угол шаблона будет располагаться в месте нахождения курсора (красного крестика) (рис. 6.51);

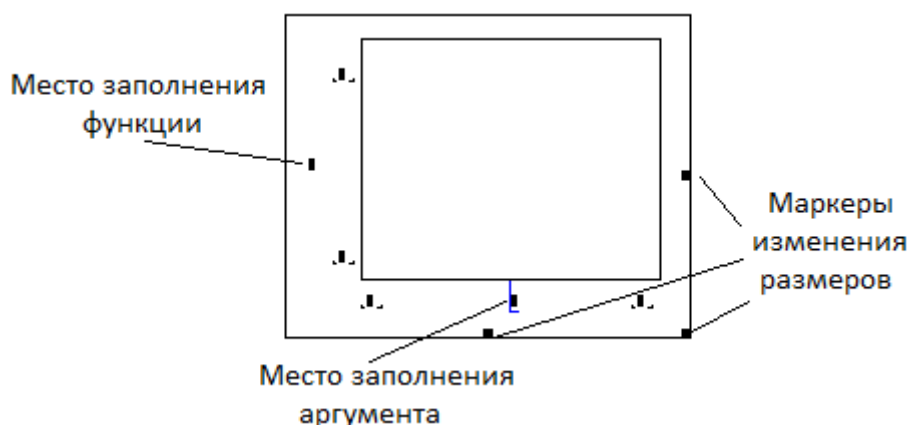


Рисунок 6.51 – Шаблон двумерного графика

– ввести в места заполнения аргумента и функции имя аргумента x и выражение, определяющее функцию $f(x)$, соответственно;

– кликнуть левой клавишей мыши вне диапазона.

По умолчанию в системе график строится в интервале $[-10, 10]$ или в той части этого интервала, в которой функция определена. Количество точек выбирается автоматически.

При необходимости можно изменить пределы варьирования аргумента. Для этого следует кликнуть мышью на значении границы интервала и набрать нужное число. В заключение нажать клавишу Enter или кликнуть мышью за пределами области графика. Аналогично изменяются пределы интервала по оси ординат.

На рисунке 6.52 представлен пример построения быстрого графика функции $\sin(x)$ с автоматически заданным интервалом изменения аргумента и с измененным.

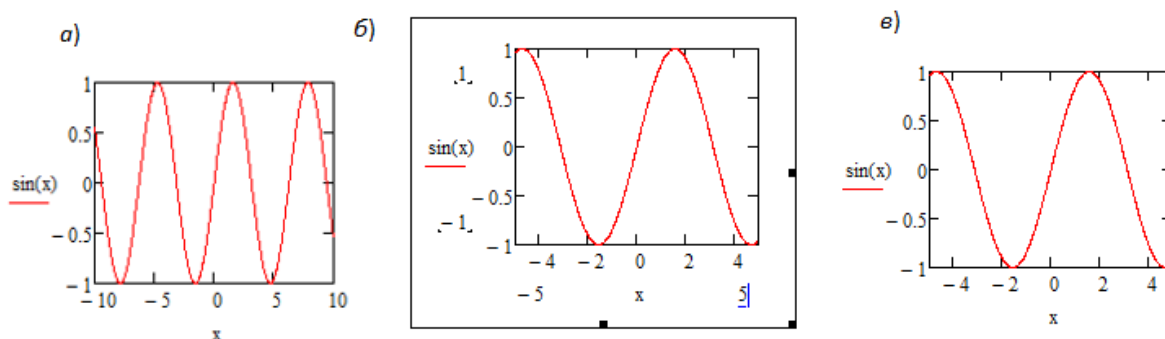


Рисунок 6.52 – Пример построения быстрого графика:

а – автоматически заданный интервал; б – изменение границ интервала; в – измененный интервал

б) построение графика *по точкам*

Этапы построения графика по точкам аналогичны этапам построения быстрого графика с той лишь разницей, что сначала нужно задать значения аргумента и определить функцию (задать уравнение).

На рисунке 6.53 показан график функции для конкретно заданных значений аргумента.

Для построения *графиков нескольких функций* от одной и той же переменной на одном шаблоне, необходимо в месте заполнения функции после имени или формулы первой функции поставить запятую (курсор должен переместиться на следующую строку) и ввести имя или формулу для другой функции (рис. 6.54).

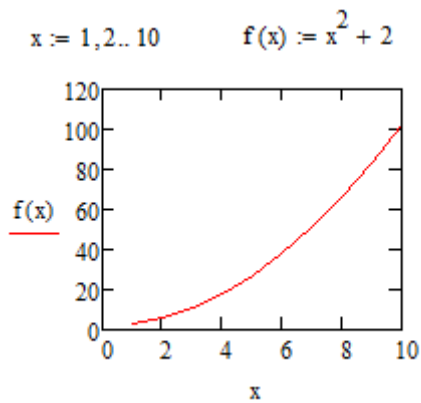


Рисунок 6.53 – Пример построения графика функции по точкам

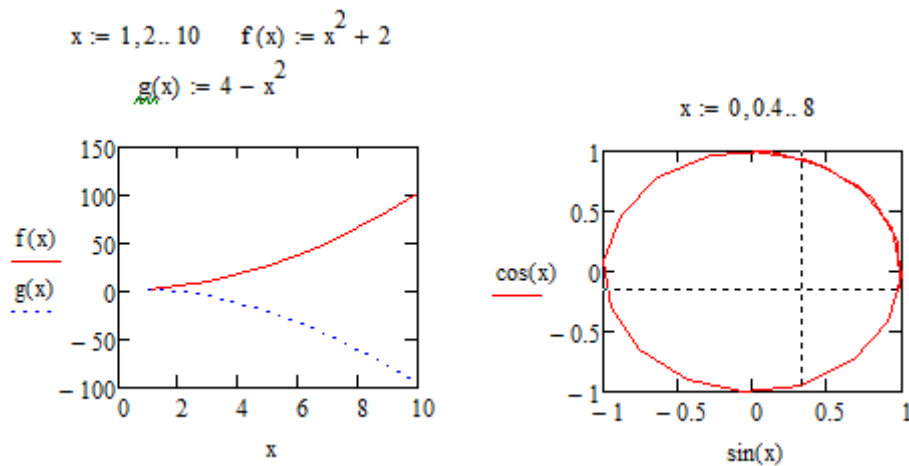



Рисунок 6.54 – Примеры построения графиков двух функций на одном шаблоне

Важно! При построении графика функции для заданного количества точек при большом шаге существует возможность пропустить некоторые особенности поведения функции (например, резкий скачок и сразу падение). Поэтому, прежде чем строить график, необходимо исследовать функцию аналитически или уменьшать шаг изменения аргумента.

Исследование графика функции в декартовых координатах

Для проведения анализа графика функции в системе MathCad доступны два режима: *Масштаб* и *Трассировка*. Они позволяют изменять размеры графика, увеличивать часть графика, получать координаты любой точки графика.

Для включения режима *Масштаб* служит одноименная кнопка  на палитре *График*.

Изменение размеров графика функции осуществляется стандартным образом: для этого щелчком левой клавиши мыши в

области графика необходимо выделить его и изменить размеры с помощью появившихся трех маркеров на рамке области графика.

Минимальное и максимальное значения аргумента задаются автоматически программой (от -10 до 10), эти значения для ординаты определяются по значению функции, а при построении графика по точкам – по исходным размерам.

Следует отметить, что использование инструмента *Масштаб* не увеличивает количества точек на графике.

Для изменения масштаба изображения части графика необходимо:

- поместить курсор в область графика, щелчком левой клавиши мыши выделить его;
- выбрать инструмент *Масштаб* на палитре *График*;
- установить курсор в одном из углов области, которую надо увеличить, и при нажатой левой клавиши мыши переместить курсор слева направо (график будет заключен в пунктирную рамку);
- в полях *Min* и *Max* палитры *График* отразятся координаты выделенной области;
- нажать на кнопку *OK* (рис. 6.55).

Кнопка *Отмена* отменяет одно увеличение.

