



10



Л. Л. Босова
А. Ю. Босова

ИНФОРМАТИКА

БАЗОВЫЙ УРОВЕНЬ



ИЗДАТЕЛЬСТВО

БИНОМ

Л. Л. Босова, А. Ю. Босова

ИНФОРМАТИКА

10 класс

Базовый уровень



Москва

БИНОМ. Лаборатория знаний

2017

УДК 004.9
ББК 32.97
Б85

Босова Л. Л.
Б85 Информатика. 10 класс. Базовый уровень / Л. Л. Босова, А. Ю. Босова. — М. : БИНОМ. Лаборатория знаний, 2017. — 288 с. : ил.

ISBN 978-5-9963-3141-3

Учебное издание предназначено для изучения информатики на базовом уровне в 10 классе общеобразовательных организаций. Включает вопросы, касающиеся информации и информационных процессов, математических основ информатики, аппаратного и программного обеспечения компьютера, информационных технологий.

Содержание учебного издания опирается на материал, изученный в 7–9 классах основной школы. Учебное издание позволяет каждому обучающемуся овладеть ключевыми понятиями и закономерностями, на которых строится современная информатика, научиться выполнять задания ЕГЭ базового уровня сложности. Мотивированный обучающийся научится выполнять ряд заданий ЕГЭ повышенного уровня сложности.

Соответствует федеральному государственному образовательному стандарту среднего общего образования и примерной основной образовательной программе среднего общего образования.

УДК 004.9
ББК 32.97

Учебное издание

Босова Людмила Леонидовна

Босова Анна Юрьевна

ИНФОРМАТИКА

10 класс

Базовый уровень

Редактор *Е. В. Баклашова*

Художник *Н. А. Новак*

Технический редактор *Е. В. Денюкова*

Корректор *Е. Н. Клитина*

Компьютерная верстка: *Л. В. Катуркина*

В оформлении учебника использованы изображения, приведённые на сайтах school-collection.edu.ru (с. 65, 66, 104), www.3dnews.ru (с. 68), www.nvidia.ru (с. 69), www.computerra.ru (с. 75), luntiks.com (с. 147), ru.wikipedia.org (с. 256, форзац), www.zedge.net (с. 256, форзац).

Подписано в печать 25.05.17. Формат 70x100/16. Усл. печ. л. 23,4.

Тираж 5000 экз. Заказ № 40316.

ООО «БИНОМ. Лаборатория знаний»

127473, Москва, ул. Краснопролетарская, д. 16, стр. 1,

тел. (495)181-53-44, e-mail: binom@Lbz.ru

http://www.Lbz.ru, http://metodist.Lbz.ru

Отпечатано в соответствии с качеством предоставленных издательством электронных носителей в АО «Саратовский полиграфкомбинат».

410004, г. Саратов, ул. Чернышевского, 59. www.sarpk.ru

ISBN 978-5-9963-3141-3

© ООО «БИНОМ. Лаборатория знаний», 2016

Уважаемые старшекласники!

Вы уже знаете, что информатика — научная дисциплина о закономерностях протекания информационных процессов в различных средах, а также о методах и средствах их автоматизации. Методы и средства информатики с каждым днём всё больше проникают во все сферы жизни и области знания. Изучение информатики в школе важно не только для тех из вас, кто планирует стать специалистом, разрабатывающим новые информационные технологии, не менее важно оно и для тех, кто планирует стать в будущем физиком или медиком, историком или филологом, руководителем предприятия или политиком, представителем любой другой области знаний или профессии.

В старшей школе информатика (как и любой другой предмет) может изучаться на двух уровнях: базовом и углублённом. Результаты изучения предмета на базовом уровне ориентированы в первую очередь на общую функциональную грамотность, получение компетентностей для повседневной жизни и общего развития. Они включают в себя:












- понимание предмета, ключевых вопросов и основных составляющих элементов изучаемой предметной области;
- умение решать основные практические задачи, характерные для использования методов и инструментария данной предметной области;
- осознание рамок изучаемой предметной области, ограниченности методов и инструментов, типичных связей с некоторыми другими областями знания.

Наш учебник предназначен для изучения предмета на базовом уровне. Его материал структурирован по уже известным вам тематическим блокам: «Математические основы информатики», «Алгоритмы и элементы программирования», «Использование программных систем и сервисов». Содержание учебника направлено на дальнейшее развитие информационных компетенций и информационной культуры выпускника.

Все ученики, изучающие информатику на базовом уровне, должны овладеть ключевыми понятиями и закономерностями, на которых строится предметная область информатики.

Те из вас, кто проявит особый интерес к изучению информатики, не только научатся выполнять задания базового уровня сложности, но и смогут научиться выполнять многие задания повышенного уровня сложности и отдельные задания высокого уровня сложности (помечены знаком «*»), в том числе входящие в контрольно-измерительные материалы единого государственного экзамена (ЕГЭ) по информатике.

В работе с учебником вам помогут навигационные значки:

-  — важное утверждение или определение;
-  — интересная информация;
-  — пример решения задачи;
-  — ссылка на ресурс в Интернете;
-  — дополнительный материал, содержащийся в авторской мастерской Л. Л. Босовой (<http://metodist.Lbz.ru>);
-  — вопросы в тексте параграфа, вопросы и задания для самоконтроля;
-  — задания для подготовки к итоговой аттестации;
-  — домашний проект или исследование;
-  — задания для выполнения на компьютере;
-  — групповая работа;
-  — межпредметные связи.

Желаем успехов в изучении информатики!

Глава 1

ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

§ 1

Информация. Информационная грамотность и информационная культура

1.1. Информация, её свойства и виды

Информация (от лат. *informatio* — осведомление, разъяснение, изложение) — одно из фундаментальных понятий современной науки, не объясняемых через другие понятия. Наряду с такими понятиями, как «вещество» и «энергия», понятие «информация» определяет основу современной научной картины мира. Строгое и однозначное определение этому термину дать невозможно.

Можно выделить некоторые особенности информации, подчёркивающие её принципиальное отличие от вещества и энергии:

- по отношению к информации неприменим закон сохранения (информация может внезапно, «ниоткуда» возникнуть в воображении человека; человек может её забыть, утратив навсегда и невратно);
- информация не расходуется при её использовании;
- информация может быть передана от одного человека к другому, при этом её количество у первого человека не уменьшится;
- ценность суммы информации может превосходить сумму ценностей её частей.

Согласно одной из философских концепций (её называют атрибутивной), информация является свойством (атрибутом) всего сущего: информация возникла вместе со Вселенной и проявляется в изменениях, к которым приводит воздействие одних объектов на другие.

Согласно так называемой функциональной философской концепции, информация связана с функционированием сложных самоорганизующихся систем, к которым относятся живые организ-

мы. Иначе говоря, информация является атрибутом, присущим только живой природе.

С точки зрения антропоцентрической философской концепции информация существует лишь в человеческом сознании и информационная деятельность присуща только человеку.

Подумайте, как соотносятся между собой три приведённые выше философские концепции. К какой из них ближе понимание термина «информация» с точки зрения обыденных житейских представлений?

Закон Российской Федерации «Об информации, информационных технологиях и о защите информации» определяет информацию как «сведения (сообщения, данные) независимо от формы их представления».

В курсе информатики основной школы мы говорили, что информация для человека — это содержание сигналов (сообщения), воспринимаемых человеком непосредственно или с помощью специальных устройств, расширяющее его знания об окружающем мире и протекающих в нём процессах.

Информация обладает такими характеристиками, как объективность, достоверность, полнота, актуальность, понятность, релевантность и др.

Информация **объективна**, если она не зависит от свойств источника информации. Объективную информацию можно получить с помощью измерительных приборов. Отражаясь в сознании конкретного человека, информация перестаёт быть объективной, становится субъективной, т. к. преобразовывается (в большей или меньшей степени) в зависимости от опыта, знаний, пристрастий конкретного человека (субъекта).

Возможна ли ситуация, когда информация абсолютно не зависит от свойств источника информации?

Информация **достоверна**, если она отражает реальное положение дел, в частности не вступает в противоречие с уже имеющейся информацией, признаваемой в качестве достоверной. При этом не исключается, что с появлением новой информации та или иная информация может перестать быть достоверной.

Информация **полна**, если её достаточно для понимания ситуации и принятия решения. Неполная информация может привести к ошибочному выводу или решению. Информация избыточна, если для решения достаточно только части данной информации.

Информация **актуальна** (своевременна), если она важна, существенна именно в данный момент времени.

Информация **понятна**, если она выражена на языке, доступном для получателя.

Информация **релевантна**, если она соответствует нуждам (запросам) потребителя.

Важно понимать, что теми или иными свойствами информация обладает только в некоторый промежуток времени, т. е. свойства информации имеют относительный по времени характер.

Приведите известные вам из истории примеры, когда информация, считавшаяся достоверной, становилась недостоверной в связи с появлением новой информации.

В зависимости от выбираемого основания классификации можно выделять разные виды информации. Некоторые из возможных вариантов классификаций представлены на рисунке 1.1.



Рис. 1.1. Виды информации

1.2. Информационная культура и информационная грамотность

В материалах ЮНЕСКО (от англ. *UNESCO — United Nations Educational, Scientific and Cultural Organization* — специализированное учреждение Организации Объединённых Наций по вопросам образования, науки и культуры) отмечается всеобъемлющий характер понятия информации: информация — «универсальная

субстанция, пронизывающая все сферы человеческой деятельности, служащая проводником знаний и мыслей, инструментом общения, взаимопонимания и сотрудничества».

Важная и всё возрастающая роль информации в современном мире требует от человека наличия информационной культуры, включающей в себя:

- понимание закономерностей протекания информационных процессов;
- умение оценивать объективность, достоверность, полноту, актуальность, полезность поступающей информации;
- умение представлять информацию в разных формах;
- умение обрабатывать информацию с помощью подходящих информационных технологий;
- умение применять полученную информацию для принятия решений;
- соблюдение этических норм и правил при использовании информации.

Информационная культура — готовность человека к жизни и деятельности в современном высокотехнологичном информационном обществе, умение эффективно использовать возможности этого общества и защищаться от его негативных воздействий.

Важнейшими компонентами информационной культуры являются информационное мировоззрение и информационная грамотность.

Под информационной грамотностью понимается наличие знаний и умений, требуемых для:

- правильной идентификации информации, необходимой для выполнения определённого задания или решения проблемы;
- выработки стратегии эффективного поиска информации;
- организации и реорганизации информации;
- интерпретации и анализа найденной и извлеченной информации (например, после скачивания из Интернета);
- оценки объективности, достоверности, полноты, актуальности, полезности полученной информацией;
- соблюдения этических норм и правил пользования полученной информацией;
- передачи и представления результатов анализа и интерпретации информации другим лицам;

- последующего применения информации для осуществления определённых действий и достижения определённых результатов.

Такая трактовка информационной грамотности была предложена в 2006 году Международной ассоциацией школьных библиотек (IASL).

На рисунке 1.2 изображена международная эмблема информационной грамотности.



Рис. 1.2. Международная эмблема информационной грамотности

Информационная грамотность — способность человека идентифицировать потребность в информации, умение её эффективно искать, оценивать и использовать.

1.3. Этапы работы с информацией

Работу человека с информацией можно разделить на несколько этапов:

- 1) стартовый этап;
- 2) этап поиска информации;
- 3) этап осмысления полученной информации;
- 4) этап рефлексии.

На стартовом этапе необходимо сформулировать цель работы с информацией (Для чего мне нужна информация?); постараться вспомнить всю информацию по интересующему вопросу, которой вы уже обладаете; осознать и чётко сформулировать информационную потребность.

Этап поиска информации включает в себя, прежде всего, поиск источников информации и их проверку на актуальность, достоверность, полезность и т. д. Различают:

- медийные источники информации (интернет-издания, газеты, журналы, телевидение, радио);
- специализированные источники информации (научные труды, аналитические статьи, сайты определённых компаний и др.);
- живые источники информации (люди, компетентные в том или ином вопросе).

Желательно иметь несколько надёжных источников информации по интересующему вас вопросу. Если все они предоставляют схожие сведения, то вероятность того, что найденная информация достоверна, существенно повышается.

Многие годы человеку для получения образования было достаточно заучить и уметь воспроизвести некоторую фиксированную сумму знаний. В наше время потоки информации лавинообразно возрастают. Для успешной профессиональной деятельности требуется постоянное обновление знаний и продолжение образования в течение всей жизни. Сегодня на первый план выдвигается требование творческого освоения информации, предполагающее её осмысление, оценку и создание на этой основе новой информации.

На этапе осмысления происходит в первую очередь восприятие информации. Следующий шаг — извлечение смысла: воспринятая информация упорядочивается, осуществляется анализ, сравнение и обобщение полученной информации, её соотнесение с собственным опытом человека. На этой основе формулируются гипотезы, высказываются предположения, делаются обобщения и выводы. В ответ на полученную информацию у человека возникают собственные мысли. Этот шаг можно охарактеризовать как создание собственного смысла.

На этапе осмысления информации большую роль играет умение правильно ставить вопросы. Можно выделить шесть групп вопросов, постановка которых позволяет эффективно извлекать необходимую информацию.

1. Простые вопросы, отвечая на которые можно получить какие-то конкретные сведения, вспомнить и воспроизвести некую информацию. (Когда произошло ...? Что произошло ...? Кто совершил ...?)
 2. Уточняющие вопросы, служащие для установления обратной связи с источником информации. Иногда их задают для получения информации, отсутствующей, но подразумеваемой в сообщении. (Верно ли я понял, что ...?)
 3. Интерпретационные, или выясняющие причины, вопросы, начинающиеся, как правило, со слова «почему».
 4. Творческие вопросы, в формулировке которых присутствуют элементы условности, предположения, прогноза. (Что будет, если ...? Можно ли сделать так, чтобы ...?)
 5. Оценочные вопросы, помогающие определить значимость информации, дать предельно точную оценку связанным с вашей проблемой предметам, событиям, фактам. (Что привлекательного в ...? Почему есть сомнения в ...?)
 6. Практические вопросы, устанавливающие взаимосвязь между теорией и практикой, возможность практического использования информации. (Где может пригодиться ...? Как использовать ...?)
- Часто используется «ромашка Блума» — графическое изображение системы рассмотренных вопросов, названное так в честь американского педагога и психолога Бенджамина Блума (рис. 1.3).

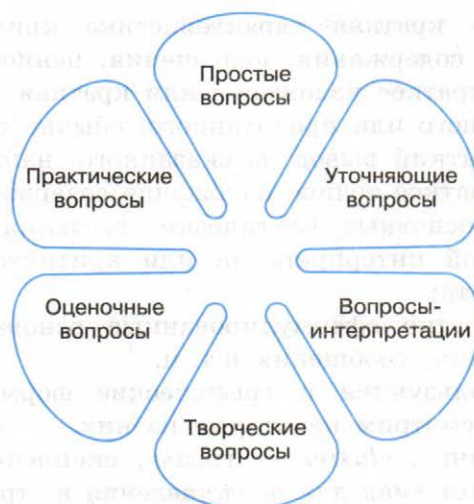


Рис. 1.3. Ромашка Блума

На этапе рефлексии важно оценить эффективность проделанной работы, понять возможности использования полученной информации при решении других задач, осознать влияние этой информации на свои личные взгляды и поведение.

1.4. Некоторые приёмы работы с текстовой информацией

Основная часть накопленной человечеством информации зафиксирована в текстовой форме: Умение эффективно работать с текстовой информацией — важное условие успешной учёбы и профессиональной деятельности в современном мире.

При чтении текстов их можно размечать знаками, например такими:

- + — знакомая информация;
- ! — новая информация;
- ? — информация, вызывающая сомнение и вопросы.

Очень важно уметь выделять в тексте главное, концентрируя внимание на опорных (ключевых) словах и фразах, несущих основную смысловую нагрузку.

Как правило, тексты избыточны — в них используется гораздо больше слов, чем требуется для передачи информации. С одной стороны, «лишние» слова нужны, т. к. позволяют лучше понять смысл текста. С другой стороны, поняв текст, можно выполнить его смысловое свёртывание, избавившись от лишних слов. Вот некоторые формы представления свёрнутой информации:

- аннотация — краткая характеристика книги, статьи или рукописи, их содержания, назначения, ценности и т. д.;
- конспект — краткое изложение или краткая запись содержания услышанного или прочитанного, обычно своими словами;
- резюме — краткий вывод из сказанного, написанного;
- реферат — краткое точное изложение содержания документа, включающее основные фактические сведения и выводы, без дополнительной интерпретации или критических замечаний автора реферата;
- тезисы — кратко сформулированные основные положения доклада, лекции, сообщения и т. п.

Широко используются и графические формы свёртывания информации. Рассмотрим некоторые из них.

Кластер (от англ. *cluster* — гроздь, скопление) — графическая схема, используемая для представления в структурированном виде ключевых слов и словосочетаний, относящихся к рассматриваемому вопросу. При построении кластера в центральном овале располагают ключевое понятие. В овалах второго уровня — понятия, раскрывающие смысл ключевого. В овалах третьего уровня идёт детализация понятий, упомянутых на предыдущем уровне и т. д. На рисунке 1.4 представлен кластер, описывающий виды компьютерного интерфейса.



Рис. 1.4. Пример кластера

Ещё одна графическая форма свёртывания информации — **денотатный граф** (от лат. *denoto* — обозначаю). При его построении необходимо придерживаться следующих правил.

1. Выделите ключевое понятие (слово или словосочетание) и проанализируйте его существенные признаки. Впишите ключевое понятие в верхний прямоугольник.

2. Как можно более точно подберите глаголы, связывающие ключевое понятие с его существенными признаками и отражающие движение от понятия к этим признакам. Это могут быть:
- глаголы, обозначающие цель (направлять, предполагать, приводить, давать и т. д.);
 - глаголы, обозначающие процесс достижения результата (достигать, осуществлять, выполнять и т. д.);
 - глаголы, обозначающие предпосылки достижения результата (основываться, опираться, базироваться и т. д.);
 - самые разнообразные глаголы-связки, с помощью которых осуществляется выход на определение понятия.

Впишите глаголы в прямоугольники второго уровня.

3. Конкретизируйте в прямоугольниках следующего уровня смысл выбранных вами глаголов для более полного раскрытия ключевого понятия.
4. Следите за чередованием имени и глагола. Именем может быть одно существительное или группа существительных в сочетании с другими именными частями речи.
5. Проверяйте каждый блок включённой в граф информации во избежание ошибок, несоответствий и противоречий.

На рисунке 1.5 изображён денотатный граф, раскрывающий существенные признаки понятия «система».

Для обобщения и систематизации крупных блоков информации можно использовать **интеллект-карты**, принципы построения

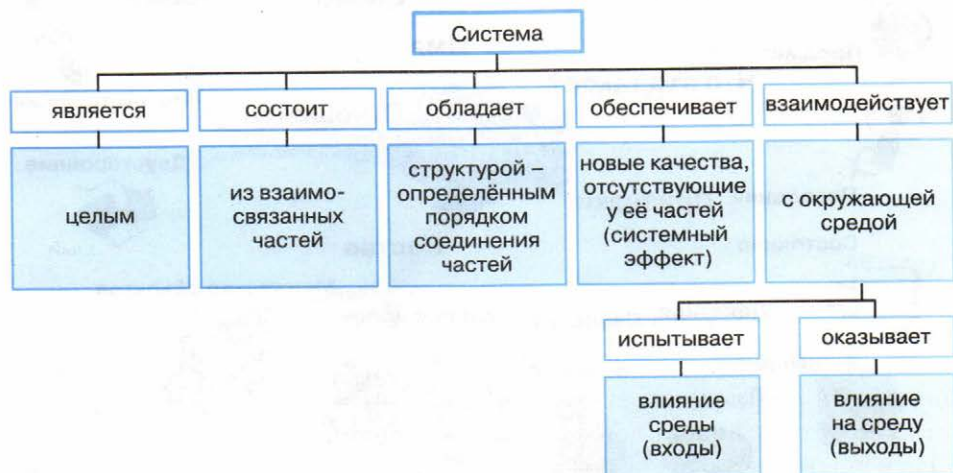


Рис. 1.5. Пример денотатного графа

которых придумал британский психолог Тони Бьюзен. Строгих правил построения интеллект-карт не существует, но при их создании можно учитывать следующие рекомендации.

1. В центре листа картинкой или несколькими словами обозначьте основное понятие, возьмите его в рамку или в кружок.
2. От центрального объекта нарисуйте в разные стороны цветные ветви — основные связанные с ним понятия, свойства, ассоциации. Подпишите каждую ветвь одним-двумя словами (надпись должна быть разборчивой).
3. Рисуя интеллект-карту, применяйте как можно больше цветов и как можно чаще используйте рисунки.
4. От каждой ветви нарисуйте несколько более тонких веточек — развитие ассоциаций, уточнение понятий, детализация свойств, конкретизация направлений.
5. Смысловые блоки отделяйте линиями, берите в рамку.
6. Связи между элементами интеллект-карты показывайте стрелками разного цвета и толщины.
7. Сначала оформляйте основные идеи, а затем уже их редактируйте, делая карту более понятной и красивой.

На рисунке 1.6 представлена интеллект-карта, дающая развернутое представление о понятии «объект».



Рис. 1.6. Пример интеллект-карты

САМОЕ ГЛАВНОЕ

Информация (от лат. *informatio* — осведомление, разъяснение, изложение) — одно из фундаментальных понятий современной науки, не объясняемых через другие понятия. Информация обладает такими характеристиками, как объективность, достоверность, полнота, актуальность, полезность, понятность и др.

Информационная культура — готовность человека к жизни и деятельности в современном высокотехнологичном информационном обществе, умение эффективно использовать возможности этого общества и защищаться от его негативных воздействий.

Информационная грамотность — способность человека идентифицировать потребность в информации, умение её эффективно искать, оценивать и использовать.

Работу человека с информацией можно разделить на несколько этапов: стартовый этап, этап поиска информации, этап осмысления полученной информации, этап рефлексии.

Основная часть накопленной человечеством информации зафиксирована в текстовой форме. Умение эффективно работать с текстовой информацией — важное условие успешной учёбы и профессиональной деятельности в современном мире.

Вопросы и задания



1. Почему нельзя дать строгое и однозначное определение термину «информация»? В чём состоит принципиальное отличие информации от вещества и энергии?
2. Какие существуют основные философские концепции, связанные с понятием информации?
3. Что такое информационная культура человека? Каковы её основные составляющие?
4. Перечислите и охарактеризуйте основные этапы работы человека с информацией.
5. Объясните значение слов «идентификация», «организация», «реорганизация», «интерпретация», «рефлексия». Представьте один из перечисленных процессов с помощью сиквейна — пятистрочника, строящегося по следующей схеме:
 - 1) название темы сиквейна (в данном случае название процесса) одним словом или словосочетанием;
 - 2) описание темы двумя прилагательными или причастиями;
 - 3) описание действий, связанных с темой, тремя словами;
 - 4) фраза из четырёх слов, показывающая отношение к теме, объекту;



- 5) синоним, выраженный одним словом, повторяющий суть темы; ассоциация с темой.
6. В какой ситуации вам могут пригодиться вопросы, образующие ромашку Блума? Попробуйте сформулировать хотя бы по одному вопросу из каждой группы.
7. Какие слова называют ключевыми? Перечислите ключевые слова этого параграфа.
8. Поясните суть смыслового свёртывания информации.
9. Какие текстовые формы представления свёрнутой информации вам известны? Используйте одну из них для представления содержания этого параграфа.
10. Какие графические формы представления свёрнутой информации вам известны? Используйте одну из них для представления содержания этого параграфа.
11. Используйте интеллект-карту (рис. 1.6) для того, чтобы сделать краткое сообщение об объектах.
12. Используйте денотатный граф (рис. 1.5) для того, чтобы сделать краткое сообщение о системах.

§ 2

Подходы к измерению информации

Информация и её свойства являются объектом исследования целого ряда научных дисциплин, таких как:

- теория информации (математическая теория систем передачи информации);
- кибернетика (наука об общих закономерностях процессов управления и передачи информации в машинах, живых организмах и обществе);
- информатика (изучение процессов сбора, преобразования, хранения, защиты, поиска и передачи всех видов информации и средств их автоматизированной обработки);
- семиотика (наука о знаках и знаковых системах);
- теория массовой коммуникации (исследование средств массовой информации и их влияния на общество) и др.

Рассмотрим более детально подходы к определению понятия информации, важные с позиций её измерения:

- 1) определение К. Шеннона, применяемое в математической теории информации;

- 2) определение А. Н. Колмогорова, применяемое в отраслях информатики, связанных с использованием компьютеров.

2.1. Содержательный подход к измерению информации

Клод Шеннон, разрабатывая теорию связи, предложил характеризовать информативность сообщения содержащейся в нём полезной информацией, т. е. той частью сообщения, которая снимает полностью или уменьшает существующую до её получения неопределённость какой-либо ситуации.



Клод Элвуд Шеннон (1916–2001) — американский инженер и математик. Является основателем теории информации, нашедшей применение в современных высокотехнологических системах связи. В 1948 году предложил использовать слово «бит» для обозначения наименьшей единицы информации.

Информация — это снятая неопределённость. Величина неопределённости некоторого события — это количество возможных результатов (исходов) данного события.

Сообщение, уменьшающее неопределённость знания в 2 раза, несёт 1 бит информации.

Такой подход к измерению информации называют содержательным.

Пример 1. Допустим, вы подбрасываете монету, загадывая, что выпадет: «орёл» или «решка». Перед подбрасыванием монеты неопределённость знания о результате равна двум. Действительно, есть всего два возможных результата этого события (бросания монеты). Эти результаты мы считаем равновероятными, т. к. ни один из них не имеет преимущества перед другим.

После того как конкретный исход стал известен (например, подброшенная монета упала «орлом» вверх), неопределённость уменьшилась в 2 раза. Таким образом, сообщение о том, что подброшенная монета упала «орлом» вверх, несёт в себе 1 бит информации.

Пример 2. Предположим, в книжном шкафу восемь полок. Книга может быть поставлена на любую из них. Сколько бит

информации несёт сообщение о том, что книга поставлена на третью полку?

Ответ на этот вопрос можно получить, если дополнить исходное сообщение ещё несколькими сообщениями так, чтобы каждое из них уменьшало неопределённость знания в 2 раза.

Итак, количество возможных результатов (исходов) события, состоящего в том, что книга поставлена в шкаф, равно восьми: 1, 2, 3, 4, 5, 6, 7 и 8.

Сообщение «Книга поставлена на полку не выше четвёртой» уменьшает неопределённость знания о результате в два раза. Действительно, после такого сообщения остаётся всего четыре варианта: 1, 2, 3 и 4. Получен один бит информации.

Сообщение «Книга поставлена на полку выше второй» уменьшает неопределённость знания о результате в два раза: после этого сообщения остаётся всего два варианта: 3 и 4. Получен ещё один (второй) бит информации.

Сообщение «Книга поставлена на третью полку» также уменьшает неопределённость знания о результате в два раза. Получен третий бит информации.

Итак, мы построили цепочку сообщений, каждое из которых уменьшало неопределённость знания о результате в два раза, т. е. несло 1 бит информации. Всего было набрано 3 бита информации. Именно столько информации и содержится в сообщении «Книга поставлена на третью полку».

Подумайте, сколько информации содержится в сообщении о том, что книга поставлена на пятую полку. Обоснуйте свой ответ, построив соответствующую цепочку сообщений.

Метод поиска, на каждом шаге которого отбрасывается половина вариантов, называется методом половинного деления. Этот метод широко используется в компьютерных науках.

Пример 3. О результатах футбольного матча между клубами «Спартак» и «Динамо» известно, что больше трёх мячей никто не забил. Всего возможных вариантов счёта матча — 16:

0 : 0	0 : 1	0 : 2	0 : 3
1 : 0	1 : 1	1 : 2	1 : 3
2 : 0	2 : 1	2 : 2	2 : 3
3 : 0	3 : 1	3 : 2	3 : 3

Здесь первая цифра в каждой паре соответствует количеству мячей, забитых командой «Спартак», вторая — командой «Динамо».

Будем считать все варианты равновероятными и отгадывать счёт, задавая вопросы, на которые можно ответить только «да» или «нет». Вопросы будем формулировать так, чтобы количество возможных вариантов счёта каждый раз уменьшалось вдвое. Это позволит нам:

- 1) обойтись минимальным количеством вопросов;
- 2) подсчитать, сколько бит информации содержит сообщение о счёте матча.

Вопрос 1. «Спартак» забил больше одного мяча? Предположим, получен ответ «Нет». Такой ответ позволяет не рассматривать варианты, расположенные в нижней части таблицы, т. е. сокращает количество возможных исходов в 2 раза:

0 : 0	0 : 1	0 : 2	0 : 3
1 : 0	1 : 1	1 : 2	1 : 3
2 : 0	2 : 1	2 : 2	2 : 3
3 : 0	3 : 1	3 : 2	3 : 3

Вопрос 2. «Спартак» забил один мяч? Предположим, получен ответ «Да». Такой ответ позволяет не рассматривать варианты, расположенные в верхней строке таблицы, т. е. сокращает количество возможных исходов ещё в 2 раза:

0 : 0	0 : 1	0 : 2	0 : 3
1 : 0	1 : 1	1 : 2	1 : 3

Вопрос 3. «Спартак» пропустил больше одного мяча? Предположим, получен ответ «Нет». Можно отбросить ещё два варианта:

1 : 0	1 : 1	1 : 2	1 : 3
-------	-------	-------	-------

Вопрос 4. «Спартак» пропустил один мяч? Предположим, получен ответ «Да». Получаем единственный вариант:

1 : 0	1 : 1
-------	-------

Итак, нам удалось выяснить счёт матча, задав четыре вопроса, ответ на каждый из которых уменьшал неопределённость результата в два раза, т. е. несёт 1 бит информации. Сообщение о счёте матча несёт четыре бита информации.



Выясните, какому счёту матча будут соответствовать следующие цепочки ответов на поставленные выше вопросы:

- 1) Да – Да – Да – Да;
- 2) Нет – Нет – Нет – Нет;
- 3) Да – Нет – Да – Нет.

Попробуйте придумать такие вопросы, чтобы цепочка ответов Нет – Да – Нет – Да приводила к счёту 2 : 3.

Вычислять количество информации, содержащееся в сообщении о том, что имел место один из множества равновероятных результатов некоторого события, с помощью метода половинного деления возможно, но затруднительно. Гораздо проще воспользоваться следующей закономерностью.



Количество информации i , содержащееся в сообщении об одном из N равновероятных результатов некоторого события, определяется из формулы:

$$2^i = N.$$

При N , равном целой степени двойки (2, 4, 8, 16, 32 и т. д.), это уравнение легко решается в уме. Решать такие уравнения при других N вы научитесь чуть позже, в курсе математики 11 класса.



Пример 4. Петя и Вася заинтересовались игрой «Крестики-нолики» на поле $n \times n$. Количество информации, полученное вторым игроком после первого хода первого игрока, составляет 6 бит. Требуется выяснить размеры поля, на котором играют Петя и Вася.

Дано:

$$\begin{array}{l|l|l} i = 6 & 2^i = N & 2^6 = 64 \\ n = ? & n \times n = N & 64 = 8 \times 8 \end{array}$$

Ответ: 8×8 .

2.2. Алфавитный подход к измерению информации

Определение количества информации на основе уменьшения неопределённости наших знаний рассматривает информацию с точки зрения её содержания, понятности и новизны для чело-

века. С этой точки зрения в примере о подбрасывании монеты одинаковое количество информации содержит и зрительный образ упавшей монеты, и короткое сообщение «Орёл», и длинная фраза «В результате подбрасывания монета упала так, что на её видимой части изображён орёл».

Однако при хранении и передаче информации с помощью технических устройств целесообразно отвлекаться от её содержания и рассматривать информацию как последовательность символов (букв, цифр, кодов цвета точек изображения и т. д.) некоторого алфавита.

Информация — последовательность символов (букв, цифр, кодов цвета точек изображения и т. д.) некоторого алфавита.

Минимальная мощность алфавита (количество входящих в него символов), пригодного для кодирования информации, равна 2. Такой алфавит называется двоичным. Один символ двоичного алфавита несёт 1 бит информации.

Согласно Колмогорову, количество информации, содержащейся в последовательности символов, определяется минимально возможным количеством двоичных знаков, необходимых для кодирования этой последовательности, безотносительно к содержанию представленного ею сообщения. Данный подход к определению количества информации называют алфавитным.



Андрей Николаевич Колмогоров (1903–1987) — один из крупнейших математиков XX века. Им получены основополагающие результаты в математической логике, теории сложности алгоритмов, теории информации, теории множеств и ряде других областей математики и её приложений.

Информационным объёмом сообщения называется количество двоичных символов, которое используется для кодирования этого сообщения. В двоичном коде один двоичный разряд несёт 1 бит информации.

В отличие от определения количества информации по Колмогорову в определении информационного объёма не требуется,

чтобы число двоичных символов было минимально возможным. При оптимальном кодировании понятия количества информации и информационного объёма совпадают.

Из курса информатики основной школы вы знаете, что двоичные коды бывают равномерные и неравномерные. Равномерные коды в кодовых комбинациях содержат одинаковое число символов, неравномерные — разное.

Первый равномерный двоичный код был изобретён французом Жаном Морисом Бодо в 1870 году. В коде Бодо используются сигналы двух видов, имеющие одинаковую длительность и абсолютную величину, но разную полярность. Длина кодов всех символов алфавита равна пяти (рис. 1.7).

A	.. 0..	B	.0 .0 0	K	00 0..	S	0. .0 0
É	.. 00.	C	.0 0.0	L	00 00.	T	0. 0.0
E	.. .0.	D	.0 000	M	00 .0.	V	0. 000
I	.. .00	F	.0 .00	N	00 .00	W	0. .00
O	.. 000	G	.0 .0.	P	00 000	X	0. .0.
U	.. 0.0	H	.0 00.	Q	00 0.0	Z	0. 00.
Y	.. .0 0	J	.0 0..	R	00 .0 0	—	0. 0.0

Рис. 1.7. Фрагмент кодовой таблицы кода Бодо

Всего с помощью кода Бодо можно составить $2^5 = 32$ комбинации.

Пример 5. Слово WORD, закодированное с помощью кода Бодо, будет выглядеть так:

0	.	.	0	0	.	.	0	0	0	0	0	.	.	0	.	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Информационный объём такого сообщения равен 20 битам; таково количество двоичных символов, которое используется для кодирования этого сообщения.



Пример 6. Для двоичного представления текстов в компьютере чаще всего используется равномерный восьмиразрядный код. С его помощью можно закодировать алфавит из 256 символов ($2^8 = 256$). Фрагмент кодовой таблицы ASCII представлен на рисунке 1.8.

65	A	01000001	78	N	01001110
66	B	01000010	79	O	01001111
67	C	01000011	80	P	01010000
68	D	01000100	81	Q	01010001
69	E	01000101	82	R	01010010
70	F	01000110	83	S	01010011
71	G	01000111	84	T	01010100
72	H	01001000	85	U	01010101
73	I	01001001	86	V	01010110
74	J	01001010	87	W	01010111
75	K	01001011	88	X	01011000
76	L	01001100	89	Y	01011001
77	M	01001101	90	Z	01011010

Рис. 1.8. Фрагмент кодовой таблицы ASCII

Слово WORD, закодированное с помощью таблицы ASCII:

0 1 0 1 0 1 1 1 0 1 0 0 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 1 0 0

Информационный объем такого сообщения равен 32 битам.

Из курса информатики основной школы вам известно, что с помощью i -разрядного двоичного кода можно закодировать алфавит, мощность N которого определяется из соотношения:

$$2^i = N.$$

Иными словами, зная мощность используемого алфавита, всегда можно вычислить информационный вес символа — минимально возможное количество бит, требуемое для кодирования символов этого алфавита. При этом информационный вес символа должен быть выражен целым числом.

Соотношение для определения информационного веса символа алфавита можно получить и из следующих соображений.

Пусть N — мощность алфавита, используемого для кодирования сообщений. При этом в каждом конкретном сообщении какие-то символы алфавита будут использоваться чаще, какие-то — реже, а какие-то — не будут использоваться вообще. Не станем обращать на это внимание, наоборот, будем считать, что любой из символов может появиться в сообщении с равной вероятностью. Другими словами, появление в сообщении очередного символа — одно из N равновероятных событий. Но количество информации i , содержащееся в сообщении об одном из N равновероятных результатов некоторого события, определяется из формулы $2^i = N$.

Алгоритм вычисления информационного объёма сообщения выглядит так:

- 1) определить мощность используемого алфавита N ;
- 2) из соотношения $2^i = N$ определить i — информационный вес символа алфавита в битах (длину двоичного кода символа из используемого алфавита мощности N);
- 3) вычислить информационный объём сообщения I , умножив информационный вес символа i на количество символов в сообщении K .

При алфавитном подходе информационный объём сообщения I , состоящего из K символов, вычисляется по формуле:

$$I = K \cdot i,$$

где i — информационный вес символа в битах, связанный с мощностью используемого алфавита N соотношением:

$$2^i = N.$$

Пример 7. Для регистрации на некотором сайте пользователю надо придумать пароль, состоящий из 10 символов. В качестве символов можно использовать десятичные цифры и шесть первых букв латинского алфавита, причём буквы используются только заглавные. Пароли кодируются посимвольно. Все символы кодируются одинаковым и минимально возможным количеством бит. Для хранения сведений о каждом пользователе в системе отведено одинаковое и минимально возможное целое число байт.

Необходимо выяснить, какой объём памяти потребуется для хранения 100 паролей.

Дано:

$$\begin{array}{l|l|l}
 N = 10 + 6 = 16 & I_{100} = 100 \cdot I & 16 = 2^i, i = 4 \text{ (бита на символ)} \\
 K = 10 & I = K \cdot i & I = 10 \cdot 4 = 40 \text{ (бит)} = 5 \text{ (байт)} \\
 I_{100} = ? & N = 2^i & I_{100} = 100 \cdot 5 = 500 \text{ (байт)}
 \end{array}$$

Ответ: 500 байт.

2.3. Единицы измерения информации

Итак, в двоичном коде один двоичный разряд несёт 1 бит информации. 8 бит образуют один байт. Помимо бита и байта, для измерения информации используются более крупные единицы:

- 1 Кбайт (килобайт) = 2^{10} байт;
- 1 Мбайт (мегабайт) = 2^{10} Кбайт = 2^{20} байт;
- 1 Гбайт (гигабайт) = 2^{10} Мбайт = 2^{20} Кбайт = 2^{30} байт;
- 1 Тбайт (терабайт) = 2^{10} Гбайт = 2^{20} Мбайт = 2^{30} Кбайт = 2^{40} байт;
- 1 Пбайт (петабайт) = 2^{10} Тбайт = 2^{20} Гбайт = 2^{30} Мбайт = 2^{40} Кбайт = 2^{50} байт.

Исторически сложилось так, что приставки «кило», «мега», «гига», «тера» и др. в информатике трактуются не так, как в математике, где «кило» соответствует 10^3 , «мега» — 10^6 , «гига» — 10^9 , «тера» — 10^{12} и т. д.

Это произошло потому, что $2^{10} = 1024 \approx 1000 = 10^3$. Поэтому 1024 байта и стали называть килобайтом, 2^{10} килобайта стали называть мегабайтом и т. д.

Чтобы избежать путаницы с различным использованием одних и тех же приставок, в 1999 г. Международная электротехническая комиссия ввела новый стандарт наименования двоичных приставок. Согласно этому стандарту, 1 килобайт равняется 1000 байт, а величина 1024 байта получила новое название — 1 кибибайт (Кибайт).

У нас в стране в 2009 году принято «Положение о единицах величин, допускаемых к применению в Российской Федерации». В нём сказано, что наименование и обозначение единицы количества информации «байт» (1 байт = 8 бит) применяются с двоичными приставками «кило», «мега», «гига», которые соответствуют множителям « 2^{10} », « 2^{20} » и « 2^{30} » (1 Кбайт = 1024 байт, 1 Мбайт = 1024 Кбайт, 1 Гбайт = 1024 Мбайт). Данные приставки пишутся с большой буквы.



Рассмотрим ещё несколько примеров решения задач, связанных с определением информационного объёма сообщений.

Пример 8. При регистрации в компьютерной системе каждому пользователю выдаётся пароль длиной в 12 символов, образованный из десятичных цифр и первых шести букв английского алфавита, причём буквы могут использоваться как строчные, так и прописные — соответствующие символы считаются разными. Пароли кодируются посимвольно. Все символы кодируются одинаковым и минимально возможным количеством бит. Для хранения сведений о каждом пользователе в системе отведено одинаковое и минимально возможное целое число байт. Кроме собственно пароля для каждого пользователя в системе хранятся дополнительные сведения, для которых отведено 12 байт. На какое максимальное количество пользователей рассчитана система, если для хранения сведений о пользователях в ней отведено 200 Кбайт?

Прежде всего, выясним мощность алфавита, используемого для записи паролей: $N = 6$ (буквы прописные) + 6 (буквы строчные) + 10 (десятичные цифры) = 22 символа.

Для кодирования одного из 22 символов требуется 5 бит памяти (4 бита позволяют закодировать всего $2^4 = 16$ символов, 5 бит позволяют закодировать уже $2^5 = 32$ символа); 5 — минимально возможное количество бит для кодирования 22 разных символов алфавита, используемого для записи паролей.

Для хранения всех 12 символов пароля требуется $12 \cdot 5 = 60$ бит. Из условия следует, что пароль должен занимать целое число байт; т. к. 60 не кратно восьми, возьмём ближайшее большее значение, которое кратно восьми: $64 = 8 \cdot 8$. Таким образом, один пароль занимает 8 байт.

Информация о пользователе занимает 20 байт, т. к. содержит не только пароль (8 байт), но и дополнительные сведения (12 байт).

Максимальное количество пользователей («польз.»), информацию о которых можно сохранить в системе, равно 10 240:

$$\frac{200 \text{ Кбайт}}{20 \text{ байт/польз.}} = \frac{200 \cdot 1024 \text{ байт}}{20 \text{ байт/польз.}} = 10240 \text{ польз.}$$

Пример 9. Объём сообщения, состоящего из 8192 символов, равен 16 Кбайт. Какова максимальная мощность алфавита, использованного при передаче сообщения?

Дано:

$I = 16 \text{ Кбайт} =$	$I = K \cdot i, i = I/K$	$i = 16 \cdot 2^{13}/8192 =$
$= 16 \cdot 2^{13} \text{ бит}$	$N = 2^i$	$= 16 \text{ (бит)}$
$K = 8192 \text{ символа}$		$N = 2^{16} =$
$N = ?$		$= 65\,536 \text{ (символов)}$

Ответ: максимальная мощность алфавита — 65 536 символов.

САМОЕ ГЛАВНОЕ

Информация (по Шеннону) — это снятая неопределённость. Величина неопределённости некоторого события — это количество возможных результатов (исходов) данного события. Сообщение, уменьшающее неопределённость знания в 2 раза, несёт 1 бит информации. Количество информации i , содержащееся в сообщении об одном из N равновероятных результатов некоторого события, определяется из формулы: $2^i = N$. Такой подход к измерению информации называют содержательным.

Информация (по Колмогорову) — последовательность символов (букв, цифр, кодов цвета точек изображения и т. д.) некоторого алфавита. Информационным объёмом сообщения называется количество двоичных символов, которое используется для кодирования этого сообщения. В двоичном коде один двоичный разряд несёт 1 бит информации. Такой подход к измерению информации называют алфавитным.

При алфавитном подходе информационный объём сообщения I , состоящего из K символов, вычисляется по формуле:

$$I = K \cdot i,$$

где i — информационный вес символа в битах, связанный с мощностью используемого алфавита N соотношением $2^i = N$.

Единицы измерения информации:

- 1 байт = 8 бит;
- 1 Кбайт (килобайт) = 2^{10} байт;
- 1 Мбайт (мегабайт) = 2^{10} Кбайт = 2^{20} байт;
- 1 Гбайт (гигабайт) = 2^{10} Мбайт = 2^{20} Кбайт = 2^{30} байт;
- 1 Тбайт (терабайт) = 2^{10} Гбайт = 2^{20} Мбайт = 2^{30} Кбайт = 2^{40} байт;
- 1 Пбайт (петабайт) = 2^{10} Тбайт = 2^{20} Гбайт = 2^{30} Мбайт = 2^{40} Кбайт = 2^{50} байт.



Вопросы и задания

1. Что такое неопределённость знания о результате какого-либо события? Приведите пример.
2. В чём состоит суть содержательного подхода к определению количества информации? Что такое бит с точки зрения содержательного подхода?
3. Паролем для приложения служит трёхзначное число в шестнадцатеричной системе счисления. Возможные варианты пароля:

189	101	654	FFE	123
A41	880	391	110	125

Ответ на какой вопрос (см. ниже) содержит 1 бит информации?

- 1) Это число записано в двоичной системе счисления?
- 2) Это число записано в четверичной системе счисления?
- 3) Это число может быть записано в восьмеричной системе счисления?
- 4) Это число может быть записано в десятичной системе счисления?
- 5) Это число может быть записано в шестнадцатеричной системе счисления?
4. При угадывании целого числа в некотором диапазоне было получено 5 бит информации. Каковы наибольшее и наименьшее числа этого диапазона?
5. Какое максимальное количество вопросов достаточно задать вашему собеседнику, чтобы точно определить день и месяц его рождения?
6. В чём состоит суть алфавитного подхода к измерению информации? Что такое бит с точки зрения алфавитного подхода?
7. Закодируйте фразу «ALL IN GOOD TIME» кодом Бодо и восьмиразрядным компьютерным кодом. Сравните полученные информационные объёмы текста.
8. Какие единицы используются для измерения объёма информации, хранящейся на компьютере?
9. Объём сообщения, содержащего 11 264 символа, равен 11 Кбайт. Определите максимальную мощность алфавита, который мог быть использован для кодирования этого сооб-

- щения? Какова минимальная мощность алфавита, использование которого привело к такому же информационному объёму закодированного сообщения?
10. В школе 750 учащихся, коды учащихся записаны в школьной информационной системе с помощью минимального количества бит. Каков информационный объём в байтах сообщения о кодах 180 учащихся начальных классов? ✓
 11. В школьной базе данных каждый ученик получил идентификатор, состоящий ровно из 6 символов. В качестве символов используются все заглавные буквы русского алфавита, кроме «Ё», «Ы», «Ъ» и «Ь», а также все десятичные цифры за исключением цифры 0. Каждый такой идентификатор в информационной системе записывается минимально возможным и одинаковым целым количеством байт (при этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством бит). Определите объём памяти, необходимый для хранения в этой системе 180 идентификаторов учащихся начальных классов. Ответ выразите в килобайтах. ✓
 12. В ходе телевизионного шоу проводится СМС-голосование: каждый телезритель отдаёт свой голос за одного из 12 артистов-участников шоу, отправляя сообщение с его номером. Голос каждого телезрителя, отданный за того или иного участника, кодируется одинаковым и минимально возможным количеством бит и сохраняется для подведения итогов. За время телевизионного шоу в голосовании приняли участие 163 840 зрителей. Определите объём сохранённой информации о голосовании и выразите его в килобайтах. ✓
 13. При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 6 символов и содержащий только символы из шестибуквенного набора А, В, С, D, Е, F. Для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей и все символы кодируются одинаковым и минимально возможным количеством бит. Кроме собственно пароля для каждого пользователя в системе хранятся дополнительные сведения, занимающие 15 байт. Определите объём памяти в байтах, необходимый для хранения сведений о 120 пользователях. ✓

§ 3 Информационные связи в системах различной природы

3.1. Системы

Окружающий нас мир многообразен и сложен. Он наполнен разнообразными объектами — предметами, процессами и явлениями.

Сложные объекты, состоящие из взаимосвязанных частей (элементов) и существующие как единое целое, принято называть системами.

Всякая система определяется составом своих частей и структурой — порядком объединения элементов в единое целое. Для описания систем часто используется их представление в виде графа. На рисунке 1.9 в виде графа представлена система избирательных комиссий в Российской Федерации.

Соединение, интеграция отдельных элементов приводит к системному эффекту — возникновению у системы новых свойств, не присущих ни одной из её составных частей. Нарушение эле-

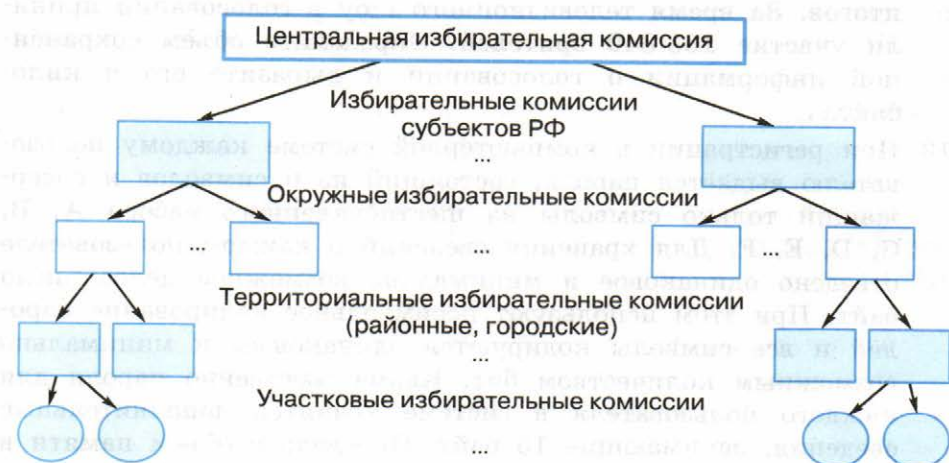


Рис. 1.9. Система избирательных комиссий в Российской Федерации

ментного состава или структуры системы ведёт к частичной или полной утрате её функциональности.