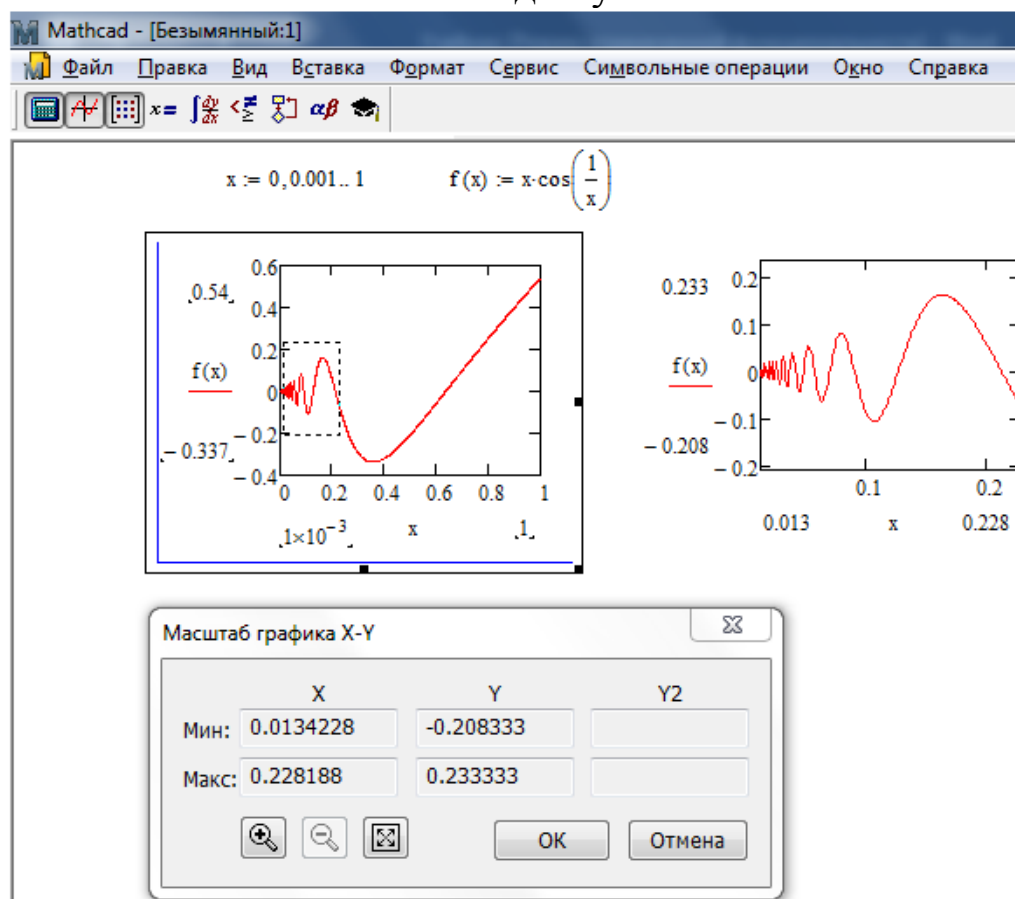


Рисунок 6.55 – Изменение масштаба части графика

Определение координат точек графика. Инструмент *Трассировка* палитры *График* служит для определения координат точек графика.

Чтобы увидеть координаты какой-либо точки кривой, необходимо:

- выделить график;
- активировать инструмент *Трассировка*, откроется диалоговое окно этого инструмента;
- перемещать мышь вдоль кривой при нажатой левой клавише, в окне инструмента будут отображаться координаты точек;
- кнопками *Копировать X* и *Копировать Y* сохранить координаты нужной точки (рис. 6.56).

Сохраненные координаты можно вставлять из буфера в любой документ (текст, электронные таблицы, документ MathCad и т.п.).

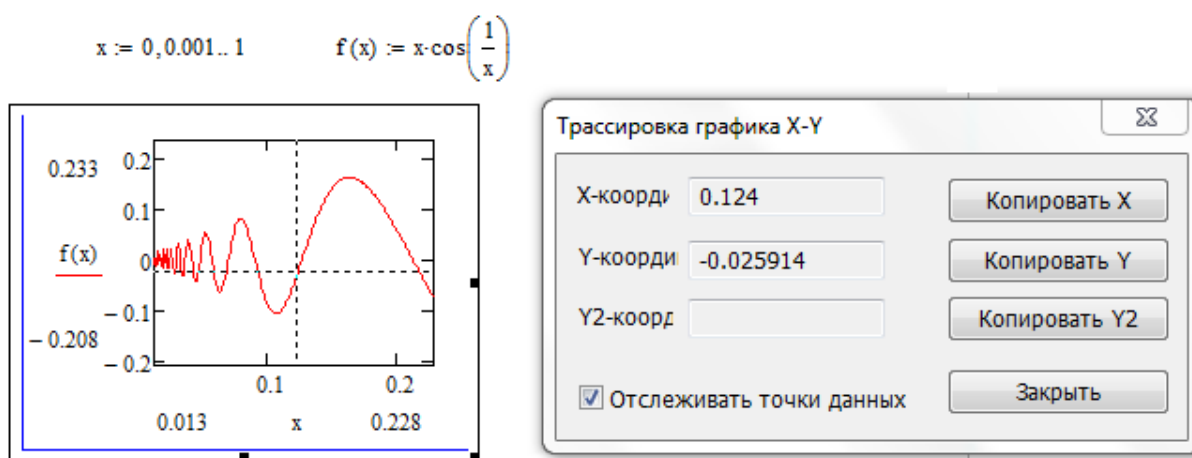


Рисунок 6.56 – Определение координат точек графика

Форматирование графика предполагает его визуальное оформление. Данный инструмент позволяет отобразить на графике дополнительные оси координат, задать цвета и надписи осей, изменить цвет кривой, выделить точки маркером и т.п. Для открытия окна инструмента *Форматирование* необходимо дважды кликнуть левой клавишей мыши на графике либо выбрать *Формат / График / График X-Y* в главном меню программы. Окно включает 5 вкладок (рис. 6.57).

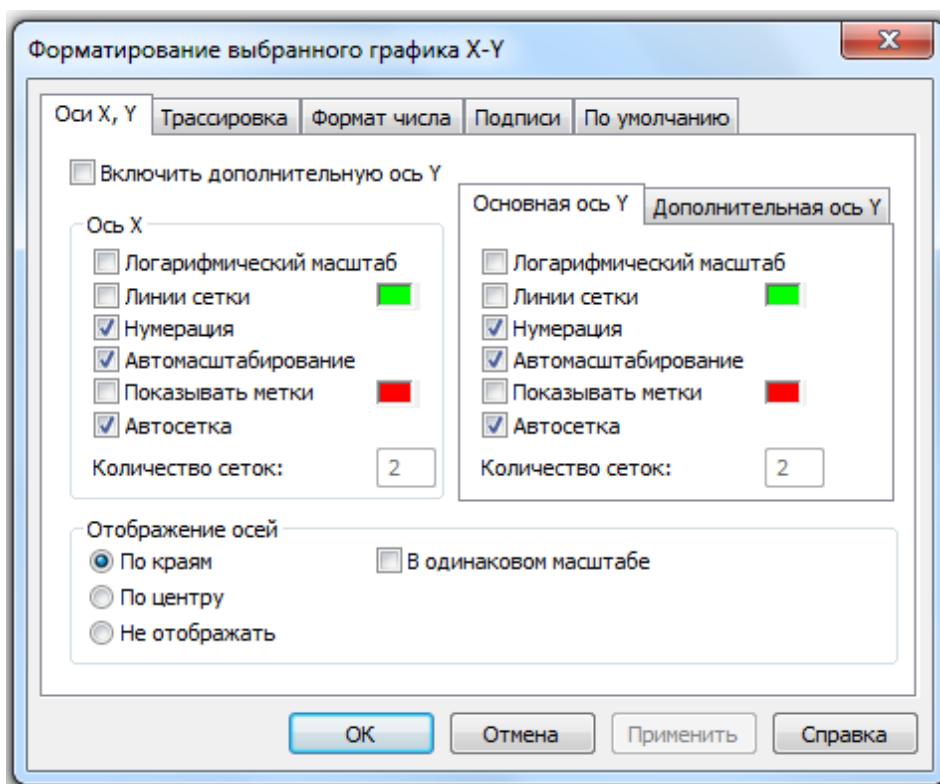


Рисунок 6.57 – Окно инструмента *Форматирование*

Кратко опишем некоторые из них.

Вкладка *Оси X, Y* позволяет выполнить настройку осей координатной плоскости. Как видно из рисунка 6.57, здесь включается и отключается отображение сетки на графике; задаются и отменяются дополнительные линии сетки; задается размещение осей на графике (по краям, по центру), можно отменить их отображение; изменяется масштаб отображения осей (можно задать логарифмический масштаб, который обычно используется в том случае, когда данные различаются на несколько порядков).

Вкладка *Трассировка* обеспечивает возможность настройки вида графики. К основным настройкам относятся:


- отображение маркеров в точках графика и задание их вида (звездочки, крестики и т.п.);
- тип представления данных (линия, точки, гистограмма и т.п.);
- вид линии (сплошная, пунктирная и т.п.);
- толщина линии;
- цвет линии.

Вкладка *Формат числа* используется для изменения формата числовых меток на графике.

Вкладка *Подписи* служит для задания подписей на графике и осях.

Вкладка *По умолчанию* позволяет сбросить заданные пользователем настройки и вернуться к стандартным.

Построение графиков в полярных координатах

Для построения графика функции в полярных координатах необходимо использовать одноименную кнопку  на палитре *График*. Так как в полярной системе координат каждая точка задается углом и радиус-вектором, в соответствующие местозаполнители нужно ввести имя углового аргумента α (нижний местозаполнитель) и выражение, которое определяет радиус-вектор $r(\alpha)$ (рис. 6.58).

Все правила построения и форматирования графика в декартовой системе координат справедливы и для полярных графиков.

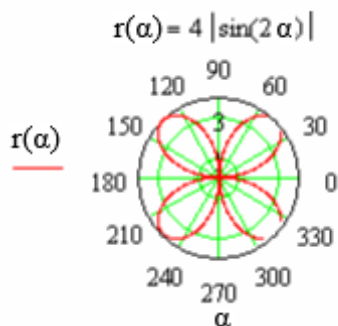







Рисунок 6.58 – Пример графика в полярных координатах

Следует отметить, что при построении полярного графика в качестве рядов данных могут быть взяты и отрицательные значения углов.

Построение трехмерных графиков

Трехмерные графики (3D графики) отображают функции двух переменных $f(x,y)$. Система MathCad допускает построение таких типов графиков, как:

- график поверхности (кнопка );
- линии уровня (кнопка );
- столбчатая 3D-диаграмма (кнопка );
- 3D-график разброса (кнопка );
- векторное поле (кнопка ).

Для построения трехмерного графика необходимо кликнуть на нужной кнопке палитры *График*. В месте расположения курсора появится область с тремя осями и одним местозаполнителем.

лем. Аналогично построению двумерных графиков трехмерный можно построить двумя способами: по заданной функции (быстрый график) либо с помощью матричной переменной, которая определяет расположение данных на плоскости. Поэтому в местозаполнитель следует ввести имя функции, либо имя матричной переменной.

Для изменения типа графика используется кнопка *Общее* окна форматирования графика.

Опишем основные действия для построения трехмерного графика двумя способами.

а) *быстрый график*

Последовательность действий заключается в следующем:

- задание функции двух переменных;
- установка курсора в место построения графика;
- выбор типа графика нажатием на соответствующей кнопке;
- введение имени функции в местозаполнитель;
- клик мышью вне области графика;

б) *построение с помощью матричной переменной*

Для построения графика по точкам необходимо

1. Задать данные, определяющие вектора значений x_i и y_i :
 - количество точек для расчета (n и m соответственно);
 - ранжированные значения i и j , которые определяют элементы векторов;
 - пределы изменения векторов x_i и $y_i(x_n, x_k, y_n, y_k)$.
2. Ввести выражения для вычисления индексных переменных x_i и y_i .
3. Ввести имя матрицы $M_{i,j}$ и выражение для вычисления элементов матрицы.
4. Указать курсором место построения графика.
5. Выбрать нужную кнопку типа графика на палитре *График*.
6. Ввести в местозаполнитель имя матрицы M .
7. Кликнуть мышью вне области графика.
8. При необходимости отформатировать график.

Следует отметить, что при этом за координаты x и y берутся номера строк и столбцов, а координата z соответствует значениям элементов матрицы. Кроме того, при построении графика данным способом шкала плоскости XU задается вручную, так как система Mathcad изображает только сами графики на основе двумерной структуры заданной матрицы.

Рассмотрим примеры построения различных типов графиков на примере графиков функции $f(x,y) = x^2 + y^2$ (быстрый график) и функции $g(x,y) = \sin(x)^2 + \cos(y)^2$ (на основе матрицы) (рис. 6.59).

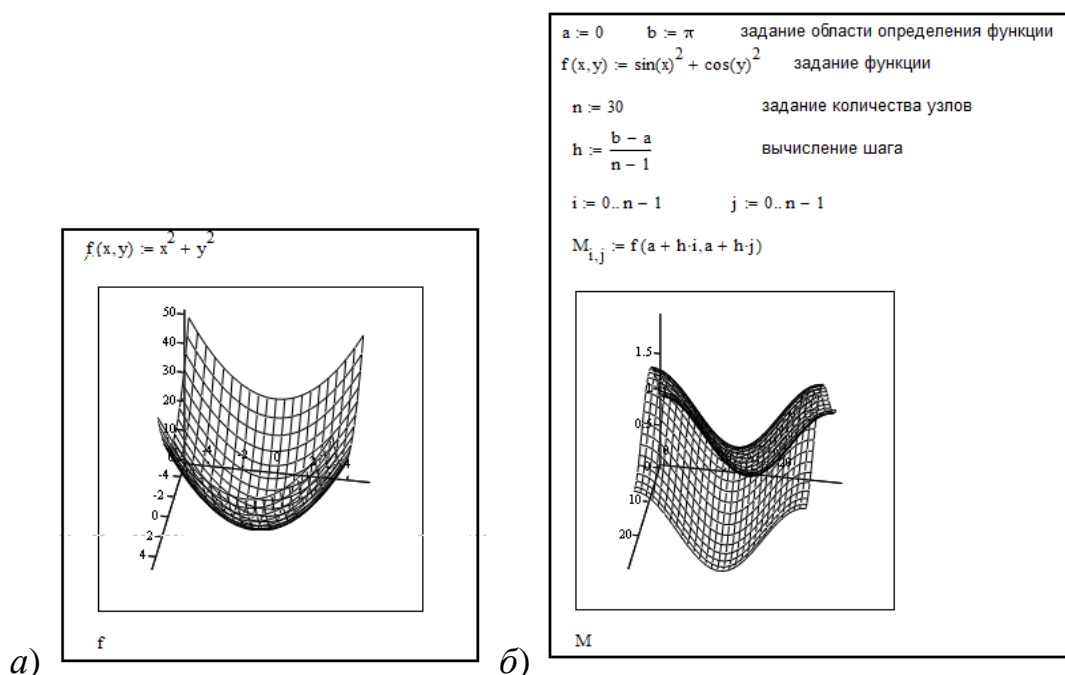


Рисунок 6.59 – Построение трехмерных графиков:
 а – быстрый график; б – график на основе матрицы

Решение уравнений и систем уравнений

Решение уравнений и систем уравнений представляет собой одну из наиболее важных задач инженерной практики. Для каждого вида уравнений система MathCAD предлагает различные методы решения. По умолчанию выбор метода решения осуществляется автоматически, при необходимости можно отключить режим автовыбора, предоставив выбор метода решения самому пользователю.

Численное решение алгебраических уравнений и систем
 Алгебраическим (полиномиальным) называется уравнение вида:

$$P(x_1, x_2, \dots, x_n) = 0,$$

где x_1, x_2, \dots, x_n – неизвестные переменные;

P – многочлен от переменных x_1, x_2, \dots, x_n .