Любой реальный объект бесконечно сложен. Степень детализации при описании его состава и структуры зависит от того, для чего это нужно. Один и тот же объект в одних случаях может рассматриваться как система, включающая в себя другие подсистемы и объекты, а в других — как простой элемент или подсистема, входящая в другую систему.

При исследовании тех или иных объектов окружающего мира недостаточно рассмотреть по отдельности каждый из элементов, входящих в их состав. Изучая деятельность системы, нужно учитывать не только все её элементы, но и связи между ними, а также взаимодействие системы с окружающим миром. В этом состоит суть системного подхода.

Способность применять системный подход (рассматривать исследуемый объект как элемент системы, связанный с множеством других элементов) и оценивать на этой основе ситуацию с разных точек зрения является основой системного мышления, позволяющего человеку выбирать наиболее эффективный путь решения возникающих проблем.

3.2. Информационные связи в системах

Во всем многообразии окружающих нас систем можно выделить системы естественные (природные) и системы искусственные (созданные человеком) — технические и общественные.

Природные системы также можно назвать материальными. И есть множество материальных систем, созданных человеком (транспортная система, энергосистема и др.). Между элементами таких систем существуют материальные связи: физические, энергетические и др.

Но во многих материальных системах, например в системах растительного и животного мира, существуют связи, которые можно назвать информационными:

- цветки и соцветия некоторых растений в течение дня поворачиваются вслед за солнцем;
- многие дикие животные пахучими метками дают знать чужакам, что эта территория уже занята;
- трели соловья служат для привлечения самки и т. д.

Информационные связи присущи и техническим системам:

- автоматическое устройство, называемое термостатом, воспринимает информацию о температуре в помещении и в зависимости от заданного человеком температурного режима включает или отключает отопительные приборы;
- автопилот управляет самолётом в соответствии с заложенной в него программой и т. д.

На основе информационных связей построены общественные (социальные) системы, представляющие собой различные объединения людей.

Информационные связи в системах любой природы состоят в передаче информации от одного элемента системы к другому, в обмене информацией между элементами системы, в хранении и обработке информации, т. е. представляют собой информационные процессы.

3.3. Системы управления

Информация ценна не сама по себе, она нужна для того, чтобы обеспечить успешность некоторых целенаправленных действий. Планомерное воздействие на некоторый объект с целью достижения определённого результата называется управлением. Изучением процессов управления в живых и неживых системах занимается наука кибернетика.

Управление — это процесс целенаправленного воздействия на объект, осуществляемый для организации его функционирования по заданной программе.

С точки зрения кибернетики управление происходит путём информационного взаимодействия между управляющим объектом и объектом управления (рис. 1.10).

Прямая связь подразумевает передачу информации от управляющего объекта к объекту управления. **Обратная связь** — это процесс передачи информации о состоянии объекта управления

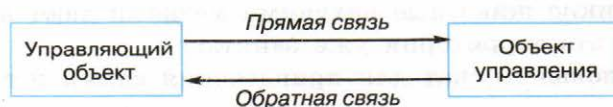


Рис. 1.10. Кибернетическая модель системы управления

управляющему объекту. Назначение обратной связи — коррективировка управляющих воздействий на объект управления в зависимости от его состояния.

Приведите примеры обратной связи, предусмотренной в бытовых приборах, в живых организмах, в обществе.

Все компоненты кибернетической системы управления имеются в организме животного и человека: мозг — управляющий объект, органы движения — объекты управления, нервная система — каналы информационной связи. Таким образом, животное и человек являются естественными (созданными природой) самоуправляемыми системами, т. е. системами, в которых управляющий объект и объект управления представляют собой единое целое.

Все системы управления можно разделить на:

- *неавтоматические системы управления* — человек занимается управлением самостоятельно;
- *автоматизированные системы управления (АСУ)* — сбор необходимой для принятия решения информации и её обработка производятся автоматически, а окончательное решение принимает человек;
- *системы автоматического управления (САУ)* — все операции, связанные с процессами управления, происходят без участия человека по программам, предварительно подготовленным человеком.

Количество автоматизированных и автоматических систем вокруг нас неуклонно возрастает.

САМОЕ ГЛАВНОЕ

Сложные объекты, состоящие из взаимосвязанных частей (элементов) и существующие как единое целое, принято называть системами. Всякая система определяется составом своих частей и структурой — порядком объединения элементов в единое целое.

Информационные связи в системах любой природы состоят в передаче информации от одного элемента системы к другому, в обмене информацией между элементами системы, в хранении и обработке информации, т. е. представляют собой информационные процессы.

Управление — это процесс целенаправленного воздействия на объект, осуществляемый для организации его функционирования по заданной программе.





Вопросы и задания



1. Что такое система? Приведите примеры естественных и искусственных систем, изученных или изучаемых вами на других предметах. Опишите их состав и структуру.
2. Рассмотрите персональный компьютер, имеющийся в кабинете информатики, как простой элемент одной системы, подсистему другой системы и самостоятельную систему, состоящую из других подсистем.
3. Что такое системный эффект? Приведите пример.
4. Выясните значение слова «эмерджентность». Какое отношение оно имеет к теме, рассматриваемой в этом параграфе?
5. В чём состоит суть системного подхода? Почему системный подход так важен для исследования и преобразования окружающего мира?
6. Приведите примеры систем, описание которых удобно представить в виде графа.
7. Какой вклад в науку внесли Николай Коперник и Карл Линней? Что, по вашему мнению, объединяет их открытия?
8. Согласны ли вы с утверждением, что одна и та же информация в разных системах может быть представлена разными способами? Обоснуйте свою точку зрения.
9. Приведите примеры материальных и информационных связей в системах разной природы.
10. Что изучает наука кибернетика? Выясните, кто считается её основоположником.
11. Опишите сущность управления с кибернетической точки зрения.
12. Приведите примеры систем управления, в которых отдельный человек, группа людей, техническое устройство, животное могут быть:
 - 1) объектом управления;
 - 2) управляющим объектом.
13. Приведите примеры автоматических устройств, получающих всё более широкое распространение в наши дни.
14. Постройте кластер, описывающий основные понятия, рассмотренные в этом параграфе.



§ 4

Обработка информации

Информационный процесс — совокупность последовательных действий (операций), производимых над информацией (в виде данных, фактов, идей, гипотез, теорий и пр.) для получения какого-либо результата (достижения цели).

Можно выделить три основных типа информационных процессов: обработка, хранение и передача информации. Рассмотрим их подробнее.

Обработка информации — это целенаправленный процесс изменения содержания или формы представления информации.



4.1. Задачи обработки информации

Из курса информатики основной школы вам известно, что существует два различных типа обработки информации:

- 1) обработка, связанная с получением нового содержания, новой информации;
- 2) обработка, связанная с изменением формы представления информации, не изменяющая её содержания.

К первому типу обработки информации относятся: преобразование по правилам (в том числе вычисления по формулам), исследование объектов познания по их моделям, логические рассуждения, обобщение и др.

Ко второму типу обработки информации можно отнести:

- кодирование — переход от одной формы представления информации к другой, более удобной для восприятия, хранения, передачи или последующей обработки;
- структурирование — организацию информации по некоторому правилу, связывающему её в единое целое;
- поиск и отбор информации, требуемой для решения некоторой задачи, из информационного массива и др.

При всём многообразии решаемых задач в процессе обработки информации всегда решается некоторая информационная задача, а именно: дан некоторый набор исходных данных — исходной информации, требуется получить некоторые результаты — итоговую информацию. Сам процесс перехода от исходных данных

к результату и есть процесс обработки. Тот объект, который осуществляет обработку, может быть назван исполнителем обработки. Для успешного выполнения обработки информации исполнителю должен быть известен способ обработки, т. е. последовательность действий, которую нужно выполнить, чтобы достичь нужного результата. Описание такой последовательности принято называть алгоритмом обработки.

Общая схема процесса обработки информации представлена на рисунке 1.11.

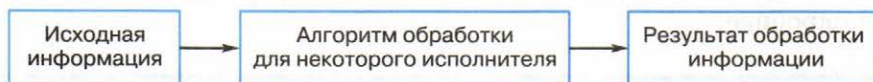


Рис. 1.11. Общая схема процесса обработки информации

Исполнителем обработки информации может быть человек или компьютер. При этом человек, как правило, является неформальным, творчески действующим исполнителем. Даже для решения самой простой математической задачи разные люди могут использовать разные способы!

Что касается компьютера, то он является формальным исполнителем, действия которого осуществляются автоматически, строго в соответствии с имеющимся алгоритмом обработки информации.

Рассмотрим отдельные процессы обработки информации более подробно.

4.2. Кодирование информации

Кодирование информации широко используется в технических средствах работы с информацией (телеграф, радио, компьютеры).

Кодирование — это обработка информации, заключающаяся в её преобразовании в некоторую форму, удобную для хранения, передачи, обработки информации в дальнейшем.

Код — это система (список) условных обозначений (кодовых слов), используемых для представления информации¹⁾.

Кодовая таблица — это совокупность используемых кодовых слов и их значений.

1) Также кодом зачастую называют результат кодирования информации.

Ранее мы уже рассмотрели примеры равномерных двоичных кодов — пятиразрядный код Бодо и восьмиразрядный код ASCII.

Самый известный пример неравномерного кода — код (азбука) Морзе, в которой цифры и буквы алфавита представляются последовательностями длинных («тире») и коротких («точек») сигналов, названный в честь американского изобретателя и художника Сэмюэля Морзе (1791–1872). Буквы, встречающиеся в сообщениях чаще, имеют в этом коде более короткий код, чем «редкие» буквы (рис. 1.12).

A	· —	S	· · ·
B	— · · ·	T	—
C	— · — ·	U	· · —
D	— · ·	V	· · · —
E	·	W	· — —
F	· · — ·	X	— · · —
G	— — ·	Y	— · — —
H	· · · ·	Z	— — · ·
I	· ·	1	· — — — —
J	· — — —	2	· · — — —
K	— · —	3	· · · — —
L	· — · ·	4	· · · · —
M	— —	5	· · · · ·
N	— ·	6	— · · · ·
O	— — —	7	— — · · ·
P	· — — ·	8	— — — · ·
Q	— — · —	9	— — — — ·
R	· — ·	0	— — — — —

Рис. 1.12. Кодовая таблица азбуки Морзе

В азбуке Морзе сигналы отделяются друг от друга паузами — отсутствием сигналов. За единицу «измерения» длительности сигналов принимается длительность сигнала «точка». Длительность тире (длинного сигнала) равна длительности трёх точек (коротких сигналов). Пауза между сигналами одного знака равна одной точке; пауза между знаками в слове — трём точкам; пауза между словами — семи точкам. Фактически пауза является третьим знаком в азбуке Морзе, а сам код — троичным.

Слово WORD, закодированное с помощью азбуки Морзе, на «временной» шкале можно представить так:



Самым знаменитым сообщением, закодированным азбукой Морзе, является сигнал бедствия «SOS». Его запрещено использовать без острой на то необходимости. Передается сигнал без межбуквенных пауз:

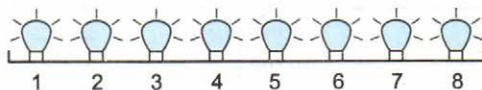


При использовании неравномерных кодов важно понимать, сколько различных кодовых слов они позволяют построить.

Пример 1. Светодиодная панель содержит восемь излучающих элементов, каждый из которых может светиться или красным, или жёлтым, или синим, или зелёным цветом. Сколько различных сигналов можно передать с помощью панели (все излучающие элементы должны гореть, порядок цветов имеет значение)?

На уроках математики и информатики в основной школе вы изучали элементы комбинаторики, в том числе правило умножения. Согласно ему, если элемент A можно выбрать n способами и при любом выборе A элемент B можно выбрать m способами, то пару (A, B) можно выбрать $n \cdot m$ способами. Это правило справедливо и для произвольного количества независимо выбираемых элементов.

Применим его к решению нашей задачи.



Существует 4 варианта выбора цвета первого элемента, 4 варианта выбора цвета второго элемента; цвета для пары элементов

(1, 2) можно выбрать $4 \cdot 4 = 4^2 = 16$ способами; цвета для тройки элементов (1, 2, 3) можно выбрать $16 \cdot 4 = 4^3 = 64$ способами и т. д. Цвета для восьми элементов (1, 2, 3, 4, 5, 6, 7, 8) можно выбрать $4^8 = 65\,536$ способами.

Можно ли отнести эту задачу к классу задач на определение максимально возможного количества комбинаций (слов) фиксированной длины определённого алфавита? Обоснуйте свой ответ.

Пример 2. Выясним, сколько всего различных символов можно закодировать, используя последовательности точек и тире, содержащие не более шести знаков.

Для кодирования различных символов можно использовать последовательности точек и тире, содержащие не более шести знаков, т. е. 1, 2, 3, 4, 5 или 6 знаков.

Последовательностями, содержащими один из двух возможных знаков, можно закодировать два символа: один будет закодирован точкой, второй — тире.

Рассмотрим последовательности, содержащие два знака из двухсимвольного алфавита. Их может быть $2 \cdot 2 = 2^2 = 4$.

Последовательностей из трёх знаков, принадлежащих двухсимвольному алфавиту, может быть $4 \cdot 2 = 2^3 = 8$.

Рассуждая аналогичным образом, подсчитаем число последовательностей, содержащих 4, 5 и 6 знаков — 16, 32 и 64 соответственно.

Число различных последовательностей, содержащих не более шести знаков двухсимвольного алфавита, будет равно $126 = 2 + 4 + 8 + 16 + 32 + 64$.

Пример 3. Имеющаяся информация должна быть закодирована в четырёхбуквенном алфавите {А, В, С, D}. Выясним, сколько существует различных последовательностей из 7 символов четырёхбуквенного алфавита {А, В, С, D}, которые содержат ровно пять букв А.

Нас интересуют семисимвольные последовательности, любые пять мест в которых будут заняты буквой А, а на двух оставшихся местах могут находиться любые из букв В, С, D.

Например:

1	2	3	4	5	6	7
А	А	А	А	А		

Так как на 6-м и 7-м местах могут стоять любые из трёх оставшихся букв В, С, D, то всего существует 9 ($3 \cdot 3 = 9$) разных семибуквенных последовательностей, в которых первые пять позиций заняты буквой А.

Но ведь буквы А могут находиться на любых пяти из имеющихся семи позиций. Например:

1	2	3	4	5	6	7
А	А	А	А			А

1	2	3	4	5	6	7
	А	А	А	А	А	

1	2	3	4	5	6	7
		А	А	А	А	А

А сколько таких вариантов всего? Сколько всего существует способов, которыми мы можем выбрать пять мест из семи для размещения там буквы А?

Для ответа на этот вопрос нужно вспомнить некоторые сведения из изученного в основной школе раздела математики, называемого комбинаторикой.

В комбинаторике набор k элементов, выбранных из данного множества, содержащего n различных элементов, называется сочетанием из n по k .

Для вычисления значения этой величины применяется формула:

$$C_n^k = \frac{n!}{k!(n-k)!}.$$

Здесь $n! = 1 \cdot 2 \cdot \dots \cdot n$.

Действительно, множество, с которым мы имеем дело, состоит из мест для записи символов в последовательности. Его элементы можно обозначить 1, 2, 3, 4, 5, 6 и 7. Требуется выбрать из этого множества пять мест для размещения буквы А. Число возможных вариантов можно вычислить как число сочетаний из 7 по 5:

$$C_7^5 = \frac{7!}{5!(7-5)!} = \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 1 \cdot 2} = \frac{6 \cdot 7}{2} = 21.$$

Итак, существует 21 вариант выбора в семибуквенной последовательности ровно пяти мест для размещения там буквы А. Для каждого из этих 21 вариантов имеется 9 разных вариантов заполнения двух оставшихся мест.

Всего существует 189 ($21 \cdot 9 = 189$) различных последовательностей из 7 символов четырёхбуквенного алфавита {А, В, С, D}, которые содержат ровно пять букв А.

Главное условие использования неравномерных кодов — возможность однозначного декодирования записанного с их помощью сообщения. Именно поэтому в технических системах широкое распространение получили префиксные коды: они состоят из слов разной длины, записываемых без разделительного символа. При этом сообщение, закодированное с их помощью, может быть однозначно декодировано.

Префиксный код — код со словом переменной длины, обладающий тем свойством, что никакое его кодовое слово не может быть началом другого (более длинного) кодового слова.

Например:

- 1) код, состоящий из слов 0, 10 и 11, является префиксным;
- 2) код, состоящий из слов 0, 10, 11 и 100, не является префиксным.

Для того чтобы сообщение, записанное с помощью неравномерного кода, однозначно декодировалось, достаточно, чтобы никакое кодовое слово не было началом другого (более длинного) кодового слова. Это условие ещё называют условием Фано (в честь Роберта Марио Фано, американского учёного, известного по работам в области теории информации).

Обратное условие Фано также является достаточным условием однозначного декодирования неравномерного кода. В нём требуется, чтобы никакой код не был окончанием другого (более длинного) кода.

Для возможности однозначного декодирования достаточно выполнения одного из условий Фано — или прямого, или обратного.

Как вы понимаете смысл этого утверждения? Можно ли на его основе заявлять, что если для некоторого кода условие Фано не выполняется, то однозначное декодирование записанного с его помощью сообщения невозможно?

Пример 4. Двоичные коды для 5 букв латинского алфавита представлены в таблице:

A	B	C	D	E
000	01	100	10	011

Выясним, какое сообщение (какой набор букв) закодировано с помощью этих кодов двоичной строкой: 0110100011000.

Проанализируем имеющиеся коды: код буквы В (01) является началом кода буквы Е (011); код буквы D (10) является началом кода буквы С (100).

Таким образом, прямое условие Фано для заданных кодов не выполняется. Следовательно, имеющуюся двоичную строку нельзя декодировать однозначно, если начать её декодирование с начала (слева направо).

Начните проводить декодирование двоичной строки 0110100011000 слева направо и убедитесь в справедливости условия Фано.

Для имеющихся кодов выполняется обратное условие Фано: никакой код не является окончанием другого кода. Следовательно, имеющуюся двоичную строку можно декодировать однозначно, если начать её декодирование с конца (справа налево).

Итак, направление однозначного декодирования установлено. Процесс декодирования может быть представлен так:

```

0110100011000
    ↓
0110100011A
    ↓
0110100EA
    ↓
0110CEA
    ↓
01DCEA
    ↓
BDCEA
    
```

Если для некоторой последовательности кодов выполняется прямое условие Фано, то её декодирование следует вести слева направо. Если для некоторой последовательности кодов выполняется обратное условие Фано, то её декодирование следует вести справа налево.

Из курса информатики основной школы вам знакомо понятие дерева — иерархической структуры, состоящей из набора вершин и рёбер. Вершина, в которую не входит ни одного ребра, называется корнем; вершины, из которых не выходит ни одного ребра, называются листьями. Дерево, из вершин которого выходит только два ребра, называется двоичным (бинарным) деревом.

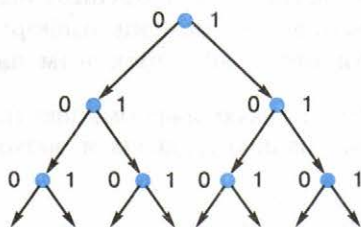
Комбинации, соответствующие листьям бинарного дерева, являются кодовыми комбинациями префиксного кода.

Префиксные коды можно наглядно представить с помощью кодовых деревьев.

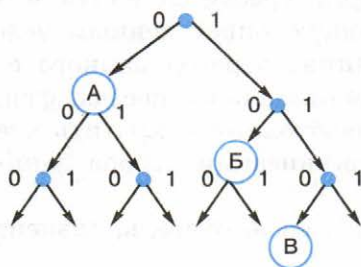
Пример 5. Для кодирования некоторой последовательности, состоящей из букв А, Б, В и Г, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Для букв А, Б и В использовали такие кодовые слова: А — 0, Б — 10, В — 110.

Каким кодовым словом может быть закодирована буква Г? Код должен удовлетворять свойству однозначного декодирования. Если можно использовать более одного кодового слова, укажите кратчайшее из них.

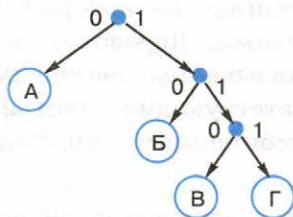
Для решения задачи воспользуемся бинарным деревом.



Отметим вершины, соответствующие используемым кодовым словам: А — 0, Б — 10, В — 110:



Так как комбинациям префиксного кода должны соответствовать листья бинарного дерева, наше кодовое дерево должно выглядеть так:



Итак, для кодирования буквы Г можно использовать код 111.

Усложним условие задачи. Теперь кодируемая последовательность состоит из букв А, Б, В, Г и Д. Кодовые слова для букв А, Б и В определены: А – 0, Б – 10, В – 110.

Какими кодовыми словами могут быть закодированы буквы Г и Д? Код должен удовлетворять свойству однозначного декодирования. Общая длина кодовых слов для всех пяти букв должна быть минимальной.

4.3. Поиск информации

Неоценима роль компьютера в организации важнейшей задачи обработки информации — поиска информации, необходимой человеку для решения стоящей перед ним задачи.

Вспомните, как часто и какую информацию приходится искать вам, членам вашей семьи, вашим друзьям и знакомым. Приведите примеры.

Задача поиска обычно формулируется следующим образом. Имеется некоторое хранилище информации — информационный массив (телефонный справочник, словарь, расписание поездов, диск с файлами и др.). Требуется найти в нём нужную информацию, удовлетворяющую определённым условиям поиска (телефон данной организации, перевод данного слова на английский язык, время отправления данного поезда, файл с рефератом). При этом, как правило, необходимо сократить время поиска, которое зависит от способа организации набора данных и используемого алгоритма поиска.

Алгоритм поиска, в свою очередь, зависит от способа организации информации.

Если данные никак не организованы, то мы имеем дело с неструктурированным набором данных. Для осуществления поиска в таком наборе данных применяется метод **последовательного перебора**: все элементы, начиная с первого, просматриваются подряд. Поиск завершается в двух случаях:

- 1) искомый элемент найден, при этом может быть просмотрена как часть имеющегося набора данных, так и весь набор, если искомый элемент оказался последним в наборе;
- 2) просмотрены все элементы имеющегося набора данных, но искомого элемента среди них не оказалось.

Длительность поиска методом последовательного перебора определяется как $N/2$, где N — размер набора данных. Действительно, искомый элемент может оказаться первым среди просматриваемых, и в этом случае длительность поиска равна 1. Если искомый элемент окажется последним или его не окажется вообще, то длительность поиска будет равна N . Если провести поиск последовательным перебором достаточно много раз, то окажется, что в среднем на поиск требуемого элемента уходит $N/2$ просмотров.

Гораздо проще осуществлять поиск в структурированном наборе данных. Структурирование связано с внесением определённого порядка, определённой организации: расположения данных в алфавитном порядке, группировки по некоторым признакам и т. д. Если информация структурирована, то поиск осуществляется быстрее, можно построить оптимальный алгоритм.

Ранее мы уже рассматривали метод **половинного деления**. Он применяется к наборам данных, элементы которых упорядочены по неубыванию, т. е. каждый последующий элемент не меньше (больше или равен) предыдущего:

$$a_1 \leq a_2 \leq a_3 \leq \dots \leq a_N.$$

Искомый элемент сравнивается с центральным элементом последовательности, номер которого находится как $[N/2] + 1$. Квадратные скобки здесь обозначают, что от результата деления берётся только целая часть, а дробная часть отбрасывается.

Если искомый элемент больше центрального, то поиск продолжается в правой части последовательности. Если искомый элемент меньше центрального, то — в левой. Если значения искомого элемента и центрального совпадают, то поиск завершается.

Рассмотрим работу этого алгоритма поиска информации на примере.

Пример 6. В последовательности чисел
061 087 154 180 208 230 290 345 367 389 456 478 523 567 590 612
требуется найти число 180.

Просмотр 1. Работаем со всей последовательностью. Определяем центральный элемент (он подчеркнут):

061 087 154 180 208 230 290 345 367 389 456 478 523 567 590 612

Сравниваем искомый элемент с центральным.

По результатам сравнения отбрасываем правую часть последовательности.

Просмотр 2. Работаем с левой частью последовательности. Определяем центральный элемент (он подчеркнут):

061 087 154 180 208 230 290 345

Сравниваем искомый элемент с центральным.

По результатам сравнения отбрасываем правую часть последовательности.

Просмотр 3. Работаем с левой частью последовательности. Определяем центральный элемент (он подчеркнут):

061 087 154 180

Сравниваем искомый элемент с центральным.

По результатам сравнения отбрасываем левую часть последовательности.

Просмотр 4. Работаем с правой частью последовательности. Определяем центральный элемент (он подчеркнут):

180

Центральный элемент совпадает с искомым. Поиск завершён.

Как связаны длительность поиска методом половинного деления и длина исходной последовательности данных?

САМОЕ ГЛАВНОЕ

Обработка информации — это целенаправленный процесс изменения содержания или формы представления информации. Существует два различных типа обработки информации:

- 1) обработка, связанная с получением нового содержания, новой информации;
- 2) обработка, связанная с изменением формы представления информации, не изменяющая её содержания.