Степенью алгебраического уравнения называют степень многочлена.

Пример алгебраического уравнения:

$$2y^2x^3 - 4x^2 + 3xy^2 - 6z + 7 = 0.$$

Данное уравнение представляет собой уравнение пятой степени от трех переменных.

Алгебраическое уравнение первой степени называют *линейным уравнением*.

Функция *root* предназначена для решения нелинейного алгебраического уравнения от одной переменной, точнее нахождения одного корня по одной переменной.

Функция записывается в виде:

$$\text{root}(f(x), x) \text{ или } \text{root}(f(x), x, x_{\min}, x_{\max}),$$

где $f(x)$ – функция, корни которой следует найти;

x – переменная, по которой вычисляется корень;

$[x_{\min}, x_{\max}]$ – интервал поиска.

Функция *root* ($f(x)$, x) определяет корень уравнения по заданному начальному приближению с заданной точностью *TOL* на основе итерационного метода секущих. Корнем считается ближайшее к начальному приближению значение x , при котором функция $f(x)$ обращается в 0. Если корней несколько, необходимо задать свое начальное приближение для каждого корня.

Переменная *TOL* по умолчанию имеет значение 0.001. При увеличении этого значения повышается точность нахождения решения, но и уменьшается вероятность нахождения точного решения. Она используется при вычислении производных, интегралов и нахождении корней полиномов.

Как правило, начальное приближение ищется с помощью построения графика этой функции. На графике следует найти точки, достаточно близкие к тем, в которых функция обращается в ноль. При необходимости можно использовать инструменты *Масштаб* и *Трассировка*, чтобы увеличить интервал и определить координаты точек.

На рисунке 6.60 приведен пример решения уравнения:

$$f(x) = 2^x - \frac{x^3}{x^2 + 1} - 5$$

с помощью функции *root* ($f(x)$, x).

Функция *root* ($f(x)$, x , x_{\min} , x_{\max}) определяет корень уравнения в указанном интервале. При этом в заданном интервале находится один корень и значения x_{\min} , x_{\max} должны иметь разные знаки.

Функция *polyroots* (v) применяется для нахождения корней полинома n -й степени. Вектор v состоит из коэффициентов поли-

нома, начиная со свободного члена. Найденное решение представляется в виде вектора, содержащего n корней полинома.

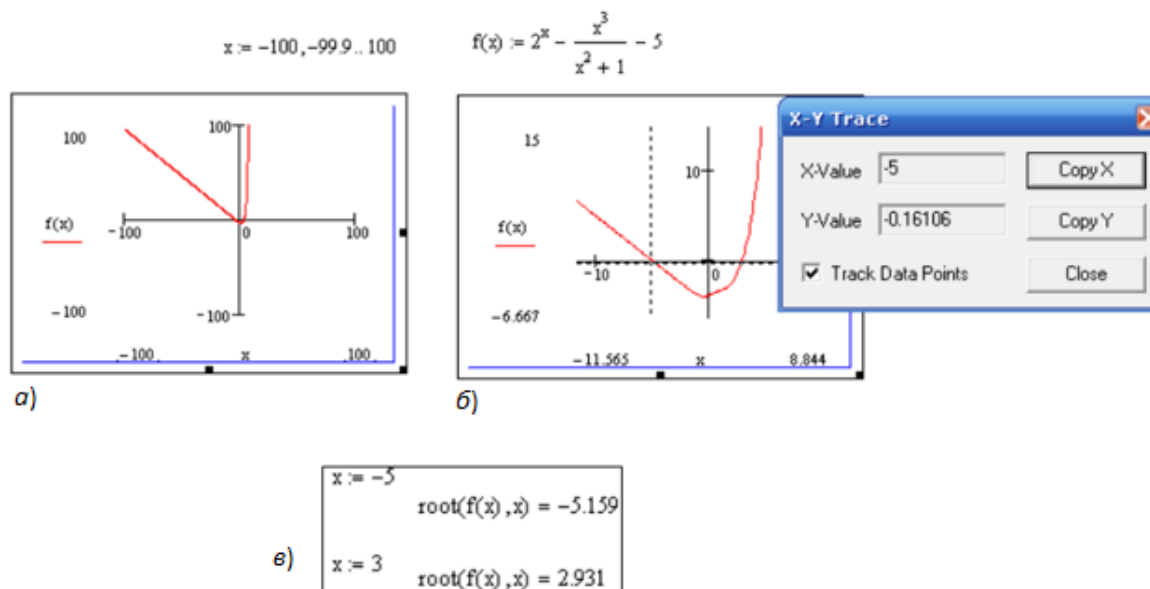


Рисунок 6.60 – Нахождение корней уравнения с помощью функции $\text{root}(f(x), x)$:
 а – график функции; б – увеличенный интервал поиска корней;
 в – найденные корни уравнения

Пример 1. Найдите корни полинома:

$$x^4 - 3x^3 - 8x^2 + 2 = 0.$$

На рисунке 6.61 приведено решение данного примера.

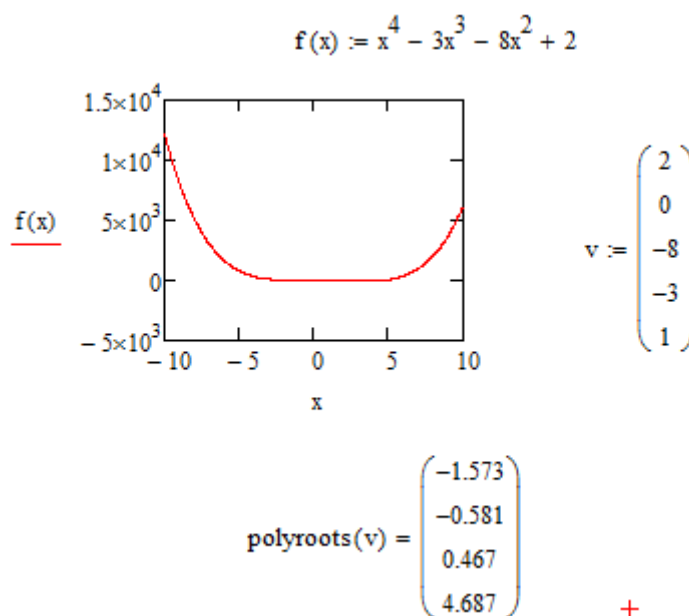


Рисунок 6.61 – Нахождение корней полинома

Функция $Find(x_1, \dots, x_n)$ используется для решения систем алгебраических уравнений (как линейных, так и нелинейных) по заданному начальному приближению неизвестных.

Вычислительный блок $Given...Find$ для решения системы уравнения в общем виде выглядит так:

- задание начальных приближений для переменных;
- Given;
- уравнения и/или неравенства;
- $Find$ (список переменных через запятую) или вектор := $Find$ (список переменных через запятую).

Функция $Find$ вычисляет точное решение уравнения или системы уравнений.

Вместо этой функции могут использоваться:

$minerr$ – для нахождения приближенного решения уравнений и систем;

$Maximize / Minimize$ – для решения задач оптимизации.

На рисунке 6.62 приведен пример решения системы уравнений с помощью блока $Given...Find$.

$x := 1 \quad y := 2 \quad z := 3$ Given $5 \cdot x + 5 \cdot y = 4$ $2 \cdot x + 3 \cdot y + z = 6$ $x + 5 \cdot y + 7 \cdot z = 0$ $M := Find(x, y, z)$ $M = \begin{pmatrix} -9.733 \\ 10.533 \\ -6.133 \end{pmatrix}$	задание начального приближения задание системы уравнений знак = ставится сочетанием Ctrl - =
$M_0 = -9.733 \quad M_1 = 10.533 \quad M_2 = -6.133$	

Рисунок 6.62 – Решение системы уравнений с помощью блока $Given...Find$

Функция $Isolve(A, B)$ применяется для решения системы линейных алгебраических уравнений, записанной в матричном виде:

$$A \times X = B,$$

где A – матрица, составленная из коэффициентов системы;

B – вектор-столбец правых частей;

X – вектор-столбец неизвестных.