Кодирование — это обработка информации, заключающаяся в её преобразовании в некоторую форму, удобную для хранения, передачи, обработки информации в дальнейшем.

Код — это система (список) условных обозначений (кодовых слов), используемых для представления информации.

Префиксный код — код со словом переменной длины, обладающий тем свойством, что никакое его кодовое слово не может быть началом другого (более длинного) кодового слова. Сообщение, закодированное с помощью префиксного кода, может быть однозначно декодировано.

Задача поиска информации состоит в том, чтобы в некотором хранилище информации (информационном массиве) найти информацию, удовлетворяющую определённым условиям поиска.

Время поиска зависит от способа организации набора данных и используемого алгоритма поиска.

Для осуществления поиска в неструктурированном наборе данных применяется метод последовательного перебора.

Поиск информации в упорядоченном по неубыванию наборе данных может быть осуществлён методом половинного деления.

Вопросы и задания



1. Приведите примеры процессов обработки информации, которые чаще всего вам приходится выполнять в жизни. Для каждого примера определите исходные данные, алгоритм (правила) обработки и получаемые результаты. К каким типам обработки информации относятся эти процессы?
2. Поясните суть понятий «кодирование», «код», «кодовая таблица».
3. Светодиодная панель содержит шесть излучающих элементов, каждый из которых может светиться или красным, или жёлтым, или зелёным цветом. Сколько различных сигналов можно передать с помощью панели (все излучающие элементы должны гореть, порядок цветов имеет значение)?
4. Автомобильный номер состоит из нескольких букв (количество букв одинаковое во всех номерах), за которыми следуют три цифры. При этом используются 10 цифр и только 5 букв: А, В, С, D и F. Требуется не менее 100 тысяч различных номеров. Какое наименьшее количество букв должно быть в автомобильном номере?



5. Сколько существует различных последовательностей из 6 символов четырёхбуквенного алфавита {A, B, C, D}, которые содержат не менее двух букв A (т. е. две и более буквы A)?
6. Сравните равномерные и неравномерные коды. Каковы их основные достоинства и недостатки?
7. Какие коды называют префиксными? Почему они так важны? В чём суть прямого и обратного условий Фано?
8. Двоичные коды для 5 букв латинского алфавита представлены в таблице:

A	B	C	D	E
000	01	10	11	001

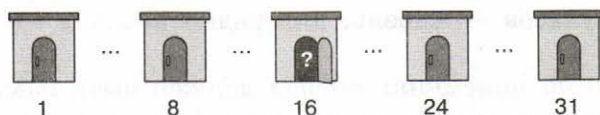
Из четырёх сообщений, закодированных этими кодами, только одно пришло без ошибки. Найдите его:

- 1) 110100000100110011;
- 2) 111010000010010011;
- 3) 110100001001100111;
- 4) 110110000100110010.

9. Для кодирования некоторой последовательности, состоящей из букв A, B, B, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. При этом используются следующие коды: A — 1110, B — 0, B — 10, Г — 110. Каким кодовым словом может быть закодирована буква Д? Код должен удовлетворять свойству однозначного декодирования. Если можно использовать более одного кодового слова, укажите кратчайшее из них.
10. Для кодирования некоторой последовательности, состоящей из букв A, B, B, Г и Д, используется неравномерный троичный код, позволяющий однозначно декодировать полученную троичную последовательность. Вот этот код: A — 0, B — 11, B — 20, Г — 21, Д — 22. Можно ли сократить для одной из букв длину кодового слова так, чтобы закодированную последовательность по-прежнему можно было декодировать однозначно? Коды остальных букв меняться не должны.

11. Для передачи закодированных сообщений используется таблица кодовых слов из четырёх букв. Причем используются только буквы А, Р и У. Сколько различных кодовых слов может быть в такой таблице, если ни в одном слове нет трёх одинаковых букв, идущих подряд?
12. Методом половинного деления в последовательности чисел 061 087 154 180 208 230 290 345 367 389 456 478 523 567 590 612 требуется найти число 590. Опишите процесс поиска.
13. В Международном конкурсе по информатике «Бобёр» школьникам была предложена задача «Склад», подготовленная специалистами из Японии. Вот её условие.

Плотник в Бобровой Деревне использует 31 склад, пронумерованный от 1 до 31. Однажды он забыл, сколько складов уже заполнил, но помнит, что заполнял их в порядке возрастания номеров.



Чтобы уменьшить количество открывания дверей, он действует следующим образом:

Сначала открывает склад со средним номером — склад № 16.

Затем:

- если склад № 16 пуст, он решает искать первый незаполненный склад в промежутке от № 1 до № 15, открывает опять средний склад — склад № 8 — и повторяет процедуру;
- если склад № 16 заполнен, то нужный склад он ищет между № 17 и № 31, открывает средний склад — склад № 24 — и повторяет процедуру.

После всех действий плотник обнаружил, что заполнены были склады от № 1 до № 15 включительно. Сколько дверей ему пришлось открыть?

Решите эту задачу. Какой из рассмотренных нами методов поиска был использован героем этой задачи?

§ 5

Передача и хранение информации

5.1. Передача информации

Передача информации — один из самых распространённых информационных процессов. Из курса информатики основной школы вам известно, что процесс передачи информации происходит от источника к приёмнику по информационным каналам связи. При этом можно выделить следующие шаги:

- 1) передаваемая информация кодируется — представляется в форме некоторой последовательности сигналов, символов, знаков;
- 2) информация переносится на носитель, допускающий транспортировку на расстояние (бумага, электрический импульс, радиосигнал и др.);
- 3) используются свойства носителя, позволяющие ему преодолеть расстояние, отделяющее источник от приёмника (для бумажных писем — автомобиль, поезд, самолёт; для электрических импульсов — кабель, для радиосигналов — радиоволны и т. д.);
- 4) дошедший до приёмника сигнал должен быть извлечён из канала связи и перенесён на доступный для обработки носитель;
- 5) закодированная информация должна быть расшифрована и преобразована в форму, доступную для восприятия с помощью органов чувств человека.

Рассмотрим более подробно технические системы передачи информации, в которых для передачи информации используются технические средства связи (телефон, радио, телевидение и Интернет).

Вспомните, с какими открытиями и изобретениями в области передачи информации связаны имена Александра Белла, Генриха Герца, Александра Степановича Попова и Гульельмо Маркони.

На рисунке 1.13 представлена схема передачи информации по техническим каналам связи, предложенная Клодом Шенноном.

Поясните представленные на схеме (рис. 1.13) процессы на примере разговора по мобильному телефону. Используйте описанные выше шаги процесса передачи информации.





Рис. 1.13. Схема передачи информации по техническим каналам связи

Разного рода помехи, искажающие передаваемый сигнал и приводящие к потере информации, называют шумом. Если по каналу передается аналоговый сигнал (например, разговор двух друзей в вагоне метро), то при небольших шумах слушателю всё же удастся понять содержание сообщения благодаря избыточности, существующей у любого естественного языка. Для систем дискретной цифровой связи потеря даже одного бита (если не используется избыточный код) может привести к полному обесцениванию информации.



Большой вклад в развитие теории связи внёс выдающийся советский и российский учёный Владимир Александрович Котельников (1908–2005), заслуги которого признаны во всём мире. Его исследования посвящены проблемам совершенствования методов радиоприёма, изучению радиопомех и разработке методов борьбы с ними.

В современных технических системах связи борьба с шумом (защита от шума) осуществляется по следующим двум направлениям.

1. Устранение технических помех, связанных с плохим качеством линий связи, незащищённостью друг от друга различных потоков информации, передаваемых по одним и тем же каналам. Для устранения таких помех используют экранированные кабели, применяют различные фильтры, отделяющие полезный сигнал от шума и т. д.
2. Избыточное кодирование самого передаваемого сообщения, позволяющее компенсировать потерю какой-то части передаваемой по линиям связи информации.



Избыточность кода — это многократное повторение передаваемых данных.

Но чрезмерная избыточность приводит к задержкам и удорожанию связи. Поэтому очень важно иметь алгоритмы получения оптимального кода, одновременно обеспечивающего минимальную избыточность передаваемой информации и максимальную достоверность принятой информации.

В современных системах цифровой связи для борьбы с потерей информации часто применяется следующий приём. Всё сообщение разбивается на порции — блоки. Для каждого блока вычисляется контрольная сумма, которая передаётся вместе с данным блоком. В месте приёма заново вычисляется контрольная сумма принятого блока, и если она не совпадает с первоначальной, то передача данного блока повторяется.

Важной характеристикой современных технических каналов передачи информации является их **пропускная способность** — максимально возможная скорость передачи информации, измеряемая в битах в секунду (бит/с). Пропускная способность канала связи зависит от свойств используемых носителей (электрический ток, радиоволны, свет). Так, каналы связи, использующие оптоволоконные кабели и радиосвязь, обладают пропускной способностью, в тысячи раз превышающей пропускную способность телефонных линий.

Скорость передачи информации по тому или иному каналу зависит от пропускной способности канала, а также от длины закодированного сообщения, определяемой выбранным алгоритмом кодирования информации.

Современные технические каналы связи обладают целым рядом достоинств:

- **высокая пропускная способность**, обеспечиваемая свойствами используемых носителей;
- **надёжность**, связанная с использованием параллельных каналов связи;
- **помехозащищённость**, основанная на автоматических системах проверки целостности переданной информации;
- **универсальность** используемого двоичного кода, позволяющего передавать любую информацию — текст, изображение, звук.

Объём переданной информации I вычисляется по формуле:

$$I = v \cdot t,$$

где v — скорость передачи информации (в битах в секунду), а t — время передачи.

Рассмотрим несколько примеров решения задач, имеющих отношение к процессу передачи информации.

Для более наглядного представления условий некоторых задач, связанных с процессами передачи информации, удобно рисовать диаграмму, изображающую два и более процесса с разметкой временных отметок их начал и окончаний. Примером такой диаграммы является диаграмма Ганта.

Диаграмма Ганта представляет собой размещённые вдоль горизонтальной шкалы времени отрезки (прямоугольные полоски), каждый из которых соответствует отдельному процессу или задаче. Начало, конец и длина каждого такого отрезка соответствуют началу, концу и длительности того или иного процесса, а сами отрезки располагаются друг под другом со сдвигом по горизонтали.

Для наглядного представления последовательности реализации (синхронизации) нескольких процессов можно использовать диаграммы процессов или диаграммы Ганта.

Пример 1. У Толи есть доступ к сети Интернет по высокоскоростному одностороннему радиоканалу, обеспечивающему скорость получения информации 2^{20} бит/с. У Миши нет скоростного доступа в Интернет, но есть возможность получать информацию от Толи по низкоскоростному телефонному каналу со средней скоростью 2^{13} бит/с. Миша договорился с Толей, что тот будет скачивать для него данные объёмом 5 Мбайт по высокоскоростному каналу и ретранслировать их Мише по низкоскоростному каналу. Компьютер Толи может начать ретрансляцию данных не раньше, чем им будут получены первые 0,5 Мбайт этих данных.

Каков минимально возможный промежуток времени (в секундах) с момента завершения скачивания данных Толей до полного их получения Мишей?

Как скоро Миша сможет полностью получить все данные, если Толя начнёт их скачивание в 16:00?



В этой задаче мы имеем дело с двумя процессами передачи информации, осуществляемыми с разной скоростью.

1. Процесс скачивания информации Толей по высокоскоростному каналу. Длительность этого процесса $5 \cdot 2^{23}/2^{20} = 5 \cdot 2^3 = 40$ с.
2. Процесс скачивания информации Мишей по низкоскоростному каналу. Длительность этого процесса $5 \cdot 2^{23}/2^{13} = 5 \cdot 2^{10} = 5120$ с.

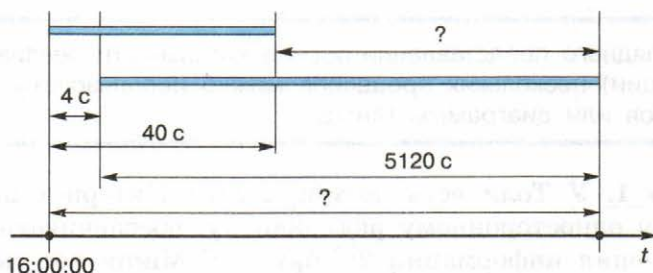
При решении задач необходимо согласовывать размерности величин.

Так, если скорость передачи информации задана в битах в секунду, то и значения объёмов информации следует выразить в битах.

Из условия задачи следует, что второй процесс начинается спустя некоторое время после начала первого процесса. Вычислим это время:

$$0,5 \cdot 2^{23}/2^{20} = 0,5 \cdot 2^3 = 4 \text{ с.}$$

На диаграмме Гантта результаты проведённых расчётов можно изобразить так:



На диаграмме видно, что для ответа на первый вопрос надо найти разность:

$$5120 - (40 - 4) = 5120 - 36 = 5084 \text{ с.}$$

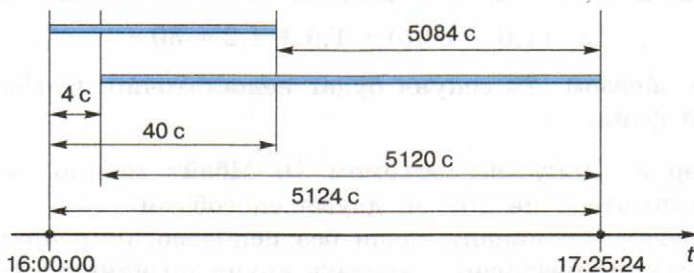
Ответ на второй вопрос получим, выполнив следующие вычисления:

$$5120 + 4 = 5124 \text{ с,}$$

$$5124 \text{ с} = 85 \text{ мин } 24 \text{ с} = 1 \text{ ч } 25 \text{ мин } 24 \text{ с.}$$

Полностью получить все данные Миша сможет не ранее чем в 17:25:24.

Диаграмму Гантта можно дополнить полученными результатами:



Пример 2. У вас есть высокоскоростной спутниковый доступ в Интернет, обеспечивающий получение данных от сервера со скоростью 4 000 000 бит/с. Запросы с вашего компьютера передаются на сервер со скоростью 128 000 бит/с через подключённый к компьютеру сотовый телефон, выполняющий функции модема. Вам необходимо скачать файл с музыкальной записью объёмом 12 Мбайт. Информация по спутниковому каналу передаётся с сервера пакетами, объём которых не превышает 5 Мбайт. При этом для получения каждого пакета ваш компьютер сначала должен передать в сеть запрос объёмом в 25 Кбайт.

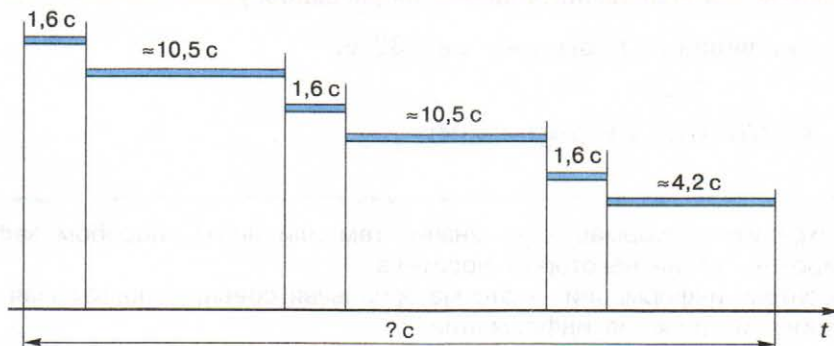
Сможете ли вы скачать требуемый файл за 25 с?

Файл объёмом 12 Мбайт будет передаваться пакетами, не превышающими 5 Мбайт; следовательно, всего пакетов будет три — два по 5 Мбайт и один — 2 Мбайт.

Процессы передачи каждого из двух пакетов по 5 Мбайт займут по $5 \cdot 2^{23} / 4000000 \approx 10,5$ с. На передачу 2 Мбайт потребуется $\approx 4,2$ с.

Приёму каждого пакета будет предшествовать передача запроса объёмом 25 Кбайт. Каждый раз этот процесс будет осуществляться за $25 \cdot 2^{13} / 128000 = 25 \cdot 2^{13} / (2^{10} \cdot 125) = 1,6$ с.

Изобразим имеющуюся информацию на диаграмме Гантта:



Вычислим общее время, требуемое для скачивания файла:

$$2 \cdot (1,6 + 10,5) + 1,6 + 4,2 = 30 \text{ с.}$$

Таким образом, 25 секунд будет недостаточно, чтобы скачать требуемый файл.

Пример 3. Документ объёмом 10 Мбайт можно передать с одного компьютера на другой двумя способами.

А. Передать по каналу связи без использования архиватора.

Б. Сжать архиватором, передать архив по каналу связи, распаковать.

Какой способ быстрее и насколько, если:

- средняя скорость передачи данных по каналу связи составляет 2^{18} бит/с;
- объём сжатого архиватором документа равен 25% от исходного объёма;
- время, требуемое на сжатие документа, — 5 секунд, на распаковку — 3 секунды?

Для решения данной задачи диаграмма Ганта не нужна; достаточно выполнить расчёты для каждого из имеющихся вариантов передачи информации.

Рассмотрим вариант А. Длительность передачи информации в этом случае составит: $10 \cdot 2^{23} / 2^{18} = 10 \cdot 2^5 = 320 \text{ с.}$

Рассмотрим вариант Б. Длительность передачи информации в этом случае составит:

$$5 + \frac{0,25 \cdot 10 \cdot 2^{23}}{2^{18}} + 3 = 5 + 80 + 3 = 88 \text{ с.}$$

↑
↑
↑

Время на сжатие Время на передачу в упакованном виде Время на распаковку

Итак, вариант Б быстрее на 232 с.

5.2. Хранение информации

Сохранить информацию — значит тем или иным способом зафиксировать её на некотором носителе.

Носитель информации — это материальная среда, используемая для записи и хранения информации.

Основным носителем информации для человека является его собственная память. По отношению к человеку все прочие виды носителей информации можно назвать внешними.

Основное свойство человеческой памяти — быстрота, оперативность воспроизведения хранящейся в ней информации. Но наша память не надёжна: человеку свойственно забывать информацию. Хотя психологи утверждают, что из памяти человека ничего не исчезает, тем не менее человек довольно часто теряет способность к воспроизведению некоторых знаний. Именно для более надёжного хранения информации человек использует внешние носители, организует внешние хранилища информации.

Виды внешних носителей менялись со временем: в древности это были камень, дерево, папирус, кожа и др. Долгие годы основным носителем информации была бумага. Развитие компьютерной техники привело к созданию магнитных (магнитная лента, гибкий магнитный диск, жёсткий магнитный диск), оптических (CD, DVD, BD) и других современных носителей информации.

В настоящее время наиболее надёжными носителями информации, записанной цифровым способом, являются оптические диски.

Применение оптического способа записи началось в 1980-х годах. Тогда и появились оптические диски, иначе называемые компакт-дисками (CD). Процесс записи и считывания информации таких дисков осуществляется при помощи лазера. Информационная ёмкость CD составляет от 190 до 700 МБ.

Во второй половине 1990-х годов появились цифровые многоцелевые диски DVD, информационная ёмкость которых измерялась уже в гигабайтах (до 17 GB). DVD имеют такой же размер, как и CD, но для записи и считывания данных в них применяется лазер с меньшей длиной волны, что обеспечивает более плотную структуру рабочей поверхности, позволяя хранить и считывать больший объём информации.

В конце 2000-го года впервые был представлен Blu-ray Disc (BD) — оптический носитель, используемый для записи с повышенной плотностью хранения цифровых данных, включая видео высокой чёткости. Коммерческий запуск BD, положивший начало его широкому распространению, был осуществлён в 2006 году. В BD для записи и чтения данных используется коротковолновый (405 нм) сине-фиолетовый лазер. Это позволяет при сохранении физических размеров CD и DVD (12 см) увеличить информационную ёмкость BD до 50 GB и более.



Какая основная тенденция прослеживается в развитии оптических носителей информации? Сформулируйте её и попробуйте обосновать.

В последние годы появились и получили широкое распространение всевозможные мобильные электронные (цифровые) устройства: планшетные компьютеры, смартфоны, устройства для чтения электронных книг, GPS-навигаторы и др. Появление таких устройств стало возможно, в том числе, благодаря разработке принципиально новых носителей информации, которые:

- обладают большой информационной ёмкостью при небольших физических размерах;
- характеризуются низким энергопотреблением при работе, обеспечивая наряду с этим высокие скорости записи и чтения данных;
- энергонезависимы при хранении;
- имеют долгий срок службы.

Всеми этими качествами обладает флеш-память (англ. *flash-memory*). Выпуск построенных на их основе флеш-накопителей, называемых в просторечии «флешками», был начат в 2000 году. Сегодня широко используются флеш-накопители от 8 GB до 128 GB.



В настоящее время активно ведутся работы по созданию ещё более компактных носителей информации на основе нанотехнологий, имеющих дело с молекулами и атомами вещества. По предположениям экспертов приблизительно через 15–20 лет плотность хранения информации возрастёт настолько, что на носитель с физическими размерами в 1 см³ можно будет записать каждую секунду человеческой жизни.

Современные системы хранения информации, центральным звеном которых являются носители информации или запоминающие устройства, включают в себя также инструменты поиска (поисковые системы) и средства отображения информации (устройства вывода). Такие системы хранения информации определённого назначения (определённой предметной области) представляют собой базы данных, банки данных и базы знаний.



В этом параграфе при разговоре об информационной ёмкости носителей мы использовали обозначения MB и GB для таких единиц, как мегабайт и гигабайт. Не показалось ли вам это необычным, тем более что в предыдущем параграфе мы говорили об обозначениях Мбайт и Гбайт?

Дело в том, что многие современные операционные системы продолжают придерживаться принципа, что 1 килобайт равен 1024 байт, в то время как производители запоминающих устройств считают его равным 1000 байт. Если на CD или другом носителе информации вы видите надпись, например, 700 MB, то понимать приставку «мега» здесь следует в традиционном математическом смысле, а именно:

$$700 \text{ MB} = 700 \cdot 10^3 \text{ KB} = 700 \cdot 10^3 \cdot 10^3 \text{ байт} = 700\,000\,000 \text{ байт.}$$

САМОЕ ГЛАВНОЕ

Процесс передачи информации происходит от источника к приёмнику по информационным каналам связи.

Современные технические каналы связи обладают перед ранее известными целым рядом достоинств:

- высокая пропускная способность — максимально возможная скорость передачи информации, измеряемая в битах в секунду;
- надёжность, связанная с использованием параллельных каналов связи;
- помехозащищённость, основанная на автоматических системах проверки целостности переданной информации;
- универсальность используемого двоичного кода, позволяющего передавать любую информацию — текст, изображение, звук.

Объём переданной информации I вычисляется по формуле $I = v \cdot t$, где v — скорость передачи информации (в битах в секунду), а t — время передачи.

Сохранить информацию — значит тем или иным способом зафиксировать её на некотором носителе.

Носитель информации — это материальная среда, используемая для записи и хранения информации. Современные носители информации обладают большой информационной ёмкостью при небольших физических размерах; характеризуются низким энергопотреблением при работе, обеспечивая наряду с этим высокие скорости записи и чтения данных; энергонезависимы при хранении и имеют долгий срок службы.

Вопросы и задания



1. Опишите схему передачи информации по техническим каналам связи. Укажите компоненты этой схемы в процессе передачи информации при использовании сотовой связи.
2. Какие существуют способы борьбы с шумом в процессе передачи информации?

3. Как вычисляется объём информации, переданной по каналу связи?
4. Охарактеризуйте современные каналы связи. Какими достоинствами они обладают?
5. Скорость передачи информации по некоторому каналу связи составляет 256 000 бит/с. Передача файла через это соединение заняла 2 минуты. Определите размер переданного файла в килобайтах.
6. Для чего используются диаграммы Ганта? Как они строятся? При решении каких жизненных задач вы можете их применить?
7. Данные объёмом 100 Мбайт передаются из пункта А в пункт В по каналу связи, обеспечивающему скорость передачи данных 2^{20} бит/с, а затем из пункта В в пункт В по каналу связи, обеспечивающему скорость передачи данных 2^{22} бит/с. Задержка в пункте В (время между окончанием приема данных из пункта А и началом передачи в пункт В) составляет 24 секунды. Сколько времени (в секундах) прошло с момента начала передачи данных из пункта А до их полного получения в пункте В?
8. Документ (без упаковки) можно передать по каналу связи с одного компьютера на другой за 40 с. Если предварительно упаковать документ архиватором, передать упакованный документ, а потом распаковать на компьютере получателя, то общее время передачи (включая упаковку и распаковку) составит 20 с. Размер упакованного документа составляет 20% размера исходного документа. Сколько времени (в секундах) ушло на упаковку данных, если известно, что на их распаковку времени потребовалось в два раза больше?
9. Лена скачивает дистрибутив ОС Linux с зарубежного сайта-репозитория, пользуясь односторонним каналом цифровой передачи данных через телевизионное эфирное вещание, обеспечивающим приём информации со скоростью $4 \cdot 2^{23}$ бит/с. При этом информация передаётся фрагментами по 10 Мбайт. Для начала передачи каждого фрагмента компьютер Лены должен отправить на сервер сообщение-запрос объёмом 32 Кбайт, а после получения фрагмента подтвердить его безошибочный приём отдельным сообщением объёмом 16 Кбайт. Для отправки таких сообщений Лена пользуется радиомодемом GPRS, который обеспечивает скорость передачи информации до $128 \cdot 2^{13}$ бит/с. Определите

минимально возможное время (в секундах), за которое Лена сможет скачать файл дистрибутива объёмом 350 Мбайт.