Решение находится с помощью обратной к заданной матрицы A^{-1} :

$$X = A^{-1} \times B.$$

Решить систему алгебраических уравнений можно как с помощью функции $lsolve(A,B)$, так и используя обратную матрицу либо блок $Given...Find$ (рис. 6.63). Однако некоторые системы уравнений, которые не удастся решить, применяя блок $Given...Find$, можно решить с помощью данной функции.

задание коэффициентов матрицы	задание вектора правых частей	
$A := \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$	$B := \begin{pmatrix} 3 \\ 6 \\ 4 \\ 7 \end{pmatrix}$	
решение с помощью функции $lsolve(A,B)$	решение с помощью обратной матрицы	решение с помощью блока $Given-Find$
$X := lsolve(A,B)$	$X := A^{-1} \cdot B$	$X := \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ <p>задание начальных приближений</p>
$X = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 2 \end{pmatrix}$	$X = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 2 \end{pmatrix}$	<p>Given $A \cdot X = B$ $X := Find(X)$</p> $X = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 2 \end{pmatrix}$

Рисунок 6.63 – Решение системы линейных алгебраических уравнений тремя методами

Функции $Maximize(f, x)$, $Minimize(f, x)$ используются для решения задач оптимизации. Как правило, суть таких задач заключается в нахождении максимального или минимального значения некоторого параметра.

Обозначения:

f – функция, экстремум (минимум или максимум) которой следует найти, при этом ее уравнение должно быть задано заранее;

x – переменная, по которой необходимо найти оптимальное значение, для нее нужно задать начальное приближение.

Пример 2. Найти наименьшее значение функции

$$f(x) = 3\cos(2x) + \sin(x).$$

Решение приведено на рисунке 6.64.

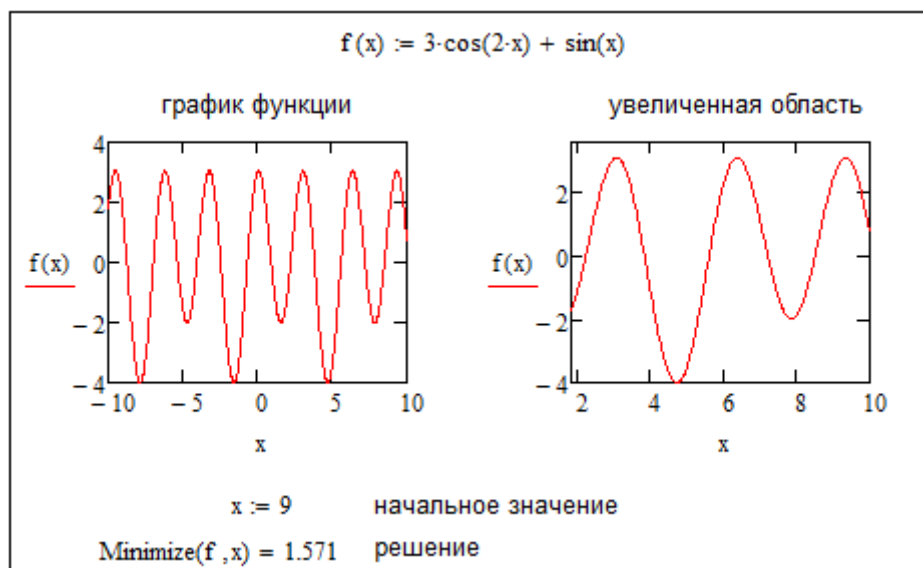


Рисунок 6.64 – Нахождение наименьшего значения функции

Следует помнить, что найденное значение является точкой локального минимума. Для того, чтобы найти точку глобального минимума (или максимума), необходимо задать несколько разных начальных значений.

Если функция зависит от нескольких переменных, например от двух $f(x, y)$, функции $\text{Maximize}(f, x)$, $\text{Minimize}(f, x)$ записываются в виде:

$\text{Maximize}(f, x, y)$, $\text{Minimize}(f, x, y)$.

Пример 3. Найти максимум функции

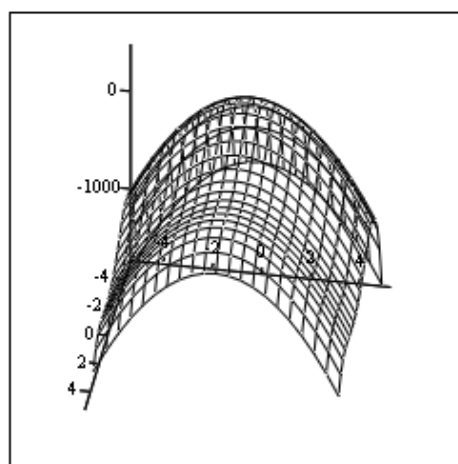
$$f(x, y) = 100 - 2x^4 + 6x^3 + 30x^2 - 120x - 45y^2,$$

где $x \in [-5, 5]$, $y \in [-5, 5]$.

Решение приведено на рисунке 6.65.

$$g(x, y) := 100 - 2x^4 + 6x^3 + 30x^2 - 120x - 45y^2 \quad \text{задание функции}$$

$$x := -5 \quad y := -5 \quad \text{задание начальных значений}$$



$$\begin{pmatrix} x_{\text{opt}} \\ y_{\text{opt}} \end{pmatrix} := \text{Maximize}(g, x, y)$$


$$x_{\text{opt}} = -2.668$$

$$y_{\text{opt}} = -2.097 \times 10^{-9}$$

g

Рисунок 6.65 – Пример нахождения максимального значения функции

Дифференцирование и интегрирование

Операции дифференцирования и интегрирования в системе Mathcad реализуются с помощью инструментов палитры *Математический анализ*  панели *Математика*.

Для их проведения следует:

- кликнуть мышью на значке с нужной операцией;
- заполнить появившийся шаблон с местозаполнителями;
- кликнуть на значке численного вывода (знак « = »).

Производные функций от 0-го до 5-го порядка вычисляются с помощью вычислительного процессора, а выше 5-го порядка – символьного. Для этого вместо знака « = » используется символ « \rightarrow ».

Вычисление частных производных производится аналогично.

Результатом интегрирования является приближенное значение, точность которого задается переменной *TOL*.

Следует учесть, что неопределенные интегралы, а также интегралы, зависящие от параметра, и расходящиеся интегралы можно вычислить только в символьном виде.

Определенный интеграл вычисляется либо в численном, либо в символьном виде.

Система Mathcad позволяет выбрать один из четырех методов интегрирования (рис. 6.66):

- 1) метод Ромберга, который обычно используют при интегрировании функций, не имеющих особенности;
- 2) адаптивный метод применяют для быстроменяющихся функций;
- 3) предел в бесконечности – для функций с бесконечными пределами;
- 4) особая конечная точка – для функций, у которых один из пределов не определен.

Удобнее всего для вычисления интегралов использовать режим *Автовыбор* (*AutoSelect*), позволяющий программе автоматически выбрать наиболее подходящий метод интегрирования.

На рисунке 6.67 приведены примеры интегрирования функций.

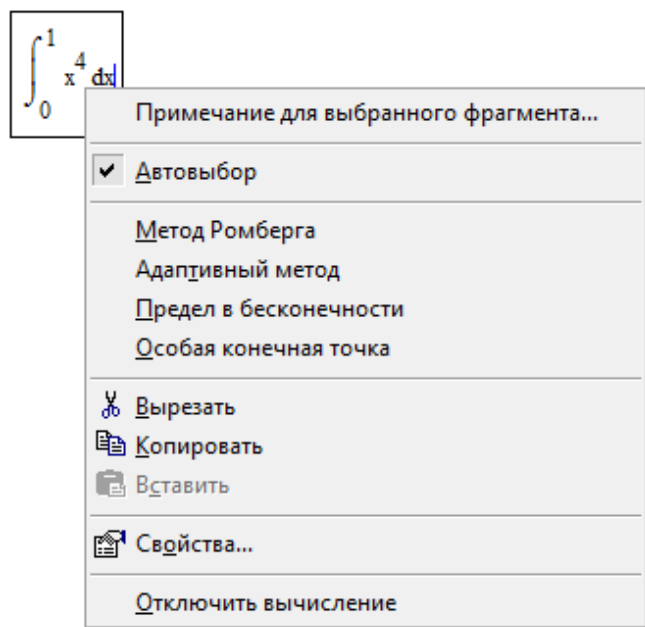


Рисунок 6.66 – Выбор метода интегрирования

Интегрирование	
Численное вычисление	Символьное вычисление
$\int \frac{x-2}{x^3} dx = \blacksquare$ <p style="text-align: center; background-color: yellow;">This variable is undefined.</p>	$\int \frac{x-2}{x^3} dx \rightarrow \frac{(x-2)^2}{4x^2}$
$\int_0^1 \frac{x^2}{(x+1)^4} dx = 0.042$	$\int_0^1 \frac{x^2}{(x+1)^4} dx \rightarrow \frac{1}{24}$
$\int_0^2 \int_0^1 x^2 y dx dy = 0.667$	$\int_0^2 \int_0^1 x^2 y dx dy \rightarrow \frac{2}{3}$
расходящийся интеграл	
$\int_0^{\infty} \frac{1}{\sqrt{x}} dx = \blacksquare$ <p style="text-align: center; background-color: yellow;">This calculation does not converge to a solution.</p>	$\int_0^{\infty} \frac{1}{\sqrt{x}} dx \rightarrow \infty$

Рисунок 6.67 – Примеры интегрирования функций

На рисунке 6.68 приведены примеры дифференцирования функций.

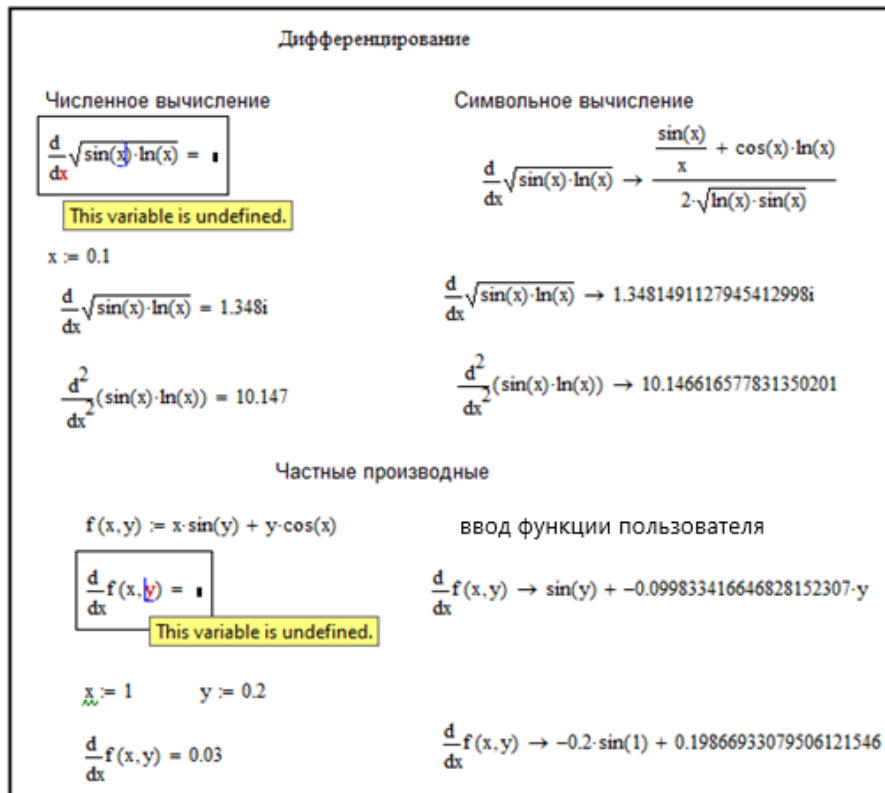


Рисунок 6.68 – Примеры дифференцирования функций

Решение дифференциальных и разностных уравнений

Определения. Уравнение, связывающее функцию y , ее производные и аргумент x , называется *обыкновенным дифференциальным уравнением*.

Обыкновенным дифференциальным уравнением называется уравнение, в котором функция y зависит только от одной переменной, при этом и функция и ее производные входят в уравнение как значения в одной и той же переменной точке.

Обыкновенное дифференциальное уравнение в общем виде можно записать следующим образом:

$$F(x, y, y', y'', \dots, y^{(n)}) = 0$$

или

$$F\left(x, y, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, \dots, \frac{\partial^n y}{\partial x^n}\right) = 0.$$

Порядком дифференциального уравнения называется наивысший порядок производной, входящей в уравнение.

Например:

а) $y' + 2xy^2 + 5 = 0$ является дифференциальным уравнением 1-го порядка;

б) $y'' + ky' - by - \sin x = 0$ является дифференциальным уравнением 2-го порядка;

в) $y^{(n)} - g(x, y^{(n-1)}) = 0$ является дифференциальным уравнением n -го порядка.

Решением дифференциального уравнения называется всякая функция $y = f(x)$, которая, будучи подставленной в уравнение, обращает его в тождество.

Задача нахождения решения дифференциального уравнения $y' = f(x, y)$, удовлетворяющего начальному условию $y(x_0) = y_0$, называется *задачей Коши*.

Математические методы решения

Уравнения с разделяющимися переменными. Дифференциальное уравнение первого порядка $F(x, y, y') = 0$ называется *уравнением с разделяющимися переменными*, если оно может быть представлено в виде

$$y' = g(x)h(y) \text{ или } \frac{\partial y}{\partial x} = g(x)h(y).$$

Разделяя переменные x и y и их дифференциалы в разные стороны такого уравнения $\frac{1}{h(y)} \partial y = g(x) \partial x$ и интегрируя каждую часть полученного уравнения, находим решение.

Однородные уравнения. Функция $f(x, y)$ называется *однородной функцией* степени m , если $f(\lambda x, \lambda y) = \lambda^m f(x, y)$.

Функция $f(x, y)$ называется *однородной нулевой степени*, если $f(\lambda x, \lambda y) = \lambda^0 f(x, y) = f(x, y)$.

Дифференциальное уравнение первого порядка $F(x, y, y') = 0$ называется *однородным*, если оно может быть представлено в виде

$$y' = f(x, y) \text{ или } \frac{\partial y}{\partial x} = f(x, y),$$

где $f(x, y)$ – однородная функция нулевой степени.

Решение однородного дифференциального уравнения сводится к решению уравнения с разделяющимися переменными заменой: $y = ux$, где u – функция от x .

Линейным дифференциальным уравнением первого порядка называется уравнение первого порядка, которое линейно относительно неизвестной функции y и ее производной

$$y' + g(x)y = h(x).$$

Общее решение данного уравнения представляется в виде: $y = u(x)v(x)$, где $v(x)$ – частное решение однородного урав-

нения $y' + g(x)y = 0$, при $c=1$, $u(x, c)$ – общее решение уравнения $u'v(x) = h(x)$.

Линейным однородным дифференциальным уравнением второго порядка называется уравнение вида

$$y'' + ay' + by = 0,$$

где a и b – действительные числа.

Общее решение линейного однородного дифференциального уравнения второго порядка с постоянными коэффициентами имеет вид $y_0 = C_1y_1 + C_2y_2$, где y_1 и y_2 – частные линейно независимые решения, а C_1 и C_2 – произвольные постоянные.

Функция $y = e^{\lambda x}$, где λ – действительное или комплексное число, является частным решением данного уравнения, если λ – корень уравнения $\lambda^2 + a\lambda + b = 0$. Данное уравнение называется *характеристическим уравнением*.

Разностное уравнение. Рассмотрим дифференциальное уравнение первого порядка $y'(x) + A \cdot y(x) = 0$, удовлетворяющее начальному условию $y(0) = 1$.

Разобьем ось Ox точками на равные отрезки, отстоящие друг от друга на величину $h > 0$. Вместо функции $y(x)$ будем искать таблицу ее значений $y(0), y(h), y(2h), \dots, y(n \cdot h), \dots$.