- *10. Ровно в 12:00 папа поставил на скачивание файл и определил, что загрузка займёт 20 минут. Когда файл папы был скачан на 20%, мама поставила на скачивание свой файл. Через 6 минут после мамы Коля поставил на скачивание свой файл. При скачивании двух файлов скорость скачивания каждого в два раза ниже первоначальной, при скачивании трёх файлов скорость скачивания каждого в три раза ниже первоначальной, при скачивании одного — равна первоначальной. В какое время закончит скачивание своего файла каждый из членов этой семьи, если объёмы всех скачиваемых файлов равны?¹⁾
- *11. Для передачи помехоустойчивых сообщений, алфавит которых содержит 16 различных символов, используется равномерный двоичный код. Этот код удовлетворяет следующему свойству: в любом кодовом слове содержится чётное количество единиц (возможно, ни одной). Какую наименьшую длину может иметь кодовое слово?
12. Какое значение имеет хранение информации для всего человечества? Для отдельного человека?
13. Когда была изобретена магнитная запись? Подготовьте небольшое сообщение об этом изобретении и о магнитных носителях информации.
14. Подготовьте небольшое сообщение о физической природе оптического способа записи информации.
15. Вычислите объём учебника информатики в знаках. Выясните, сколько учебников такого же объёма можно записать на CD ёмкостью 700 МВ при условии использования восьмиразрядного компьютерного кода. Представьте, помещение каких размеров потребуется для складирования такого количества учебников.

Дополнительные материалы к главе смотрите в авторской мастерской.

¹⁾ По материалам конкурса «Кит — компьютеры, информатика, технологии» (konkurskit.org).

Глава 2

КОМПЬЮТЕР И ЕГО ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

§ 6

История развития вычислительной техники

6.1. Этапы информационных преобразований в обществе

Веками люди совершенствовали способы и методы передачи, накопления, обработки и хранения информации. При этом средства обработки информации, как и сама информация, видоизменялись и расширяли свои возможности настолько, что это приводило к переменам в общественных отношениях. Появился термин «информационная революция», которым принято обозначать кардинальные изменения инструментальной основы, способа передачи и хранения информации, а также объема информации, доступной активной части населения.



Информационная революция — кардинальное изменение инструментальной основы, способов передачи и хранения информации, а также объема информации, доступной активной части населения.

Принято выделять пять информационных революций, определяющих, по сути, пять этапов информационных преобразований в обществе (табл. 2.1).

Содержание первой информационной революции составляет распространение и внедрение в деятельность и сознание человека языка. Вторая информационная революция была связана с изобретением письменности. Сущность третьей информационной революции состоит в изобретении книгопечатания, сделавшего любую информацию, и особенно научные знания, продукцией массового потребления. Четвертая информационная революция состояла в применении электрической аппаратуры для скоростного

Таблица 2.1

Этапы информационных преобразований в обществе

Информационная революция	Содержание	Результат
Первая	Появление и распространение языка	Передача информации в пространстве и во времени с помощью устной речи. Запоминание информации
Вторая	Изобретение письменности	Принципиальное улучшение возможностей хранения информации. Накапливание знаний и их передача от поколения к поколению с помощью письменных документов
Третья (XV — середина XVI в.)	Изобретение книгопечатания как одной из первых информационных технологий	Не только сохранение информации, но и повышение её доступности и расширение сферы её распространения за счёт увеличения тиражей. Широкое распространение информации, научных знаний, информационной культуры
Четвёртая (конец XIX в.)	Открытие электричества и создание на этой основе средств коммуникации (телеграф, телефон, радио, телевизор)	Появление широкого спектра способов хранения информации. Оперативная передача и накапливание информации в достаточном объёме
Пятая (70-е гг. XX в.)	Изобретение микропроцессорной технологии и появление персонального компьютера	Создание систем накапливания и передачи данных, внедрение компьютерных сетей, применение компьютерных информационных технологий. Формирование личности с высоким уровнем информированности и информационной культуры

и массового распространения всех видов информации и знаний. Пятая, последняя, информационная революция связана с созданием сверхскоростных вычислительных устройств — компьютеров. С появлением и массовым распространением компьютеров человек впервые за всю историю развития цивилизации получил мощное средство для эффективного использования информационных ресурсов, для усиления своей интеллектуальной деятельности.

Возможность использования членами общества полной, своевременной и достоверной информации в значительной мере зависит от степени развития и освоения новых информационных технологий, основой которых является вычислительная техника.

6.2. История развития устройств для вычислений

В развитии устройств для вычислений можно выделить несколько этапов:

- «домеханический» этап — изобретение счётов, таблиц Непера и логарифмической линейки;
- «механический» этап, начало которого связывается с первыми попытками создания механической счётной машины, предпринятыми Леонардо да Винчи (1452–1519). К этому же этапу относится создание многочисленных вариантов счётных машин и арифмометров, а также проект аналитической машины Чарльза Беббиджа, которая должна была выполнять вычисления без участия человека;
- создание электрорелейных вычислительных машин (вычислительные машины Говарда Эйкена, Джона Атанасова, Конрада Цузе и др.);
- создание электронных вычислительных машин (ЭВМ):
 - создание ЭВМ фон-неймановской архитектуры;
 - отход от традиционной фон-неймановской архитектуры, использование процессоров, работающих параллельно.

Понятие «вычислительная техника» сегодня тесно связывается с компьютерами, которые до 80-х годов прошлого века у нас в стране называли электронными вычислительными машинами. В этом смысле в развитии вычислительной техники также можно выделить несколько этапов, связанных с возникновением разных поколений ЭВМ:

- 40-е — начало 50-х гг. XX в. (создание ЭВМ на электронных лампах);
- середина 50-х — 60-е гг. XX в. (разработка ЭВМ на дискретных полупроводниковых приборах);

- середина 60-х гг. XX в. — середина 70-х гг. XX в. (появление ЭВМ на интегральных микросхемах);
- середина 70-х гг. XX в. — наши дни (использование больших и сверхбольших интегральных схем).

Рассмотрим особенности и характеристики каждого из приведённых этапов более подробно.

6.3. Поколения ЭВМ

Первое поколение ЭВМ — ламповые машины 50-х годов. Скорость счёта самых быстрых машин первого поколения доходила до 20 тысяч операций в секунду. Для ввода программ и данных использовались перфоленты и перфокарты. Поскольку внутренняя память этих машин была невелика (могла вместить в себя несколько тысяч чисел и команд программы), они, главным образом, использовались для инженерных и научных расчётов, не связанных с переработкой больших объёмов данных. Это были довольно громоздкие сооружения, содержавшие в себе тысячи ламп, занимавшие иногда сотни квадратных метров, потреблявшие электроэнергию в сотни киловатт. Программы для таких машин составлялись на языках машинных команд, поэтому программирование в те времена было доступно немногим.

Первая ЭВМ ЭНИАК (ENIAC) была создана в конце 1945 г. в США; она весила 30 т и размещалась на 170 м². В нашей стране первая ЭВМ была создана в 1951 году. Называлась она МЭСМ — Малая Электронная Счётная Машина (рис. 2.1).

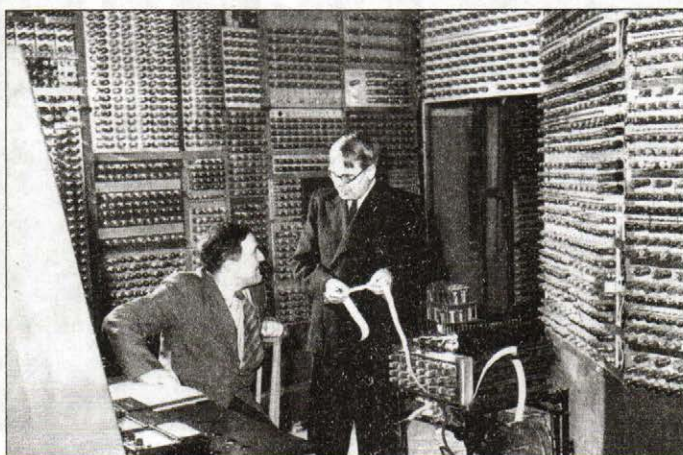


Рис. 2.1. ЭВМ первого поколения МЭСМ

К концу 40-х гг. XX в., когда вошли в строй первые большие электронные компьютеры, специалисты начали искать замену громоздким и хрупким, часто выходящим из строя лампам, на которых они были построены. В 1948 году в США был создан первый полупроводниковый прибор, заменяющий электронную лампу. Он получил название транзистор. В 60-х годах транзисторы стали элементной базой для ЭВМ второго поколения. Переход на полупроводниковые элементы улучшил качество ЭВМ по всем параметрам: они стали компактнее, надёжнее, менее энергоёмкими (рис. 2.2). Быстродействие большинства машин достигло нескольких сотен тысяч операций в секунду. Объём внутренней памяти возрос в сотни раз по сравнению с ЭВМ первого поколения. Большое развитие получили устройства внешней (магнитной) памяти: магнитные барабаны, накопители на магнитных лентах. Это способствовало созданию на ЭВМ информационно-справочных, поисковых систем, нуждающихся в длительном хранении больших объёмов информации. Во времена второго поколения ЭВМ активно начали развиваться языки программирования высокого уровня, одним из первых среди которых был Фортран (Fortran — сокращение от англ. *FORmula TRANslation* — трансляция формулы).

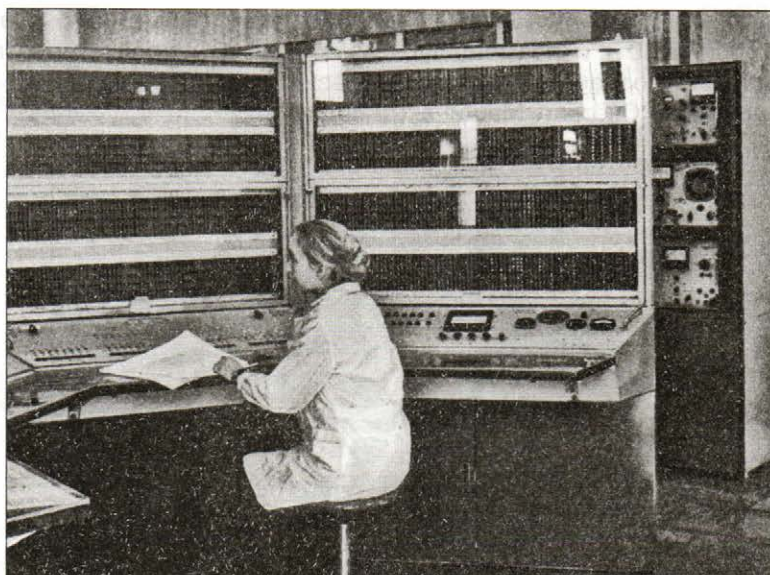


Рис. 2.2. ЭВМ второго поколения БЭСМ-6

Ключевыми фигурами среди физиков, занимавшихся изучением полупроводников, стали американские учёные Джон Бардин (1908–1991), Уолтер Браттейн (1902–1987), Уильям Брэдфорд Шокли (1910–1989). В 1948 году в газете «Нью-Йорк таймс» была напечатана короткая заметка, в которой сообщалось об изобретении ими нового устройства — транзистора. Эта информация прошла практически незамеченной, мало кто смог в то время оценить её по достоинству. Позже транзистор был признан одним из важнейших изобретений века, а его изобретатели получили Нобелевскую премию по физике.

Благодаря транзистору — германиевому кристаллу величиной с булавочную головку, заключённому в металлический цилиндр длиной около сантиметра, — электроника ступила на путь миниатюризации: один транзистор был способен заменить 40 электронных ламп.

Хотя транзистор был выдающимся научным изобретением, он не сразу получил широкое практическое применение в вычислительной технике. Германий, из которого изготавливали первые транзисторы, — довольно редкий химический элемент, поэтому стоимость транзисторов была очень высокой. Резко снизить стоимость транзисторов удалось только в середине 50-х гг. XX в.: в 1954 году был изготовлен первый транзистор из кремния — основного компонента обычного песка, — одного из самых распространённых на Земле химических элементов.

Третье поколение ЭВМ создавалось на новой элементной базе — сложные электронные схемы монтировались на маленькой пластине из полупроводникового материала, площадью менее 1 см². Их назвали интегральными схемами (ИС). Первые ИС содержали в себе десятки, затем — сотни элементов (транзисторов, сопротивлений и др.). Когда степень интеграции (количество элементов) приблизилась к тысяче, их стали называть большими интегральными схемами — БИС, а затем появились сверхбольшие интегральные схемы — СБИС. ЭВМ третьего поколения начали производить во второй половине 60-х годов, когда американская фирма IBM приступила к выпуску системы машин IBM-360. В Советском Союзе в 70-х годах начался выпуск машин серии ЕС ЭВМ (Единая Система ЭВМ). Скорость работы наиболее мощных моделей ЭВМ достигла нескольких миллионов операций в секунду. На машинах третьего поколения появился новый тип внешних запоминающих устройств — магнитные диски. В ЭВМ третьего поколения широко использовались новые типы устройств ввода-вывода: дисплеи, графопостроители (рис. 2.3).



Рис. 2.3. Рабочее помещение с установленной ЕС-1060

В этот период были созданы операционные системы (ОС), позволявшие управлять большим количеством внешних устройств и выполнять на одной машине несколько программ одновременно. Широкое распространение получили ранее созданные языки программирования. Начали появляться пакеты прикладных программ для решения задач в конкретных областях. Это существенно расширило области применения ЭВМ.

Первая интегральная схема, представлявшая собой кристалл, в котором была размещена целая схема из нескольких транзисторов, была разработана в 1958 г. американским физиком Джексом Килби, удостоенным за это изобретение Нобелевской премии.

Очередное революционное событие в электронике произошло в 1971 году, когда американская фирма Intel объявила о создании микропроцессора. Микропроцессор — это СБИС, способная выполнять функции основного блока компьютера — процессора. Соединив микропроцессор с устройствами ввода-вывода и внешней памяти, получили новый тип компьютера — микроЭВМ. МикроЭВМ относятся к машинам четвертого поколения. Существенным отличием микроЭВМ от своих предшественников являются их малые габариты (размеры бытового телевизора) и сравнительная дешевизна. Это первый тип компьютеров, который появился в розничной продаже.

Сегодня самой популярной разновидностью ЭВМ являются персональные компьютеры (ПК). Первый ПК был создан в 1976 году в США. С 1980 года и на долгие годы вперёд на рынке ПК ведущей становится американская фирма IBM. Её конструкторам удалось создать такую архитектуру, которая стала фактически международным стандартом на профессиональные ПК. Машины этой серии получили название IBM PC (Personal Computer). С точки зрения общественного развития появление и распространение ПК сопоставимы с появлением книгопечатания. Именно ПК сделали компьютерную грамотность массовым явлением. С развитием компьютеров этого типа появилось такое понятие, как информационные технологии, без которых уже невозможно обойтись в большинстве областей человеческой деятельности.

Ещё одной линией в развитии ЭВМ четвёртого поколения являются суперкомпьютеры — мощные многопроцессорные компьютеры, выполняющие параллельную обработку данных (рис. 2.4).

Все компьютеры, используемые в настоящее время, по-прежнему построены на базе идей четвёртого поколения.

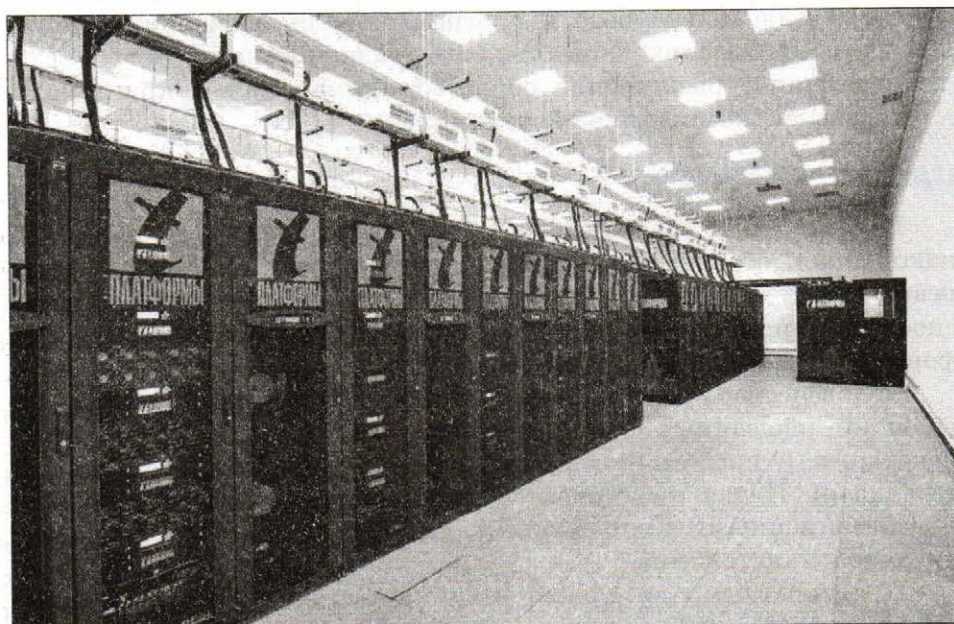


Рис. 2.4. Суперкомпьютер «Ломоносов»



В начале 90-х годов прошлого века в Японии начались работы по созданию компьютера пятого поколения. По замыслу японских специалистов основой работы этих компьютеров должны были стать не вычисления, а логические рассуждения, что означало переход от обработки данных к обработке знаний. Машину обещали научить воспринимать речь человека, рукописный текст и графические изображения. Окончательные результаты в этом направлении всё ещё не достигнуты. Исследования продолжаются.

Можно проследить несколько основных тенденций, имеющих место в развитии вычислительной техники:

- возрастание вычислительной мощности компьютеров от поколения к поколению;
- изменение целей использования компьютеров от сугубо военных и научно-технических расчётов к техническим и экономическим расчётам, коммуникационному и информационному обслуживанию, управлению;
- изменение в режиме работы компьютеров от однопрограммного к пакетной обработке, работе в режиме разделения времени, персональной работе и сетевой обработке данных;
- движение от машинного языка к языкам высокого уровня;
- повышение удобства работы пользователя за счёт усовершенствования аппаратного и программного обеспечения, возможности произвольного мобильного расположения;
- неуклонное расширение областей применения и круга пользователей компьютерной техники.

САМОЕ ГЛАВНОЕ

Веками люди совершенствовали способы и методы передачи, накопления, обработки и хранения информации. Информационная революция — кардинальное изменение инструментальной основы, способов передачи и хранения информации, а также объёма информации, доступной активной части населения.

Человечество прошло через несколько информационных революций, связанных с появлением речи, письменности, книгопечатания и средств коммуникации (телеграф, телефон, радио, телевизор). Пятая информационная революция связана с новыми информационными технологиями, основой которых является вычислительная техника.

Понятие «вычислительная техника» сегодня тесно связывает с компьютерами, которые до 80-х годов прошлого века у нас в стране называли электронными вычислительными машинами.

В развитии вычислительной техники также можно выделить несколько этапов, связанных с возникновением разных поколений ЭВМ:

- 1) 40-е — начало 50-х гг. XX в. (создание ЭВМ на электронных лампах);
- 2) середина 50-х — 60-е гг. XX в. (разработка ЭВМ на дискретных полупроводниковых приборах);
- 3) середина 60-х — середина 70-х гг. XX в. (появление ЭВМ на интегральных микросхемах);
- 4) середина 70-х гг. XX в. — наши дни (использование больших и сверхбольших интегральных схем).

Все компьютеры, используемые в настоящее время, по-прежнему построены на базе идей четвёртого поколения.

Вопросы и задания

1. Что понимают под информационными революциями? Какие информационные революции пережило человечество?
2. Выясните, когда отмечается День российской информатики. С чем связан выбор именно этой даты?
3. Дайте краткую характеристику «домеханического» периода развития вычислительных устройств, связанного с изобретением и использованием счётов, таблиц и логарифмической линейки.
4. Дайте краткую характеристику «механического» периода создания вычислительных устройств, связанного с именами таких изобретателей, как Леонардо да Винчи, Вильгельм Шиккард, Блез Паскаль, Готфрид Вильгельм Лейбниц, Филипп Маттеус Ган, Евна Якобсон и др.
5. Попытайтесь обнаружить «ткацкий след» в развитии вычислительной техники.
6. Подготовьте краткое сообщение об истории создания арифмометров, рассмотрев в нём изобретения Карла Томаса, Пафнутия Львовича Чебышёва, Вильгодта Теофиловича Однера.
7. По какому принципу ЭВМ делятся на поколения? Дайте краткую характеристику каждому поколению компьютеров.
8. Предложите классификацию современных персональных компьютеров. Изобразите её в виде графа.
9. Подготовьте небольшое сообщение о роли личности в развитии вычислительной техники. Героем сообщения может быть Стив



Джобс, Стив Возняк, Грейс Мюррэй Хоппер, Билл Гейтс или кто-то другой по вашему усмотрению.

10. Что такое суперкомпьютеры? Для решения каких задач они используются?
11. Какое место в рейтинге суперкомпьютеров (Топ500) занимают российские разработки?
12. Назовите основные тенденции, прослеживаемые в развитии вычислительной техники.

§ 7

Основополагающие принципы устройства ЭВМ

7.1. Принципы Неймана–Лебедева

В каждой области науки и техники существуют фундаментальные идеи или принципы, определяющие на многие годы вперёд её содержание и направление развития. В компьютерных науках роль таких фундаментальных идей сыграли принципы, сформулированные независимо друг от друга двумя крупнейшими учёными XX века — Джоном фон Нейманом и Сергеем Алексеевичем Лебедевым.

Принцип — основное, исходное положение какой-нибудь теории, учения, науки и пр.

Принципы Неймана–Лебедева — базовые принципы построения ЭВМ, сформулированные в середине прошлого века, не утратили свою актуальность и в наши дни.



Джон фон Нейман (1903–1957) — американский учёный, сделавший важный вклад в развитие целого ряда областей математики и физики. В 1946 г., анализируя сильные и слабые стороны ЭНИАКа, совместно с коллегами пришёл к идее нового типа организации ЭВМ.



Сергей Алексеевич Лебедев (1902–1974) — академик, основоположник вычислительной техники в СССР, главный конструктор первой отечественной электронной вычислительной машины МЭСМ, автор проектов компьютеров серии БЭСМ (Большая Электронная Счётная Машина), разработчик принципиальных положений суперкомпьютера «Эльбрус». В 1996 году посмертно награждён медалью «Пионер компьютерной техники» — самой престижной наградой международного компьютерного сообщества.

Рассмотрим сущность основных принципов Неймана–Лебедева:

- 1) состав основных компонентов вычислительной машины;
- 2) принцип двоичного кодирования;
- 3) принцип однородности памяти;
- 4) принцип адресности памяти;
- 5) принцип иерархической организации памяти;
- 6) принцип программного управления.

Первый принцип определяет **состав основных компонентов вычислительной машины**.

Любое устройство, способное производить автоматические вычисления, должно иметь определённый набор компонентов: блок обработки данных, блок управления, блок памяти и блоки ввода/вывода информации.

Функциональная схема такого компьютера, отражающая программное управление работой и взаимодействием его основных узлов, представлена на рисунке 2.5. Его информационным центром является процессор:

- все информационные потоки (тонкие стрелки на рисунке) проходят через процессор;
- управление всеми процессами (толстые стрелки на рисунке) также осуществляется процессором.

Такие блоки есть и у современных компьютеров. Это:

- **процессор**, состоящий из арифметико-логического устройства (АЛУ), выполняющего обработку данных, и устройства управления (УУ), обеспечивающего выполнение программы и организующего согласованное взаимодействие всех узлов компьютера;



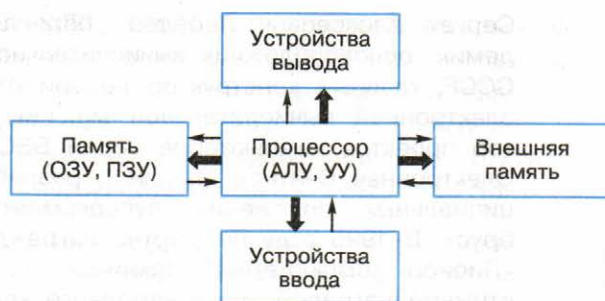


Рис. 2.5. Функциональная схема компьютеров первых поколений

- **память**, предназначенная для хранения исходных данных, промежуточных величин и результатов обработки информации, а также самой программы обработки информации. Различают память внутреннюю и внешнюю. Основная часть внутренней памяти используется для временного хранения программ и данных в процессе обработки. Такой вид памяти принято называть оперативным запоминающим устройством (ОЗУ). Ещё одним видом внутренней памяти является постоянное запоминающее устройство (ПЗУ), содержащее программу начальной загрузки компьютера. Внешняя или долговременная память предназначена для длительного хранения программ и данных в периоды между сеансами обработки;
- **устройства ввода**, преобразующие входную информацию в форму, доступную компьютеру;
- **устройства вывода**, преобразующие результаты работы компьютера в форму, доступную для восприятия человеком.