В соответствии с определением производной ее можно заменить отношением

$$\frac{y(x+h) - u(x)}{h}.$$

При достаточно малом шаге h после такой замены вместо дифференциального уравнения мы получим приближающее его разностное уравнение

$$\frac{y(x+h) - u(x)}{h} + A \cdot y(x) = 0.$$

Производную можно заменить и другим разностным отношением

$$\frac{y(x+h) - u(x-h)}{2h},$$

Тогда дифференциальное уравнение будет заменено разностным такого вида:

$$\frac{y(x+h) - u(x-h)}{2h} + A \cdot y(x) = 0.$$

В дифференциальном уравнении второго порядка

$$y''(x) + A \cdot y'(x) + B \cdot y(x) = f(x)$$

$y'(x)$ можно заменить одним из приведенных выше способов, а $y''(x)$, например, таким разностным отношением:

$$\frac{\frac{y(x+h)-y(x)}{h} - \frac{y(x)-y(x-h)}{h}}{h} = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2}.$$

В результате заданное дифференциальное уравнение будет заменено разностным

$$\frac{y(x+h) - 2y(x) + y(x-h)}{h^2} + A \cdot \frac{y(x+h) - y(x-h)}{2h} + B \cdot y(x) = f(x).$$

Разностным уравнением называется уравнение, которое связывает значение некоторой неизвестной функции в любой точке с ее значением в одной или нескольких точках, отстоящих от данной, на определенный интервал $f(y_h, y_{h+1}, \dots, y_{h+n}, y) = 0$.

Разностные уравнения используют для получения приближенного решения обыкновенного дифференциального уравнения, когда получить точное решение не удастся.

Численные методы решения обыкновенных дифференциальных уравнений

Получить аналитически решения дифференциальных уравнения, особенно нелинейных, удастся очень редко. Поэтому на практике чаще всего прибегают к численным методам. Кратко опишем некоторые из них.

Метод Эйлера – это самый простой метод решения задачи Коши.

Рассмотрим дифференциальное уравнение первого порядка: $y' = f(x(t), y(t), t)$, с начальными условиями $y(t_0) = y_0$, $x(t_0) = x_0$.

По определению производной $y' \cong tg(\alpha) = \frac{\Delta y}{\Delta t}$.

Зная значение функции в начальной точке, ее значение в последующих точках можно вычислить по формуле

$$\begin{aligned} y(t + \Delta t) &\approx y(t) + \Delta y = y(t) + tg(\alpha) \cdot \Delta t = y(t) + \frac{\Delta y}{\Delta t} \cdot \Delta t \\ &= y(t) + f(x, y, t) \cdot \Delta t \end{aligned}$$

$$\text{или } y(t + \Delta t) \approx y(t) + f(x, y, t) \cdot \Delta t.$$

Эта формула называется *формулой Эйлера*.

Метод Эйлера достаточно легко реализовать. Однако точность решения, полученного этим методом, сильно зависит от величины шага интегрирования Δt . Замена дифференциального уравнения разностным приводит к увеличению погрешности.

Метод Рунге-Кутты представляет собой численный метод решения задачи Коши для обыкновенного дифференциального уравнения или системы уравнений. Его суть заключается в пошаговом вычислении решения с заданными начальными условиями. Классический метод Рунге-Кутты имеет четвертый порядок точности.

Рассмотрим дифференциальное уравнение первого порядка:

$$\frac{dy}{dx} = f(x, y(x)), \quad 0 < x < N, \quad y(0) = y_0.$$

Разложим функцию в ряд Тейлора:

$$y(x_{k+1}) = y(x_k) + y'(x_k)h + y''(x_k)\frac{h^2}{2} + \dots$$

Вторую производную заменим разностным отношением:

$$y''(x_k) = (y'(x_k))' = f'(x_k, y(x_k)) \approx \frac{f(\tilde{x}, \tilde{y}) - f(x_k, y(x_k))}{\Delta x},$$

где $\tilde{x} = x_k + \Delta x$, $\tilde{y} = y(x_k + \Delta x)$.

Теперь заменяя \tilde{y} разложением в ряд Тейлора, получаем расчетные формулы для решения дифференциального уравнения:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = f(x_n, y_n),$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right),$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right),$$

$$k_4 = f(x_n + h, y_n + hk_3).$$

Метод Булирша-Штера предназначен для решения обыкновенных дифференциальных уравнений (или систем уравнений) первого порядка, у которых правые части представляют собой гладкие функции.

Гладкой называют функцию, у которой все частные производные непрерывны в области определения функции.

В этом случае применение метода Рунге-Кутты дает менее точное решение. Отличие метода Булирша-Штера заключается в применении рациональной экстраполяции вместо полиномиальной. *Экстраполяцией* называют особый тип аппроксимации (приближения), при котором значение переменной оценивается вне интервала изменения, а не внутри него.

Основой экстраполяции Булирша-Штера является симметричный многошаговый метод прямоугольников:

$$y_{n+2} = y_n + 2hf(x_{n+1}, y_{n+1}).$$

При этом, чтобы сохранить симметрию, в качестве приближения к $y(x)$ следует брать, например, $y(X, h) = y_N$.

Основная идея метода состоит в том, что состояние системы в точке $x + h$ вычисляется как результат двух шагов длиной $h/2$, четырех шагов длиной $h/4$, восьми шагов длиной $h/8$ и так далее с последующей экстраполяцией результатов. Таким образом строится рациональная интерполирующая функция, которая в точке $h/2$ проходит через состояние системы после двух таких шагов, в точке $h/4$ проходит через состояние системы после четырех таких шагов, и т.д., а затем вычисляет значение этой функции в точке $h = 0$, проводя экстраполяцию.

Вопросы и задания для самоконтроля

1. Опишите основные возможности математического пакета программ MathCAD.

2. Что размещается в областях документа MathCAD?

3. Какие палитры содержит панель инструментов *Математика*?

4. Определите, какая палитра панели инструментов *Математика* содержит наибольшее число кнопок инструментов.

5. Какие виды курсора в зависимости от ввода типа данных существуют?

6. Опишите основные компоненты математических выражений программы MathCad – операторы, константы, переменные, массивы, функции.

7. Используя встроенные функции палитры инструментов *Калькулятор*, введите и вычислите выражения:

А) $\frac{8\sqrt{\pi+2}}{\sin\frac{2\pi}{6}}$,

Б) $e^{\sqrt[3]{38^2-101}} \cdot 200^{\frac{6}{7}}$,

В) $\log_{0,2}(e - 3) - \cos\frac{2\pi}{9} \geq 0$.

8. Задайте переменным x , y , z значения приведенных в пункте 7 арифметических выражений с помощью оператора присваивания (т.е. первое выражение определяет переменную x , второе – переменную y , третье – переменную z).

9. Найдите определитель и обратную матрицу матрицы A

$$A = \begin{pmatrix} -1 & -6 & 3 \\ 0 & 4 & 12 \\ 5 & -11 & 7 \end{pmatrix}$$

10. Опишите процедуру построения двумерных и трехмерных графиков.

11. Задайте дискретную переменную x , меняющуюся от 0 до $3\pi/2$ с шагом 0.05, и определите функцию $y = x^2 \sin x$. Постройте ее график.

12. На одном рисунке постройте графики функций: $y_1 = \frac{1}{3} \sin x$, $y_1 = e^{x^2}$ для $x \in [0.2, 6]$ с шагом 0.01. Используя вкладку окна форматирования Trace (Трассировка), измените стиль отображения и цвет кривых.

13. Постройте графики функций в полярной системе координат на разных рисунках (угол φ меняется от 0 до π): $\rho(\varphi) = \sin \varphi + \cos \varphi$, $\rho(\varphi) = \varphi$. Выполните форматирование графиков.

14. Какие возможности предоставляет MathCAD для решения уравнений и систем уравнений?

15. Как называется палитра, с помощью инструментов которой выполняются операции дифференцирования и интегрирования в системе Mathcad?

16. Вычислите значения следующих выражений с помощью специальных операторов в системе Mathcad:

А) $\int_0^3 \frac{x^2}{x-1} dx,$

Б) $\int_{-1}^2 \int_5^7 (6x^4 - 5x^5) dx,$

В) $\sum_{x=1}^{15} e^{\sqrt{x}}.$

17. Дайте определение дифференциального уравнения. Укажите виды дифференциальных уравнений.

ЗАКЛЮЧЕНИЕ

Современный этап развития общества характеризуется широким внедрением компьютерной техники, новых информационных технологий, телекоммуникаций, различных видов документальной связи во все сферы жизнедеятельности человека. Востребованность будущего инженера сегодня напрямую зависит от того, насколько учебный процесс ориентирован на будущую профессиональную деятельность, которая связана с решением разнообразных задач обработки, передачи, трансформации протекающих информационных потоков и процессов.

В основе решения профессиональных задач лежит построение различных алгоритмов, их анализ, оценка и выбор наиболее эффективных вариантов решения. Это обуславливает необходимость формирования компетенций бакалавров, специалистов и магистрантов в области алгоритмизации, моделирования и программирования, как для образовательной деятельности, так и для подготовки конкурентоспособных специалистов.

Изучение предложенных в учебном пособии тем дает возможность не только познакомиться с сущностью понятия «информация», архитектурой персонального компьютера, основами программирования, но и обеспечивает развитие алгоритмического стиля мышления, необходимого для успешного обучения в вузе и последующей работы в профессиональной области.

ЛИТЕРАТУРА

1. Беляева, Л.Р. Основы работы в MathCAD: метод. указания к практическим занятиям / Л.Р. Беляева, Р.С. Зарипова, Р.А. Ишмуратов. – Казань: Казан.гос. энерг. ун-т, 2012. – 60 с.
2. Берман, Н.Д. Основы информатики: учеб. пособие / Н.Д. Берман, В.В. Стригунов, Н.И. Шадрина; науч. ред. Э.М. Вихтенко. – Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2014. – 76 с.
3. Виленкин, Н.Я. Дифференциальные уравнения: учеб. пособие / Н.Я. Виленкин, М.А. Доброхотова, А.Н. Сафонов. – Москва: Просвещение, 1984. – 176 с.
4. Графики в Mathcad: учеб.-метод. пособие / сост. С.В. Киреев, П.А. Вельмисов. – Ульяновск: УлГТУ, 2018. – 84 с.
5. Гредасова Н.В. Обыкновенные дифференциальные уравнения: учеб. пособие / Н.В. Гредасова, И.Ю. Андреева; М-во науки и высшего образования РФ, Урал. федер. ун-т им. первого Президента России Б.Н. Ельцина. – Екатеринбург: Изд-во Урал. ун-та, 2022. – 88 с.
6. Гунько, В.Д. Дифференциальные уравнения. Примеры и типовые задания: учеб. пособие/ В.Д. Гунько, Л.Ю. Суховеева, В.М. Смоленцев; КубГАУ. – Краснодар, 2005. – 105 с.
7. Гурьяшова, Р.Н. Информатика. Пакет Mathcad: учеб. пособие / Р.Н. Гурьяшова, А.В. Шеянов. – Нижний Новгород: Изд-во ФГОУ ВПО ВГАВТ, 2005. – 140 с.
8. Закляков, В.Ф. Информатика: учеб. для вузов. – 5-е изд., перераб. и доп. – Москва: ДМК Пресс, 2021. – 750 с.
9. Информатика. Базовый курс / под ред. С.В. Симоновича. – 2-е изд. – Санкт-Петербург: Питер, 2005. – 640 с.
10. Лекция. Что такое информатика [Электронный ресурс]. – URL: <https://textarchive.ru/c-2010773-p22.html>.
11. Новожилов, О.П. Архитектура ЭВМ и систем: учеб. пособие: в 2 ч. / О.П. Новожилов. – Москва: Юрайт, 2022. – Ч. 1. – 276 с.
12. Новожилов, О.П. Архитектура ЭВМ и систем: учеб. пособие: в 2 ч. / О.П. Новожилов. – Москва: Юрайт, 2022. – Ч. 2. – 246 с. – URL: <https://urait.ru/bcode/494315>.
13. Основы вычислений и программирования в пакете MathCAD: учеб. пособие / Ю.Е. Воскобойников [и др.]; под ред. Ю.Е. Воскобойникова. – Новосибирск: НГАСУ (Сибстрин), 2012. – 212 с.

14. Петраш, В.И. Вычислительная система MathCAD\; учеб. пособие / В.И. Петраш. – Санкт-Петербург: Санкт-Петерб. политехн. ун-т Петра Великого, 2017. – Ч. 2. – 23 с.

15. Попова, О.В. Информатика: учеб. пособие / О.В. Попова. – Красноярск: Краснояр. ин-т экономики Санкт-Петерб. акад. управления и экономики (НОУ ВПО), 2007. – 186 с. – URL: <http://www.klyaksa.net/htm/kopilka/uchp/index.htm>.

16. Пушкарева, Т.П. Основы компьютерной обработки информации: учеб. пособие / Т.П. Пушкарева. – Красноярск: СФУ, 2016. – 180 с. – ISBN 978-5-7638-3492-5 // ЭБС «Консультант студента»: сайт. – URL: <https://www.studentlibrary.ru/book/ISBN9785763834925.html>.

17. Самарский, А.А. Математическое моделирование: Идеи. Методы. Примеры / А.А. Самарский, А.П. Михайлов. – 2-е изд., испр. – Москва: Физматлит, 2001. – 320 с.

18. Системный анализ и принятия решений. Технология вычислений в системе компьютерной математики Mathcad: учеб. пособие / В.А. Холоднов, В.П. Дьяконов, В.В. Фонарь [и др.]. – Санкт-Петербург: СПбГТИ (ТУ), 2013. – 154 с.

19. Титовская, Н.В. Введение в информатику (базовый курс): учеб. пособие / Н.В. Титовская, С.Н. Титовский, И.И. Болдарук, Н.Д. Амбросенко; Краснояр. гос. аграр. ун-т. – Красноярск, 2022. – 315 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1 ДЕЙСТВИЯ НАД ИНФОРМАЦИЕЙ	4
1.1 Понятие информации.....	4
1.2 Операции с данными.....	8
1.3 Измерение количества информации	10
Вопросы и задания для самоконтроля	17
2 АРХИТЕКТУРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА	18
2.1 Магистрально-модульный принцип построения компьютера	18
2.2 Основные устройства персонального компьютера.....	20
2.3 Периферийные устройства персонального компьютера.....	33
Вопросы для самоконтроля	38
3 АРИФМЕТИЧЕСКИЕ ОСНОВЫ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА ...	40
3.1 Системы счисления	40
3.2 Перевод чисел из одной системы счисления в другую	43
3.3 Выполнение арифметических операций в позиционных системах счисления	45
Вопросы и задания для самоконтроля	52
4 ЛОГИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРА	53
4.1 Логика высказываний	53
4.2 Методы решения логических задач.....	57
4.3 Логические основы персонального компьютера	66
Вопросы и задания для самоконтроля	74
5 ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ.....	79
5.1 Представление текстовой информации	81
5.2 Представление числовой информации в компьютере.....	84
5.3 Машинное представление вещественных чисел.....	92
5.4 Кодирование графической информации.....	93
5.5 Кодирование звуковой информации	96
5.6 Представление видеоинформации в компьютере.....	98
Вопросы и задания для самоконтроля	101
6 КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ	103
6.1 Основные принципы работы с текстовым процессором MicrosoftOfficeWord.....	103
6.2 Основные принципы работы с табличным процессором MicrosoftOfficeExcel.....	125
6.3 Математический пакет программ MathCAD	140
Вопросы и задания для самоконтроля	177
ЗАКЛЮЧЕНИЕ	179
ЛИТЕРАТУРА	180

ИНФОРМАТИКА

Учебное пособие

*Пушкарева Татьяна Павловна
Калитина Вера Владимировна
Брит Анна Александровна
Титовская Наталья Викторовна*

Редактор Т.М. Мاستрич

Электронное издание

Подписано в свет 14.02.2024. Регистрационный номер 134
Редакционно-издательский центр Красноярского государственного аграрного университета
660017, Красноярск, ул. Ленина, 117
e-mail: rio@kgau.ru