Вместе с тем в архитектуре современных компьютеров и компьютеров первых поколений есть существенные отличия. О них будет сказано чуть ниже.

Рассмотрим суть принципа двоичного кодирования информации.

Вся информация, предназначенная для обработки на компьютере (числа, тексты, звуки, графика, видео), а также программы её обработки представляются в виде двоичного кода — последовательностей 0 и 1.

Все современные компьютеры хранят и обрабатывают информацию в двоичном коде. Выбор двоичной системы счисления обусловлен рядом важных обстоятельств: простотой выполнения арифметических операций в двоичной системе счисления, её «согласованностью» с булевой логикой, простотой технической реализации двоичного элемента памяти (триггера).

Несмотря на всеобщее признание, использование в компьютерной технике классической двоичной системы счисления не лишено недостатков. В первую очередь это проблема представления отрицательных чисел, а также нулевая избыточность (т. е. отсутствие избыточности) двоичного представления. Пути преодоления указанных проблем были найдены уже на этапе зарождения компьютерной техники.

В 1958 г. в Московском государственном университете им. М. В. Ломоносова под руководством Н. П. Брусенцова был создан троичный компьютер «Сетунь» (рис. 2.6). В нём применялась уравновешенная троичная система счисления, использование которой впервые в истории позволило представлять одинаково просто как положительные, так и отрицательные числа.

Итак, благодаря двоичному кодированию, данные и программы по форме представления становятся одинаковыми, а следовательно, их можно хранить в единой памяти.

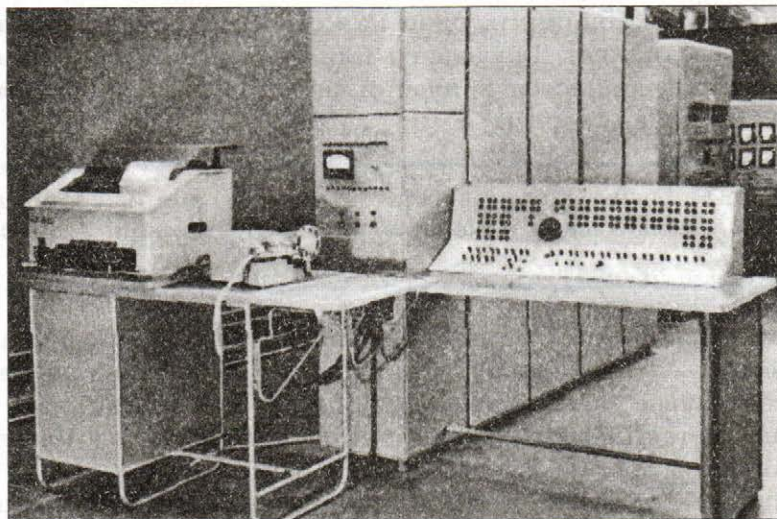


Рис. 2.6. ЭВМ «Сетунь»



Команды программ и данные хранятся в одной и той же памяти, и внешне в памяти они неразличимы. Распознать команды и данные можно только по способу использования. Это утверждение называют **принципом однородности памяти**.

Так как представленные в памяти команды и данные внешне неразличимы, то одно и то же значение в ячейке памяти может использоваться и как данные, и как команда в зависимости лишь от способа обращения к нему. Так, если к двоичной последовательности обращаются как к числу, то в ней выделяют поле (область) знака и поле значащих разрядов. Если к двоичной последовательности обращаются как к команде, то в ней выделяют поле кода операции и поле адресов операндов.

Однородность памяти позволяет производить операции не только над данными, но и над командами. Взяв в качестве данных для некоторой программы команды другой программы, в результате её исполнения можно получить команды третьей программы. Данная возможность лежит в основе трансляции — перевода текста программы с языка высокого уровня на язык конкретной вычислительной машины.

Структурно оперативная память компьютера состоит из отдельных битов — однородных элементов, обладающих двумя устойчивыми состояниями, одно из которых соответствует нулю, а другое — единице. Для записи или считывания группы соседних битов объединяются в ячейки памяти, каждая из которых имеет свой номер (адрес).



Команды и данные размещаются в единой памяти, состоящей из ячеек, имеющих свои номера (адреса). Это **принцип адресности памяти**.

Очень важно, что информация может считываться из ячеек и записываться в них в произвольном порядке, т. е. процессору в произвольный момент доступна любая ячейка памяти. Организованную таким образом память принято называть памятью с произвольным доступом.

Разрядность ячеек памяти (количество битов в ячейке) у компьютеров разных поколений была различной. Основой оперативной памяти современных компьютеров является восьмибитная

ячейка. Ячейка такой разрядности может быть использована для работы с одним символом. Для хранения чисел используется несколько последовательных ячеек (четыре — в случае 32-битного числа).

На современных компьютерах может одновременно извлекаться из памяти и одновременно обрабатываться до 64 разрядов (т. е. до восьми байтовых (восьмибитных) ячеек). Это возможно благодаря реализации на них принципа параллельной обработки данных — одновременного (параллельного) выполнения нескольких действий.

Можно выделить два основных требования, предъявляемых к памяти компьютера:

- 1) объём памяти должен быть как можно больше;
- 2) время доступа к памяти должно быть как можно меньше.

Создать запоминающее устройство, одновременно удовлетворяющее двум этим требованиям, затруднительно. Действительно, в памяти большого объёма требуемые данные искать сложнее, в результате чего их чтение замедляется. Для ускорения чтения нужно использовать более сложные технические решения, что неизбежно приводит к повышению стоимости всего компьютера. Решение проблемы — использование нескольких различных видов памяти, связанных друг с другом. В этом и состоит суть **принципа иерархической организации памяти**.

Трудности физической реализации запоминающего устройства высокого быстродействия и большого объёма требуют иерархической организации памяти.

В современных компьютерах используются устройства памяти нескольких уровней, различающиеся по своим основным характеристикам: времени доступа, сложности, объёму и стоимости. При этом более высокий уровень памяти меньше по объёму, быстрее и имеет большую стоимость в пересчёте на байт, чем более низкий уровень. Уровни иерархии взаимосвязаны: все данные на одном уровне могут быть также найдены на более низком уровне.

Большинство алгоритмов обращаются в каждый промежуток времени к небольшому набору данных, который может быть помещён в более быструю, но дорогостоящую и поэтому небольшую память. Использование более быстрой памяти увеличивает производительность вычислительного комплекса.



Главное отличие компьютеров от всех других технических устройств — это программное управление их работой.

Принцип программного управления определяет общий механизм автоматического выполнения программы.

Все вычисления, предусмотренные алгоритмом решения задачи, должны быть представлены в виде программы, состоящей из последовательности команд. Команды представляют собой закодированные управляющие слова, в которых указывается:

- какое выполнить действие;
- из каких ячеек считать операнды (данные, участвующие в операции);
- в какую ячейку записать результат операции.

Команды, входящие в программу, выполняются процессором автоматически в определённой последовательности. При этом выполняется следующий цикл действий:

- 1) чтение команды из памяти и её расшифровка;
- 2) формирование адреса очередной команды;
- 3) выполнение команды.

Этот цикл повторяется до достижения команды, означающей окончание выполнения программы, решающей некоторую конкретную задачу. В современных компьютерах по завершении работы программы управление передаётся операционной системе.

7.2. Архитектура персонального компьютера

Современные персональные компьютеры различаются по своим размерам, конструкции, разновидностям используемых микросхем и модулей памяти, другим характеристикам. В то же время все они имеют единое функциональное устройство, единую архитектуру — основные узлы и способы взаимодействия между ними (рис. 2.7).

Архитектура — это наиболее общие принципы построения компьютера, отражающие программное управление работой и взаимодействием его основных функциональных узлов.

На рисунке 2.7 изображены хорошо известные вам узлы современного компьютера: процессор, внутренняя память, устройства ввода, устройства вывода и внешняя память.

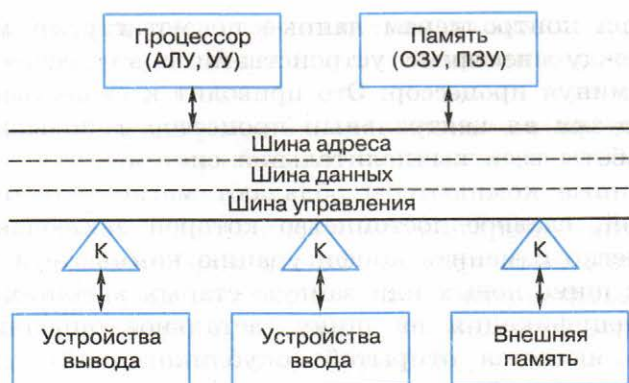


Рис. 2.7. Функциональная схема компьютера (К — контроллер)

Обмен данными между устройствами компьютера осуществляется с помощью магистрали.

Магистраль (шина) — устройство для обмена данными между устройствами компьютера.

Магистраль состоит из трёх линий связи:

- **шины адреса**, используемой для указания физического адреса, к которому устройство может обратиться для проведения операции чтения или записи;
- **шины данных**, предназначенной для передачи данных между узлами компьютера;
- **шины управления**, по которой передаются сигналы, управляющие обменом информацией между устройствами и синхронизирующие этот обмен.

В компьютерах, имевших классическую фон-неймановскую архитектуру, процессор контролировал все процессы ввода/вывода. При этом быстродействующий процессор затрачивал много времени на ожидание результатов работы от значительно более медленных внешних устройств. Для повышения эффективности работы процессора были созданы специальные электронные схемы, предназначенные для обслуживания устройств ввода/вывода или внешней памяти.

Контроллер — это специальный микропроцессор, предназначенный для управления внешними устройствами: накопителями, мониторами, принтерами и т. д.

Благодаря контроллерам данные по магистрали могут передаваться между внешними устройствами и внутренней памятью напрямую, минуя процессор. Это приводит к существенному снижению нагрузки на центральный процессор и повышает эффективность работы всей вычислительной системы.

Современные компьютеры обладают магистрально-модульной архитектурой, главное достоинство которой заключается в возможности легко изменить конфигурацию компьютера путём подключения к шине новых или замены старых внешних устройств.

Если спецификация на шину (детальное описание всех её параметров) является открытой (опубликованной), то производители могут разработать и предложить пользователям разнообразные дополнительные устройства для компьютеров с такой шиной. Подобный подход называют принципом открытой архитектуры. Благодаря ему пользователь может собрать именно такую компьютерную систему, которая ему нужна.

7.3. Перспективные направления развития компьютеров

Мир современных компьютеров необычайно разнообразен. Кроме микропроцессоров, встраиваемых во всевозможные устройства, и разных типов персональных компьютеров существуют значительно более мощные вычислительные системы.

Это серверы в глобальной компьютерной сети, управляющие её работой и хранящие огромные объёмы информации.

Это многопроцессорные системы параллельной обработки данных, обеспечивающие:

- сокращение времени решения вычислительно сложных задач;
- сокращение времени обработки больших объёмов данных;
- решение задач реального времени;
- создание систем высокой надёжности.

Время однопроцессорных вычислительных систем прошло. Не только суперкомпьютеры, но и современные персональные компьютеры, ноутбуки, игровые приставки основаны на многопроцессорных, многоядерных и других технологиях, предполагающих одновременное выполнение множества инструкций.

В наши дни электронная техника уже подошла к предельным значениям своих технических характеристик, которые определяются физическими законами. Поэтому идёт поиск неэлектрон-

ных средств хранения и обработки данных, ведутся работы по созданию квантовых и биологических компьютеров, проводятся исследования в области нанотехнологий.

САМОЕ ГЛАВНОЕ

В каждой области науки и техники существуют фундаментальные идеи или принципы, определяющие на многие годы вперёд её содержание и направление развития. В компьютерных науках роль таких фундаментальных идей сыграли принципы, сформулированные независимо друг от друга двумя крупнейшими учёными XX века — Джоном фон Нейманом и Сергеем Алексеевичем Лебедевым.

К основополагающим принципам построения компьютеров (принципам Неймана–Лебедева) можно отнести следующие:

- 1) состав основных компонентов вычислительной машины;
- 2) принцип двоичного кодирования;
- 3) принцип однородности памяти;
- 4) принцип адресности памяти;
- 5) принцип иерархической организации памяти;
- 6) принцип программного управления.

Архитектура — это наиболее общие принципы построения компьютера, отражающие программное управление работой и взаимодействием его основных функциональных узлов.

Классическая архитектура компьютеров первых поколений предполагала осуществление взаимодействия всех устройств через процессор и наличие неизменного набора внешних устройств.

Современные персональные компьютеры обладают открытой магистрально-модульной архитектурой — устройства взаимодействуют через шину, что способствует оптимизации процессов обмена информацией внутри компьютера. Второе преимущество современной архитектуры — возможность легко изменить конфигурацию компьютера путём подключения к шине новых или замены старых внешних устройств.

Вопросы и задания



1. Перечислите основные фундаментальные идеи, лежащие в основе построения компьютеров.
2. Какие устройства принято выделять в компьютерах классической архитектуры? Сравните их с устройством машины Беббиджа.

3. Чем обусловлен выбор двоичного кодирования для представления информации в компьютере?
4. Как вы понимаете утверждение «Одно и то же значение ячейки памяти в зависимости от способа обращения к нему может использоваться и как данные, и как команда»?
5. В чём состоит суть принципа адресности памяти?
6. Почему в современных компьютерах используются устройства памяти нескольких уровней, различающиеся по времени доступа, сложности, объёму и стоимости?
7. В чём состоит суть принципа программного управления?
8. Подготовьте сообщение о Джоне фон Неймане и его вкладе в развитие компьютерной техники.
9. Для чего предназначена магистраль (шина)? Из каких частей она состоит?
10. Что такое магистрально-модульная архитектура? В чём её главное достоинство?
11. В чём заключается принцип открытой архитектуры?

§ 8

Программное обеспечение компьютера

8.1. Структура программного обеспечения

Совокупность всех программ, предназначенных для выполнения на компьютере, называют **программным обеспечением** (ПО) компьютера.

Сфера применения конкретного компьютера определяется как его техническими характеристиками, так и установленным на нём ПО.

ПО современных компьютеров насчитывает тысячи программ. Оно непрерывно развивается — совершенствуются существующие программы, на смену одним программам приходят другие, появляются новые программы.

Тем не менее, отбросив второстепенные детали, всё многообразие компьютерных программ можно разделить на три груп-

пы: системное ПО, прикладное ПО, системы программирования (рис. 2.8).

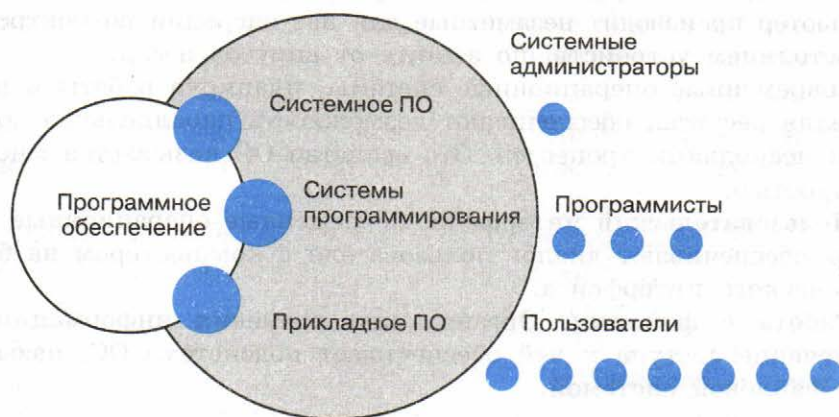


Рис. 2.8. Структура программного обеспечения

8.2. Системное программное обеспечение

Системное программное обеспечение включает в себя операционную систему и сервисные программы.

Операционная система — это комплекс программ, обеспечивающих согласованное функционирование всех устройств компьютера и предоставляющих пользователю доступ к ресурсам компьютера.

В настоящее время наиболее распространёнными ОС для персональных компьютеров являются Windows, Mac OS и Linux, для смартфонов, планшетов и других мобильных устройств — Android, iOS, Windows Phone.

Рассмотрим основные функции, выполняемые ОС современного компьютера.

Управление устройствами. Для обеспечения согласованного функционирования аппаратного обеспечения компьютера в состав ОС входят **драйверы** — специальные программы, управляющие работой подключённых к компьютеру внешних (периферийных) устройств. С помощью этих программ осуществляется контроль за нормальным функционированием оборудования, обеспечивается реакция на возникающие ошибки и аварийные ситуации.

Управление процессами. Программу, выполняемую на компьютере в текущий момент, принято называть процессом. Даже когда мы просто ищем информацию в сети Интернет, компьютер производит незаметные для нас операции по контролю за состоянием устройств, по защите от вирусов и т. д.

Современные операционные системы, планируя работы и распределяя ресурсы, обеспечивают возможность параллельной обработки нескольких процессов. Это свойство ОС называется многозадачностью.

Пользовательский интерфейс. Современные операционные системы обеспечивают диалог пользователя с компьютером на базе графического интерфейса.

Работа с файлами. Организацию хранения информации и обеспечение доступа к ней обеспечивает подсистема ОС, называемая файловой системой.

К сервисным программам или утилитам относят различные программы, выполняющие некоторые дополнительные услуги системного характера: обслуживание дисков (проверка, восстановление, очистка диска др.), архивирование файлов, защита от вирусов и многие другие.

Архиваторы — это специальные программы, осуществляющие сжатие программ и данных. Архиваторы обеспечивают уменьшение объёма хранимой информации, а значит, экономию места на диске и сокращение времени копирования этой информации, что особенно важно при пересылке информации по сети Интернет.

Многие программы сжатия данных построены на основе алгоритма Хаффмана, состоящего из двух этапов. На первом этапе читаются все входные данные и подсчитываются частоты встречаемости всех символов. Затем по этим данным строится дерево кодирования Хаффмана, а по нему — коды символов. На втором этапе входные данные читаются ещё раз, при этом генерируется выходной массив данных.

Попробуем сжать с помощью алгоритма Хаффмана фразу:

НА_ДВОРЕ_ТРАВА,_НА_ТРАВЕ_ДРОВА

Подсчитать частоты встречаемости символов достаточно просто:

Н	А	_	Д	В	О	Р	Е	Т	,
2	6	5	2	4	2	4	2	2	1

Алгоритм построения дерева Хаффмана следующий.

1. Считаем частоты встречаемости символов рассматриваемой фразы вершинами будущего графа (дерева). Выписываем их вертикально в ряд.
2. Выбираем две вершины с наименьшими весами (они соответствуют символам с наименьшим количеством повторений). Объединяем эти вершины — создаём новую вершину, от которой проводим рёбра к выбранным вершинам с наименьшими весами, а вес новой вершины задаём равным сумме их весов. Расставляем на рёбрах графа числа 0 и 1: на верхнем ребре — 0, а на нижнем — 1. Чтобы выбранные вершины больше не просматривались, стираем их веса.
3. Продолжаем выполнять объединение вершин, каждый раз выбирая пару с наименьшими весами, до тех пор, пока не останется одна вершина — корень дерева. Заметим, что вес этой вершины будет равен длине сжимаемого массива.

Процесс построения дерева Хаффмана для нашего примера изображён на рисунке 2.9.

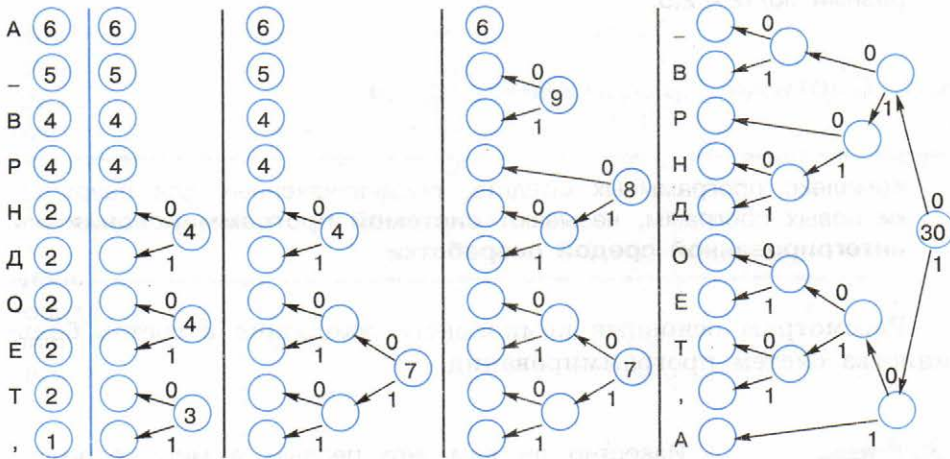


Рис. 2.9. Построение дерева Хаффмана

Теперь для определения кода каждой конкретной буквы необходимо просто пройти от корня до этой вершины, выписывая 0 и 1, встречающиеся на маршруте. В нашем примере символы получают следующие коды:

Символ	Код	Символ	Код
Н	0110	О	1000
А	11	Р	010
–	000	Е	1001
Д	0111	Т	1010
В	001	,	1011

После того как коды символов построены, остаётся сгенерировать сжатый массив данных, для чего надо снова прочесть входные данные и каждый символ заменить соответствующим ему кодом. Получаем:

```
011011000011100110000101001000101001011001111011000
01101100010100101100110010000111010100000111
```

Исходный текст состоит из 30 символов, т. е. его длина в несжатом виде будет равна 30 байт. Код сжатого текста будет занимать 95 бит или 12 байт. Получаем коэффициент сжатия, равный $30/12 = 2,5$.

8.3. Системы программирования

Комплекс программных средств, предназначенных для разработки новых программ, называют **системой программирования** или **интегрированной средой разработки**.

Рассмотрим основные компоненты, входящие в состав большинства систем программирования.



Известно ли вам, что первым в мире программистом считается англичанка Ада Лавлейс (1815–1852)? Она выполнила детальное описание вычислительной машины, проект которой был разработан Чарльзом Беббиджем, и составила первую программу для этой машины. Именно Ада Лавлейс ввела в употребление термины «цикл» и «рабочая ячейка».

Специализированный текстовый редактор позволяет программисту набрать и отредактировать текст программы на языке программирования высокого уровня. Для удобства ключевые слова языка программирования в текстовом редакторе могут выделяться различными цветами либо начертаниями шрифта.

Компьютер хранит и обрабатывает информацию, представленную в двоичных машинных кодах. Для перевода программы, написанной на языке высокого уровня, в машинные коды в состав систем программирования включены специальные программы — **трансляторы**. Существует два типа трансляторов: интерпретаторы и компиляторы.

Интерпретатор обрабатывает и исполняет команды программы последовательно, от оператора к оператору. При использовании интерпретатора, при каждом запуске программы она заново переводится в машинные коды.

Компилятор обрабатывает весь текст программы, преобразовывая его в машинный код и строя исполняемый файл, готовый к запуску. После того как программа откомпилирована, ни текст программы, ни компилятор уже не нужны.

Откомпилированные программы работают быстрее, но интерпретируемые проще исправлять и изменять.

Проведите аналогию между работой программ-трансляторов (интерпретатора и компилятора) и профессиональных переводчиков, выполняющих синхронный и письменный перевод.

Любая система программирования включает **библиотеки стандартных подпрограмм**, которые можно вызвать из вновь разрабатываемой программы. Разные части (модули) создаваемой программы и используемые в ней стандартные подпрограммы собираются в единый исполняемый файл с помощью такого элемента системы программирования, как **компоновщик**.

Важным этапом в разработке новой программы является её отладка — обнаружение и исправление возможных ошибок. Инструмент для поиска ошибок в новой программе называется **отладчиком**. Отладчик позволяет управлять процессом исполнения программы, определять место и вид ошибок в программе, наблюдать за изменением значений переменных и выражений.

Системы программирования позволяют облегчить работу программистов и сократить время на разработку сложных программ.

8.4. Прикладное программное обеспечение

Программы, с помощью которых пользователь может работать с разными видами информации, не прибегая к программированию, принято называть **прикладными программами** или **приложениями**.

Можно выделить приложения общего и специального назначения.

Приложения общего назначения требуются практически каждому пользователю для работы с разными видами информации. К приложениям общего назначения относятся: текстовые редакторы и процессоры; графические редакторы и пакеты компьютерной графики; табличные процессоры; редакторы презентаций, аудио- и видеоредакторы; системы управления базами данных; браузеры, почтовые программы и др.

Как правило, пользователь, приобретая компьютер, устанавливает на нём так называемый **офисный пакет** программ, включающий основные приложения общего назначения. Наибольшее распространение получили такие офисные пакеты, как Microsoft Office (для операционных систем Windows и Mac OS) и OpenOffice (для операционных систем Windows и Linux).

С любого компьютера, имеющего выход в Интернет, может быть доступен онлайн-офис, независимо от того, какую операционную систему этот компьютер использует.

Онлайн-офис — это набор веб-сервисов, включающий в себя все основные компоненты традиционных офисных пакетов: текстовый редактор, электронные таблицы, редактор презентаций и др. Самый известный онлайн-офис — Google Docs.

Приложения специального назначения предназначены для профессионального применения квалифицированными пользователями в различных сферах деятельности. Это: настольные издательские системы, бухгалтерские программы, системы автоматизированного проектирования (САПР), программы компьютерного моделирования, математические пакеты, геоинформационные системы (ГИС) и другие программы.

При этом границы между группами этих программ достаточно условны. По мере совершенствования интерфейсов многие программы, первоначально ориентированные на профессионалов в той или иной области, начинают широко использоваться любителями. В первую очередь это ГИС, программы компьютерного моделирования, математические пакеты и др.

САМОЕ ГЛАВНОЕ

Совокупность всех программ, предназначенных для выполнения на компьютере, называют программным обеспечением (ПО) компьютера. Всё многообразие компьютерных программ можно разделить на три группы: системное ПО, прикладное ПО, системы программирования.

Системное программное обеспечение включает в себя операционную систему и сервисные программы. Операционная система — это комплекс программ, обеспечивающих согласованное функционирование всех устройств компьютера и предоставляющих пользователю доступ к ресурсам компьютера. К сервисным программам или утилитам относят различные программы, выполняющие некоторые дополнительные услуги системного характера: обслуживание дисков (проверка, восстановление, очистка диска и др.), архивирование файлов, защита от вирусов и др.

Комплекс программных средств, предназначенных для разработки новых программ, называют системой программирования или интегрированной средой разработки. Основными компонентами системы программирования являются специализированный текстовый редактор, транслятор, отладчик и другие инструменты, позволяющие облегчить работу программистов и сократить время на разработку сложных программ.

Программы, с помощью которых пользователь может работать с разными видами информации, не прибегая к программированию, принято называть прикладными программами или приложениями. Приложения общего назначения требуются практически каждому пользователю. Приложения специального назначения предназначены для профессионального применения квалифицированными пользователями в различных сферах деятельности.

Вопросы и задания



1. Изобразите состав программного обеспечения современного компьютера в виде графа.
2. Перечислите основные функции операционной системы.
- *3. Постройте дерево Хаффмана для одной из следующих фраз:
 - 1) МАМА МЫЛА РАМУ
 - 2) ШЛА САША ПО ШОССЕ
 - 3) ТКЁТ ТКАЧ ТКАНИ
 - 4) КАРЛ У КЛАРЫ УКРАЛ КОРАЛЛЫ

4. Вспомните язык программирования высокого уровня, с которым вы познакомились в основной школе. Охарактеризуйте его алфавит, синтаксис и семантику.
5. Какое ПО называется прикладным?
6. Охарактеризуйте имеющийся в вашем распоряжении офисный пакет — укажите его название, состав, платформу, стоимость и опишите интерфейс.
7. Выполните сравнительный анализ нескольких офисных пакетов.
8. Онлайн-офисы используют технологию, известную под названием «облачные вычисления». С помощью дополнительных источников выясните, в чём её суть. Какие безусловные преимущества она имеет? Какие потенциальные опасности для пользователя таят в себе «облачные» технологии?
9. Дайте сравнительную характеристику известных вам растрового и векторного графических редакторов.
10. Перечислите основные возможности известного вам аудиоредактора.
11. Перечислите основные возможности известного вам видеоредактора.
12. Для чего предназначены табличные процессоры?
13. Каково основное назначение СУБД?
14. Назовите основные программы, которые вы используете, выйдя в Интернет.

§ 9

Файловая система компьютера

9.1. Файлы и каталоги

Из курса информатики основной школы вам известно, что **файл** — это поименованная совокупность данных определённого размера, размещаемая на внешних устройствах (носителях информации) и рассматриваемая в процессе обработки как единое целое.

Файл характеризуется набором параметров (имя, размер, дата создания, дата последней модификации) и атрибутами, используемыми операционной системой для его обработки (архивный, системный, скрытый, только для чтения). Размер файла выражается в байтах.

На каждом компьютерном носителе информации (жёстком, оптическом диске или флеш-памяти) может храниться большое количество файлов. Для удобства поиска информации файлы по определённым признакам объединяют в группы, называемые каталогами или папками.

Каталог (папка) — это поименованная совокупность файлов и подкаталогов (вложенных каталогов). Можно сказать, что каталог — это тоже файл, но только содержащий в себе информацию о заключённых в него файлах. Правда папки можно создавать средствами операционной системы, а вот для создания большинства типов файлов понадобится использование прикладного программного обеспечения.

9.2. Функции файловой системы

Работу пользователя с файлами обеспечивает подсистема ОС, называемая файловой системой.

Файловая система — это часть ОС, определяющая способ организации, хранения и именования данных на носителе информации.

Современные файловые системы решают следующие задачи:

- определяют правила построения имён файлов и каталогов;
- поддерживают программный интерфейс работы с файлами для приложений (предоставляют приложениям функции для работы с файлами);
- определяют порядок размещения файлов на диске;
- обеспечивают защиту данных в случае сбоев и ошибок;
- обеспечивают установку прав доступа к данным для каждого конкретного пользователя;
- обеспечивают совместную работу с файлами (если один пользователь открыл файл, то для остальных устанавливается режим «только чтение»).

Рассмотрим некоторые из этих функций более подробно.

Правила построения имён файлов и каталогов. Файловые системы современных ОС допускают использование имён длиной до 255 символов, причём в них можно использовать буквы национальных алфавитов и пробелы.

В ОС Windows в имени файла запрещено использование следующих символов: \, /, :, *, ?, ", <, >, |. В Linux эти символы, кроме /, допустимы, хотя использовать их следует с осторожностью, т. к. некоторые из них могут иметь специальный смысл, а также из соображений совместимости с другими ОС.

Операционная система Linux, в отличие от Windows, различает строчные и прописные буквы в имени файла, например: FILE.txt, file.txt и FiLe.txt — это в Linux три разных файла.

Порядок размещения файлов на диске. Файл, представляемый нами как единое целое, на самом деле может быть разбросан «кусочками» по всему диску. Минимальный размер такого «кусочка» (кластера, блока) — от 512 байт до 64 Кбайт в зависимости от используемой файловой системы. При размещении на диске каждому файлу отводится целое число кластеров. Например, на файл размером 65 Кбайт будет отведено два кластера по 64 Кбайта, при этом второй кластер будет считаться занятым, хотя фактически значительная его часть использоваться не будет.

Использование кластеров больших размеров обеспечивает следующие преимущества: 1) повышается скорость чтения/записи файлов, имеющих большие размеры; 2) увеличивается максимальный объём диска, который поддерживает файловая система.

Как вы можете это объяснить?

Почему в файловой системе с большими кластерами (блоками) невыгодно хранить маленькие файлы? К чему это может привести?

Защита данных в случае сбоев и ошибок. Эта функция обеспечивается за счёт журналирования, суть которого состоит в следующем. Перед началом выполнения операций с файлами ОС записывает (сохраняет) список действий, которые она будет проводить с файловой системой. Эти записи хранятся в отдельной части файловой системы, называемой журналом. Как только изменения файловой системы внесены в журнал, она применяет эти изменения к файлам, после чего удаляет эти записи из журнала. Если во время выполнения операций с файлами произошёл сбой, то по записям в журнале можно определить пострадавшие файлы и восстановить их.

9.3. Файловые структуры

Совокупность файлов на диске и взаимосвязей между ними называют **файловой структурой** диска. Первоначально файловые системы поддерживали только одноуровневые файловые структуры: все файлы хранились в одном каталоге.

Для хранения большого (сотни и тысячи) количества файлов используются иерархические (многоуровневые) файловые структуры: файлы группируются в каталоги, каталоги могут группироваться в каталоги более высокого уровня.

Пользователь, объединяя по собственному усмотрению файлы в каталоги, может создать удобную для себя систему хранения информации.

Графическое изображение иерархической файловой структуры называется деревом. Древовидные иерархические структуры можно изображать вертикально и горизонтально (рис. 2.10).

Чтобы обратиться к нужному файлу (каталогу), хранящемуся на некотором диске, можно указать путь (адрес каталога) — набор символов, показывающий расположение файла в файловой системе.



Рис. 2.10. Типовая древовидная иерархическая структура, принятая в Windows

Полное имя файла — запись пути к файлу, завершаемая именем файла.

Например:

Е:\Документы\Задачи — путь к файлу Расчёты.xlsx;

Е:\Документы\Задачи\Расчёты.xlsx — полное имя файла Расчёты.xlsx.

Для того чтобы «добраться» до нужного файла в многоуровневой файловой структуре, пользователю совсем не обязательно перемещаться по каталогам, образующим путь к файлу. Современные операционные системы имеют специальные инструменты, позволяющие достаточно быстро находить нужные файлы даже в том случае, когда точно не известно их расположение. Если пользователь знает только некоторую часть имени файла (например, он знает, что в имени файла обязательно должна быть цифра 7), то для его поиска можно воспользоваться маской имени файла.

Маска имени файла — последовательность букв, цифр и других допустимых в именах файлов символов, а также символов «?» и «*», определяющая те или иные требования к имени файла.

Символ «?» (вопросительный знак) означает, что на его месте в имени файла должен быть ровно один произвольный символ.

Символ «*» (звёздочка) означает, что на его месте в имени файла может быть последовательность любых символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Рассмотрим несколько примеров масок:

- маске *.* соответствуют абсолютно все файлы, поскольку набор произвольных символов указывается как для имени, так и для расширения файла;
- маске *.mp3 соответствуют все аудиозаписи формата mp3;
- маска info.* позволит отобразить файлы всех типов с именем info;
- маске info*.* будут соответствовать все файлы, начинающиеся с info;
- маске *info.* будут соответствовать все файлы, имена которых заканчиваются на info;
- маска ???.* позволит отобразить все файлы с трёхбуквенными именами;
- маске info??.* будут соответствовать все файлы, имена которых состоят из шести символов и начинаются с info;

- маска **info??*.*** позволяет найти все файлы, имена которых начинаются с info и содержат шесть и более символов.

Подумайте, какая из трёх следующих масок позволит отобразить больше всего файлов:

- 1) *.doc?;
- 2) *.doc;
- 3) *.doc*.

Пример. В каталоге находится 6 файлов:

```
motors.dat;                torsten.docx;
motors.doc;                victoria.docx;
storch.doc;                x_torero.doc.
```

С помощью маски из них была отобрана следующая группа файлов:

```
motors.doc;
storch.doc;
victoria.docx;
x_torero.doc.
```

Какая из следующих масок была при этом использована?

- 1) *tor?*.d*;
- 2) ?tor*.doc;
- 3) *?tor?*.do*;
- 4) *tor?.doc*.

Выясним, какие группы файлов позволит выбрать каждая из масок. Результаты анализа представим в таблице:

Имя \ Маска	*tor?*.d*	?tor*.doc	*?tor?*.do*	*tor?.doc*
motors.dat				
motors.doc				
storch.doc				
torsten.docx				
victoria.docx				
x_torero.doc				

Здесь мы серым цветом выделили строки, соответствующие файлам, которые должны быть отображены.



Если файл соответствует маске, то в ячейке, находящейся на пересечении строки с именем файла и столбца с именем маски, будем ставить «+», в противном случае — «-».

В столбце искомой маски, знаки «+» должны соответствовать отобранным файлам, знаки «-» — всем прочим.

Имя \ Маска	*tor?.d*	?tor*.doc	*?tor?.do*	*tor?.doc*
motors.dat	+	-	-	-
motors.doc		-	+	+
storch.doc			+	-
torsten.docx			-	
victoria.docx			+	
x_torero.doc			+	

Анализируя маску ***tor?.d***, ставим знак «+» в ячейку, соответствующую файлу `motors.dat`. Данная маска позволяет отобразить файл, который не входит в интересующую нас группу, следовательно, она не может обеспечить отбор нужных файлов. Дальнейшее рассмотрение этой маски можно прекратить.

Маска **?tor*.doc** не позволит отобразить файл `motors.dat`, но она же не позволит отобразить и подлежащий отбору файл `motors.doc`. Следовательно, дальнейшее рассмотрение этой маски можно прекратить.

Маска ***?tor?.do*** позволяет отобразить только те файлы, которые нам нужны. Её можно использовать для решения задачи.

Но, возможно, задача имеет не одно решение. Проверяем маску ***tor?.doc***. Она не позволит нам отобразить файл `storch.doc`.

Итак, решением задачи может быть только третья маска (***?tor?.do***).

САМОЕ ГЛАВНОЕ

Файл — это поименованная совокупность данных определённого размера, размещаемая на внешних устройствах (носителях информации) и рассматриваемая в процессе обработки как единое целое.

Каталог (папка) — это поименованная совокупность файлов и подкаталогов (вложенных каталогов).

Файловая система — это часть ОС, определяющая способ организации, хранения и именования данных на носителе информации.

Для того чтобы «добраться» до нужного файла в многоуровневой файловой структуре, пользователь может перемещаться по каталогам, образуя путь к файлу. Кроме того, можно воспользоваться поиском по маске имени файла.

Вопросы и задания



1. Что такое файл? Что такое каталог?
2. Согласны ли вы с тем, что каталог — это файл? Обоснуйте свою точку зрения.
3. Что такое файловая система? Какие задачи она решает?
4. Что такое кластер?
5. На рисунке изображён логический диск, разделённый на кластеры. Размер каждого кластера составляет 16 Кбайт. Используемые кластеры — серые, свободные кластеры — белые.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

На диск требуется записать файлы: файл *A* размером 20 Кбайт, затем файл *B* размером 30 Кбайт, затем файл *C* размером 65 Кбайт, и наконец — файл *D* размером 60 Кбайт.

Операционная система для хранения каждого файла пытается искать смежные (соседние) свободные кластеры, чтобы избежать фрагментации файла — рассеивания содержимого файла по диску, приводящего к уменьшению производительности.

В какие кластеры может быть записано содержимое файлов при соблюдении этого условия?

6. В Windows существует три возможных варианта файловой системы: NTFS, FAT32 и редко используемая устаревшая система FAT (также известная как FAT16). Найдите в Интернете информацию об этих файловых системах. Какая из них является предпочтительной файловой системой для Windows 8?

7. Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги Байкал, Путешествия, Фото, Е:, Документы, География, Карты, География, Изображения. При каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше. В каталоге, из которого были начаты перемещения, пользователь скопировал файл 1245.jpg. Этот файл он вставил в каталог, в котором оказался в результате своих перемещений, переименовав его в Листвянка.jpg. Укажите полные имена файлов 1245.jpg и Листвянка.jpg.

8. Определите, какое из следующих имён файлов удовлетворяет маске ?ba*r.?xt.

1) bar.txt; 2) obar.txt; 3) obar.xt; 4) barr.txt.

9. В каталоге находится 6 файлов:

chifera.dat;

deLafer.doc;

chifera.doc;

oferta.doc;

ferrum.doc;

tokoferol.doc.

Определите, с помощью какой маски из каталога будет отобрано ровно три файла.

1) *fer?*.d*; 2) ?fer*.doc; 3) *?fer*?.do*; 4) *fer?.doc.

Дополнительные материалы к главе смотрите в авторской мастерской.

Глава 3

ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ

§ 10

Представление чисел в позиционных системах счисления

10.1. Общие сведения о системах счисления

Как только люди начали считать, у них появилась потребность в записи чисел. Находки археологов свидетельствуют о том, что первоначально число предметов отображали равным количеством каких-либо значков (бирок): зарубок, чёрточек, точек. Такая система записи чисел называется единичной (унарной), т. к. любое число в ней образуется путём повторения одного знака, символизирующего единицу.

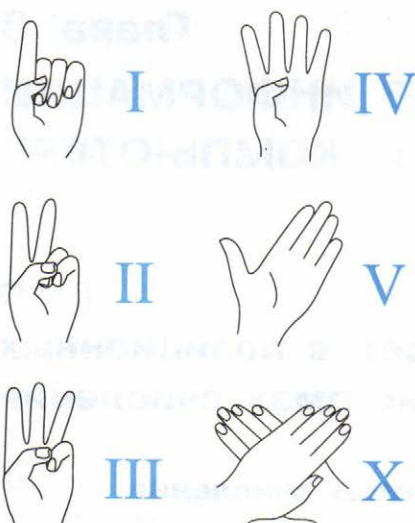
Подумайте, где в наши дни можно найти отголоски унарной системы счисления.

Унарная система — не самый удобный способ записи чисел: записывать таким способом большие значения утомительно, да и сами записи при этом получаются очень длинными. С течением времени возникли иные, более удобные и экономичные системы счисления.

Система счисления или нумерация — это способ записи (обозначения) чисел.

Система счисления — это знаковая система, в которой числа записываются по определённым правилам с помощью символов некоторого алфавита, называемых цифрами. Количество цифр, составляющих алфавит, называется его размерностью или мощностью.

Различают непозиционные и позиционные системы счисления.



В непозиционных системах счисления величина, которую обозначает цифра, не зависит от положения этой цифры в числе.

Примером непозиционной системы, которая сохранилась до наших дней, может служить система счисления, которая применялась более двух с половиной тысяч лет назад в Древнем Риме. В основе римской системы счисления лежали знаки I (один палец) для числа 1, V (раскрытая ладонь) для числа 5, X (две скрещённые ладони) для числа 10, а для обозначения чисел 100, 500 и 1000 стали применять первые буквы соответствующих

латинских слов (*centum* — сто, *demimille* — половина тысячи, *mille* — тысяча).

Чтобы записать число, римляне разлагали его на сумму тысяч (M), полутысяч (D), сотен (C), полусотен (L), десятков (X), пятёрок (V), единиц (I). Например, десятичное число 128 представляется следующим образом:

$$\text{CXXVIII} = 100 + 10 + 10 + 5 + 1 + 1 + 1$$

(одна сотня, два десятка, пять, три единицы).

Для записи промежуточных чисел римляне использовали не только сложение, но и вычитание. При этом применялось следующее правило: каждый меньший знак, поставленный справа от большего, прибавляется к его значению, а каждый меньший знак, поставленный слева от большего, вычитается из него.

Например, XI обозначает 11, а IX обозначает 9.

Римскими цифрами пользовались очень долго. Ещё 200 лет назад в деловых бумагах числа должны были обозначаться римскими цифрами (считалось, что обычные арабские цифры легко подделать). Римская система счисления сегодня используется в основном для наименования знаменательных дат, томов, разделов и глав в книгах.

Подумайте, почему римскую систему счисления нельзя считать полностью непозиционной.



Непозиционные системы счисления имеют ряд существенных недостатков:

- существует постоянная потребность введения новых знаков для записи больших чисел;
- невозможно представлять дробные и отрицательные числа;
- сложно выполнять арифметические операции, т. к. не существует алгоритмов их выполнения.

Всех перечисленных недостатков лишены позиционные системы счисления.

Система счисления называется позиционной, если количественный эквивалент цифры зависит от её положения (места, позиции) в записи числа.

Например, используемая повсеместно десятичная система счисления — позиционная. Рассмотрим число 555. Цифра 5, стоящая в записи этого числа на первом месте, обозначает количество сотен и соответствует числу 500; цифра, стоящая посередине, обозначает 5 десятков (50); последняя цифра 5 соответствует пяти единицам. Исходное число можно представить в виде суммы:

$$555 = 500 + 50 + 5.$$

Потребовалось много тысячелетий, чтобы люди научились называть и записывать числа так, как это делаем мы с вами. Начало этому было положено в Древнем Египте и Вавилоне, а завершили дело индийские математики в V–VII веках нашей эры. Важным достижением индийской науки было введение особого обозначения для пропуска разрядов — нуля. Арабы, познакомившись с этой нумерацией первыми, по достоинству её оценили, усвоили и перенесли в Европу. Получив название арабской, эта система в XII веке нашей эры распространилась по всей Европе. И так как эта система счисления проще и удобнее остальных, быстро их вытеснила.

Французский математик Пьер Симон Лаплас (1749–1827) оценил «открытие» позиционной системы такими словами: «Мысль выражать все числа немногими знаками, придавая им кроме значения по форме ещё значение по месту, настолько проста, что именно из-за этой простоты трудно оценить, насколько она удивительна».

10.2. Позиционные системы счисления

Существует бесконечно много позиционных систем счисления. Каждая из них определяется целым числом $q > 1$, называемым **основанием** системы счисления. Основание определяет (даёт)

название системы счисления: двоичная, троичная, восьмеричная, шестнадцатеричная, q -ичная и т. д. Можно говорить «система счисления с основанием q » (табл. 3.1).

Основное достоинство любой позиционной системы счисления — возможность записи произвольного числа ограниченным количеством символов. Для записи чисел в позиционной системе счисления с основанием q нужен алфавит из q цифр: 0, 1, 2, ..., $q - 1$.

Таблица 3.1

Основания и алфавиты систем счисления

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

В q -ичной системе счисления q единиц какого-либо разряда образуют единицу следующего разряда.

Целое число без знака A в q -ичной системе счисления представляется в виде конечной суммы степеней числа q — суммы разрядных слагаемых:

$$A = a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_1 \cdot q^1 + a_0 \cdot q^0 = \sum_{i=0}^{n-1} a_i \cdot q^i.$$

Здесь:

- q — основание системы счисления;
- a_i — цифры, принадлежащие алфавиту данной системы счисления ($0 \leq a_i \leq q - 1$);
- q^i — весовой коэффициент разряда.

Последовательность чисел, каждое из которых задаёт «вес» соответствующего разряда, называется **базисом позиционной системы счисления**.

Число A , свёрнутая запись которого в системе счисления с основанием q имеет вид $a_{n-1}a_{n-2}\dots a_0$, может быть представлено в развёрнутой форме как $a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_0 \cdot q^0 + a_{-1} \cdot q^{-1} + \dots + a_{-m} \cdot q^{-m}$.

Свёрнутой формой записи числа мы пользуемся в повседневной жизни, иначе её называют естественной формой или цифровой.

Развёрнутая форма записи чисел также всем хорошо известна. Ещё в начальной школе дети учатся записывать числа в виде суммы разрядных слагаемых. Например:

$$125\ 248 = 1 \cdot 100\ 000 + 2 \cdot 10\ 000 + 5 \cdot 1\ 000 + 2 \cdot 100 + 4 \cdot 10 + 8 \cdot 1.$$

Если представить разряды в виде степеней основания, то получим:

$$125\ 248 = 1 \cdot 10^5 + 2 \cdot 10^4 + 5 \cdot 10^3 + 2 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0.$$

Аналогичным образом представляются и дроби:

$$0,125 = \frac{1}{10} + \frac{2}{100} + \frac{5}{1000} = 1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 5 \cdot 10^{-3}.$$

Иногда бывает полезно преобразовать развёрнутую форму записи числа так, чтобы избежать возведения основания системы счисления в степени.

Например, можно записать:

$$\begin{aligned} 125\ 248 &= 1 \cdot 10^5 + 2 \cdot 10^4 + 5 \cdot 10^3 + 2 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0 = \\ &= (((((1 \cdot 10 + 2) \cdot 10 + 5) \cdot 10 + 2) \cdot 10 + 4) \cdot 10 + 8); \\ 0,125 &= 1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 5 \cdot 10^{-3} = ((5/10 + 2)/10 + 1)/10. \end{aligned}$$

Такую форму представления числа называют разложением по схеме Горнера.

Изучая десятичную систему счисления с раннего возраста и используя её в повседневной практике, многие люди не догадываются о существовании других систем счисления.

Но так ли хороша десятичная система счисления? Великий французский математик и естествоиспытатель Блез Паскаль (1623–1662) писал: «Десятичная система построена довольно неразумно, конечно, в соответствии с людскими обычаями, а вовсе не с требованиями естественной необходимости, как склонно думать большинство людей». В ряде теоретических и практических задач некоторые системы счисления, отличные от десятичной, имеют определённые преимущества.

Первые механические счётные машины были разработаны на основе десятичной системы счисления. Для реализации десяти устойчивых состояний в них использовались сложные системы зубчатых колёс (рис. 3.1). Такие машины были очень громоздки, занимали много места.



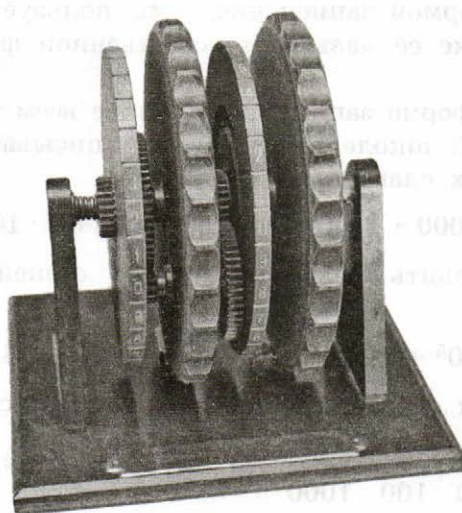


Рис. 3.1. Механизм передачи десятков в арифмометре П. Л. Чебышёва

Так, если бы проект Аналитической машины Ч. Беббиджа — механического прототипа появившихся спустя столетие ЭВМ — был реализован, то по размерам такая машина сравнялась бы с локомотивом. В 1937 году немецкий инженер К. Цузе создал вычислительную машину, основанную на принципах действия аналитической машины Ч. Беббиджа. Она была механической, но работала на основе двоичной системы счисления, что позволило значительно уменьшить её размеры: машина занимала всего 2 м² на столе в квартире изобретателя!

В наши дни большой практический интерес представляют двоичная, троичная, восьмеричная и шестнадцатеричная системы счисления.

10.3. Перевод чисел из q -ичной в десятичную систему счисления

Перевод числа, записанного в системе счисления с основанием q , в десятичную систему счисления основан на использовании развёрнутой формы записи чисел (рис. 3.2).

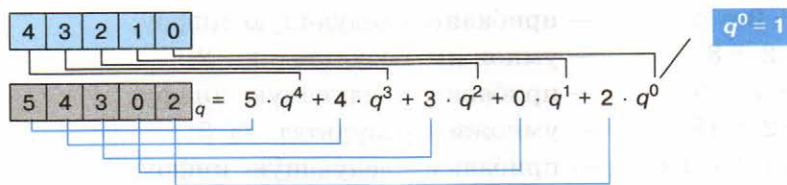


Рис. 3.2. Перевод числа из q -ичной системы счисления в десятичную

Для перевода числа A_q в десятичную систему счисления достаточно:

- 1) записать развёрнутую форму числа A_q ;
- 2) представить все числа, фигурирующие в развёрнутой форме, в десятичной системе счисления;
- 3) вычислить значение полученного выражения по правилам десятичной арифметики.

Переведём числа 212_3 , 123_5 и $12A_{16}$ в десятичную систему счисления:

$$212_3 = 2 \cdot 3^2 + 1 \cdot 3^1 + 2 \cdot 3^0 = 2 \cdot 9 + 1 \cdot 3 + 2 \cdot 1 = 23_{10};$$

$$123_5 = 1 \cdot 5^2 + 2 \cdot 5^1 + 3 \cdot 5^0 = 2 \cdot 25 + 2 \cdot 5 + 3 \cdot 1 = 63_{10};$$

$$12A_{16} = 1 \cdot 16^2 + 2 \cdot 16^1 + A \cdot 16^0 = 1 \cdot 256 + 2 \cdot 16 + 10 \cdot 1 = 298_{10}.$$

Перевод в десятичную систему счисления целых двоичных чисел будет значительно проще, если вспомнить и использовать уже знакомую вам таблицу степеней двойки:

n	0	1	2	3	4	5	6	7	8	9	10
2^n	1	2	4	8	16	32	64	128	256	512	1024

Например:

$$1001110011_2 = \overset{9}{1} \overset{8}{0} \overset{7}{0} \overset{6}{1} \overset{5}{1} \overset{4}{1} \overset{3}{1} \overset{2}{0} \overset{1}{0} \overset{0}{1} \overset{-1}{1} = 512 + 64 + 32 + 16 + 2 + 1 = 627_{10}.$$

Для перевода двоичного числа в десятичную систему счисления можно воспользоваться схемой Горнера:

- 1) $1 \cdot 2 = 2$ — возьмем 1, соответствующую самому старшему разряду числа, и умножим её на 2;
- 2) $2 + 0 = 2$ — прибавим следующую цифру;
- 3) $2 \cdot 2 = 4$ — умножим результат на 2;

- | | |
|-------------------------|-----------------------------|
| 4) $4 + 0 = 4$ | — прибавим следующую цифру; |
| 5) $4 \cdot 2 = 8$ | — умножим результат на 2; |
| 6) $8 + 1 = 9$ | — прибавим следующую цифру; |
| 7) $9 \cdot 2 = 18$ | — умножим результат на 2; |
| 8) $18 + 1 = 19$ | — прибавим следующую цифру; |
| 9) $19 \cdot 2 = 38$ | — умножим результат на 2; |
| 10) $38 + 1 = 39$ | — прибавим следующую цифру; |
| 11) $39 \cdot 2 = 78$ | — умножим результат на 2; |
| 12) $78 + 0 = 78$ | — прибавим следующую цифру; |
| 13) $78 \cdot 2 = 156$ | — умножим результат на 2; |
| 14) $156 + 0 = 156$ | — прибавим следующую цифру; |
| 15) $156 \cdot 2 = 312$ | — умножим результат на 2; |
| 16) $312 + 1 = 313$ | — прибавим следующую цифру; |
| 17) $313 \cdot 2 = 626$ | — умножим результат на 2; |
| 18) $626 + 1 = 627$ | — прибавим последнюю цифру. |

Рассмотрим несколько примеров решения задач.

Пример 1. Десятичное число 57 в некоторой системе счисления записывается как 212. Определим основание этой системы счисления.

Запишем условие задачи иначе: $212_q = 57_{10}$, $q > 2$.

Представим в виде суммы разрядных слагаемых:

$$212_q = 2 \cdot q^2 + 1 \cdot q^1 + 2 \cdot q^0 = 2q^2 + q + 2 = 57_{10}.$$

Решим уравнение: $2q^2 + q + 2 = 57$.

$$2q^2 + q - 55 = 0.$$

Это квадратное уравнение, его корни $x_1 = -5,5$; $x_2 = 5$.

Так как основание системы счисления должно быть натуральным числом, то $q = 5$.

Пример 2. Все пятибуквенные слова, составленные из пяти букв А, И, Р, С, Т, записаны в алфавитном порядке.

Вот начало списка:

1. ААААА
2. ААААИ
3. ААААР
4. ААААС
5. ААААТ
6. АААИА

...

Необходимо найти ответы на два вопроса.

1. На каком месте от начала списка стоит слово ИСТРА?
2. Сколько всего слов в этом списке?

Введём следующие обозначения: А — 0, И — 1, Р — 2, С — 3, Т — 4. Перепишем в новых обозначениях исходный список:

1. 00000
2. 00001
3. 00002
4. 00003
5. 00004
6. 00010
- ...

Теперь перед нами последовательность чисел от 0 до 44444, записанных в пятеричной системе счисления. При этом на 1-м месте в этой последовательности находится 0, на 2-м месте — 1, на 3-м месте — 2 и т. д. Это значит, что само число на единицу меньше того места (номера), которое оно занимает в последовательности.

Представив слово ИСТРА в новых обозначениях, получим 13420_5 . Переведём это пятеричное число в десятичную систему счисления:

$$13420_5 = 1 \cdot 5^4 + 3 \cdot 5^3 + 4 \cdot 5^2 + 2 \cdot 5^1 + 0 \cdot 5^0 = \\ = 625 + 375 + 100 + 10 = 1110_{10}.$$

Это число находится в списке на 1111-м месте.

Чтобы выяснить, сколько всего слов в списке, запишем его самое последнее слово: ТТТТТ. Ему соответствует число 44444_5 .

$$44444_5 = 4 \cdot 5^4 + 4 \cdot 5^3 + 4 \cdot 5^2 + 4 \cdot 5^1 + 4 \cdot 5^0 = 3124_{10}.$$

В списке это число стоит на 3125-м месте.

Вспомните о комбинаторике и предложите другой способ подсчёта количества слов в нашем списке.

Пример 3. Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 60, запись которых в четверичной системе счисления оканчивается на 31.

В четверичной системе счисления используются цифры 0, 1, 2 и 3; число представляется в виде суммы разрядных слагаемых:

4^0	4^1	4^2	4^3	...
1	4	16	64	



Из этой таблицы видно, что интересующие нас числа (≤ 60) не будут более чем трёхзначными. С учётом того что их запись заканчивается на 31, определим первую цифру (k):

$$k \cdot 16 + 3 \cdot 4 + 1 \leq 60, \quad k \cdot 16 \leq 47, \quad k \in \{0, 1, 2\}.$$

Искомые числа: $31_4 = 13_{10}$ ($k = 0$), $131_4 = 29_{10}$ ($k = 1$), $231_4 = 45_{10}$ ($k = 2$).

САМОЕ ГЛАВНОЕ

Система счисления — это способ записи (обозначения) чисел. Система счисления называется позиционной, если количественный эквивалент цифры зависит от её положения (места, позиции) в записи числа.

Существует бесконечно много позиционных систем счисления. Каждая из них определяется целым числом $q > 1$, называемым основанием системы счисления. Для записи чисел в позиционной системе счисления с основанием q нужен алфавит из q цифр: $0, 1, 2, \dots, q - 1$.

Число A , свёрнутая запись которого в системе счисления с основанием q имеет вид $a_{n-1}a_{n-2}\dots a_0, a_{-1}\dots a_{-m}$, может быть представлено в развёрнутой форме как $a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_0 \cdot q^0 + a_{-1} \cdot q^{-1} + \dots + a_{-m} \cdot q^{-m}$.

Для перевода числа A_q в десятичную систему счисления достаточно:

- 1) записать развёрнутую форму числа A_q ;
- 2) представить все числа, фигурирующие в развёрнутой форме, в десятичной системе счисления;
- 3) вычислить значение полученного выражения по правилам десятичной арифметики.

Вопросы и задания

1. Что такое система счисления? Какие классы систем счисления принято выделять?
2. Дайте определение позиционной системы счисления. Что называется алфавитом системы счисления? Что называется базисом позиционной системы счисления? Что называется основанием позиционной системы счисления?
3. Сколько цифр используется в системе счисления с основанием 2, 3, 5, 8, 16, 100, q ?
4. Какая форма записи числа называется развёрнутой?

5. Запишите в развёрнутой форме числа:
1) $143,511_{10}$; 2) $1435,11_8$; 3) $143,511_{16}$.
6. Воспользуйтесь схемой Горнера для представления чисел:
1) 12345_{10} ; 2) 12345_8 ; 3) $0,12345_6$.
7. Вычислите десятичные эквиваленты следующих чисел:
1) 120_3 ; 2) $100,21_4$; 3) $5A,124_{16}$.
8. Существует ли треугольник, длины сторон которого выражаются числами 12_8 , 122_3 и 11011_2 ?
9. Укажите целые десятичные числа, принадлежащие следующим числовым промежуткам:
1) $[202_3; 1000_3]$; 2) $[14_8, 20_8]$; 3) $[28_{16}, 30_{16}]$.
10. Найдите основание x системы счисления, если известно:
1) $47_{10} = 21_x$; 2) $1331_x = 6_{10}$.
11. Десятичное число 63 в некоторой системе счисления записывается как 120. Определите основание системы счисления.
12. Какое из чисел C , записанных в двоичной системе счисления, удовлетворяет неравенству $9D_{16} < C < 237_8$?
1) 10011010_2 ; 2) 10011110_2 ; 3) 10011111_2 ; 4) 11011110_2 .
13. Как изменится величина чисел $311,211_4$ и $23,45_6$ при переносе запятой на:
1) один знак вправо;
2) два знака влево?
14. При переносе запятой на два знака вправо число $240,13_x$ увеличилось в 25 раз. Чему равно x ?
15. Какое наибольшее десятичное число можно записать тремя цифрами в двоичной, восьмеричной и шестнадцатеричной системах счисления?
16. Найдите наименьшие основания систем счисления x и y , исходя из условий:
1) $23_x = 21_y$; 2) $51_x = 15_y$; 3) $144_x = 441_y$.
17. Решите уравнение $54_7 + x = 320_5$.
18. Все трёхбуквенные слова, составленные из букв И, М, Р, записаны в алфавитном порядке. Вот начало списка:
1. ИИИ
2. ИИМ
3. ИИР
4. ИМИ
...



Выясните общее количество слов в этом списке. На каких местах в этом списке стоят слова МИМ, МИР, РИМ?

19. Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 26, запись которых в троичной системе счисления оканчивается на 22.
20. Трёхзначное число, записанное в системе с основанием 3, при перестановке крайних цифр становится числом, выражающим то же количество, но уже в системе с основанием 4. Найдите это число.
21. Исполнитель умеет сравнивать однозначные числа, представленные в некоторой позиционной системе счисления. Составьте для него:
 - 1) алгоритм сравнения двух двухзначных чисел;
 - 2) алгоритм сравнения двух n -значных чисел.

§ 11

Перевод чисел из одной позиционной системы счисления в другую

11.1. Перевод целого десятичного числа в систему счисления с основанием q

Для перевода целого десятичного числа в систему счисления с основанием q следует:

- 1) последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления до тех пор, пока не получится частное, равное нулю;
- 2) полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие алфавиту новой системы счисления;
- 3) составить число в новой системе счисления, записывая его, начиная с последнего остатка.

Рассмотрим примеры перевода целых десятичных чисел в 2-ичную, 8-ричную и 16-ричную системы счисления.

Пример 1. $25_{10} = 11001_2$

$$\begin{array}{r}
 25 \mid 2 \\
 \hline
 24 \quad 12 \quad 6 \quad 3 \quad 2 \\
 \hline
 1 \quad 12 \quad 6 \quad 3 \quad 2 \\
 \hline
 \quad 0 \quad 6 \quad 3 \quad 2 \\
 \hline
 \quad \quad 0 \quad 2 \quad 1 \quad 2 \\
 \hline
 \quad \quad \quad 1 \quad 0 \quad 0 \\
 \hline
 \quad \quad \quad \quad 1
 \end{array}$$

Пример 2. $163_{10} = 243_8$

$$\begin{array}{r}
 163 \mid 8 \\
 \hline
 160 \quad 20 \quad 8 \\
 \hline
 3 \quad 16 \quad 2 \quad 8 \\
 \hline
 \quad 4 \quad 0 \quad 8 \\
 \hline
 \quad \quad 2
 \end{array}$$

Пример 3. $709_{10} = 2C5_{16}$

$$\begin{array}{r}
 709 \mid 16 \\
 \hline
 64 \quad 44 \quad 16 \\
 \hline
 69 \quad 32 \quad 2 \quad 16 \\
 \hline
 64 \quad 12 \quad 0 \quad 0 \\
 \hline
 5 \quad (C) \quad 2
 \end{array}$$

Пример 4. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись десятичного числа 22 оканчивается на 4.

Поскольку запись числа в системе счисления с основанием q заканчивается на 4, остаток от деления числа 22 на q равен 4: $22 \bmod q = 4^1$). Следовательно, $18 \bmod q = 0$. Это верно для $q \in \{18, 9, 6, 3, 2, 1\}$.

Так как в новой системе счисления запись числа оканчивается на 4, то $q > 4$. Следовательно, условию задачи удовлетворяют основания: 18, 9 и 6.

11.2. Перевод целого десятичного числа в двоичную систему счисления

Перевод целого десятичного числа, безусловно, может осуществляться по рассмотренному выше универсальному алгоритму. Но для чисел в пределах десяти тысяч (особенно если число

1) Операция \bmod — вычисление остатка от целочисленного деления.

немного больше некоторой степени двойки) бывает удобно снова воспользоваться таблицей степеней двойки.

Например: $1096_{10} = 1024 + 72 = 1024 + 64 + 8 = 10001001000_2$.

Здесь мы представили число в виде суммы степеней двойки: сначала взяли максимально возможное значение, не превышающее исходное число ($1024 < 1096$), и нашли разность между исходным числом и этим значением (72). Затем выписали степень двойки, не превышающую эту разность, и т. д. Когда исходное число было представлено в виде суммы, мы построили его двоичное представление, записав 1 в разрядах, соответствующих слагаемым, вошедшим в сумму, и 0 — во всех остальных разрядах.

11.3. Перевод целого числа из системы счисления с основанием p в систему счисления с основанием q

Каждый из нас может выполнять арифметические операции в привычной десятичной системе счисления. Выполнять такие же операции в других системах счисления человеку непривычно, а поэтому и неудобно.

Для того чтобы перевести целое число из системы счисления с основанием p в систему счисления с основанием q , достаточно:

- 1) основание новой системы счисления выразить в исходной системе счисления и все последующие действия производить в исходной системе счисления;
- 2) последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления до тех пор, пока не получится частное, равное нулю;
- 3) полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие алфавиту новой системы счисления;
- 4) составить число в новой системе счисления, записывая его, начиная с последнего остатка.

При необходимости перевести целое число из системы счисления с основанием p в систему счисления с основанием q можно попытаться воспользоваться описанным выше алгоритмом. Другой способ состоит в том, чтобы свести всё к хорошо знакомым действиям в десятичной системе счисления: перевести исходное число в десятичную систему счисления, после чего полученное десятичное число представить в требуемой системе счисления (рис. 3.3).

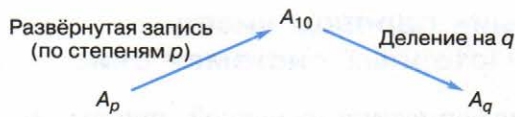


Рис. 3.3. Схема перевода целого числа из системы счисления с основанием p в систему счисления с основанием q через десятичную систему счисления

Пример 5. $1234_5 = 1 \cdot 5^3 + 2 \cdot 5^2 + 3 \cdot 5^1 + 4 \cdot 5^0 = 194_{10} = 522_6$.



11.4. Перевод конечной десятичной дроби в систему счисления с основанием q

Для перевода конечной десятичной дроби в систему счисления с основанием q следует:

- 1) последовательно умножать данное число и получаемые дробные части произведения на основание новой системы счисления до тех пор, пока дробная часть произведения не станет равна нулю или не будет достигнута требуемая точность представления числа;
- 2) полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствие алфавиту новой системы счисления;
- 3) составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

Пример 6. Переведём число $0,1875_{10}$ в двоичную систему счисления.



Выполним умножение числа $0,1875_{10}$ на 2:

Операция	Результат
$0,1875 \cdot 2 =$	0,3750 (1)
$0,3750 \cdot 2 =$	0,7500 (2)
$0,7500 \cdot 2 =$	1,5000 (3)
$0,5000 \cdot 2 =$	1,0000 (4)

Здесь жирным выделены цифры, участвующие в двоичном представлении дроби, а в скобках указан номер цифры в дроби.

$$0,1875_{10} = 0,0011_2.$$

11.5. «Быстрый» перевод чисел в компьютерных системах счисления

Из курса информатики основной школы вы знаете, что в компьютерных науках широко используются двоичная, восьмеричная и шестнадцатеричная системы счисления, благодаря чему их называют «компьютерными». Между основаниями этих систем существует очевидная связь: $16 = 2^4$, $8 = 2^3$.

Способ «быстрого» перевода основан на том, что каждой цифре числа в системе счисления, основание которой q кратно степени двойки, соответствует число, состоящее из n ($q = 2^n$) цифр в двоичной системе счисления. Замена восьмеричных цифр двоичными тройками (триадами) и шестнадцатеричных цифр двоичными четвёрками (тетрадами) позволяет осуществлять быстрый перевод между этими системами счисления, не прибегая к арифметическим операциям.

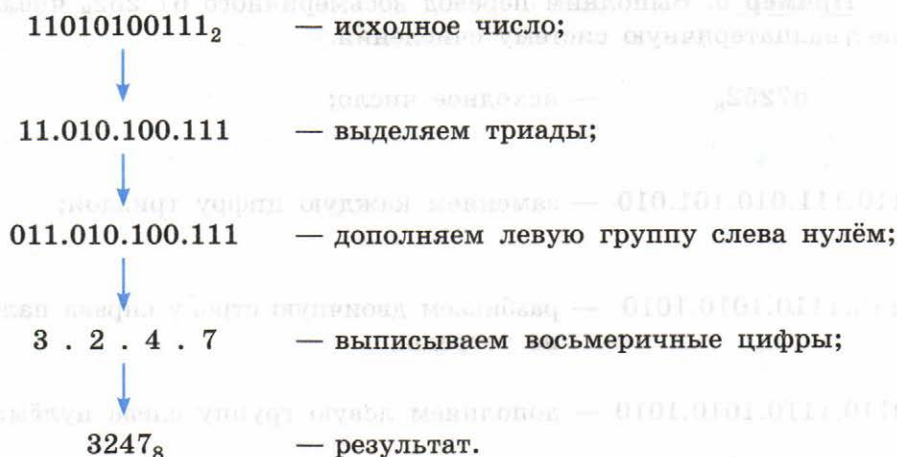
Восьмеричная цифра	Двоичная триада
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Шестнадцатеричная цифра	Двоичная тетрада
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

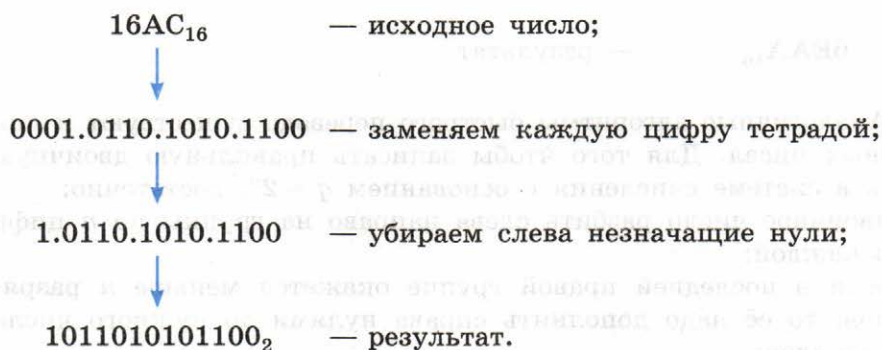
Для того чтобы целое двоичное число записать в системе счисления с основанием $q = 2^n$, достаточно:

- 1) данное двоичное число разбить справа налево на группы по n цифр в каждой;
- 2) если в последней левой группе окажется меньше n разрядов, то её надо дополнить слева нулями до нужного числа разрядов;
- 3) рассмотреть каждую группу как n -разрядное двоичное число и записать её соответствующей цифрой системы счисления с основанием $q = 2^n$.

Пример 7. Переведём число 11010100111_2 в восьмеричную систему счисления.



Пример 8. Переведём число $16AC_{16}$ в двоичную систему счисления.



Через двоичную систему счисления можно проводить быстрые переводы из восьмеричной системы счисления в шестнадцатеричную и обратно (рис. 3.4)

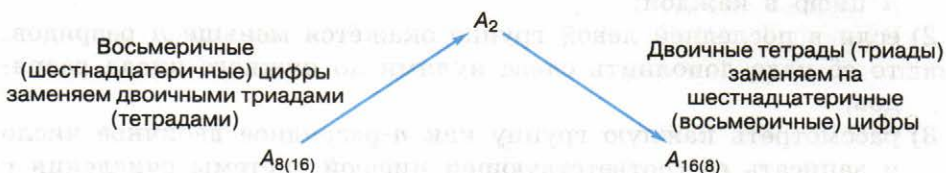
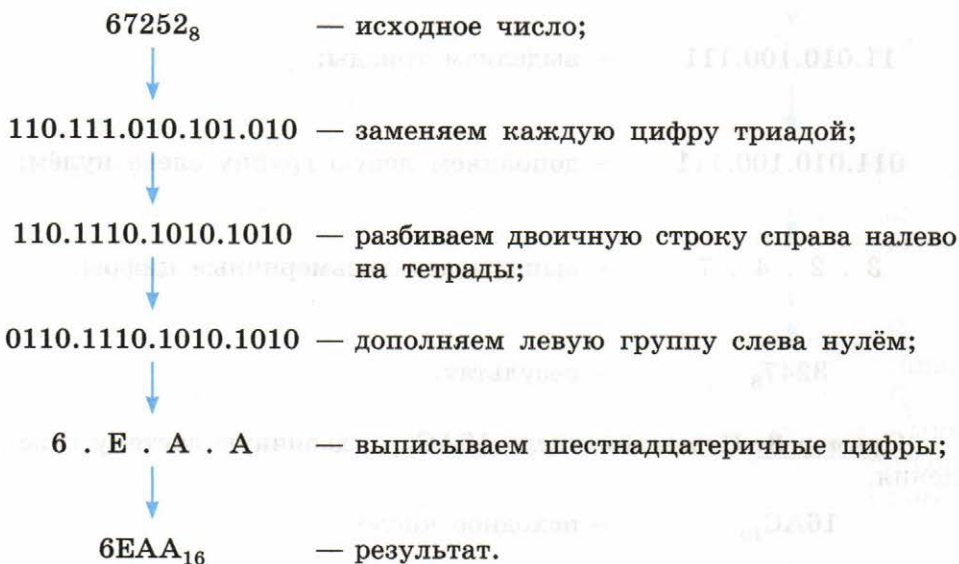


Рис. 3.4. Схема перевода целых чисел из восьмеричной системы счисления в шестнадцатеричную и обратно через двоичную систему счисления

Пример 9. Выполним перевод восьмеричного $67\ 252_8$ числа в шестнадцатеричную систему счисления.

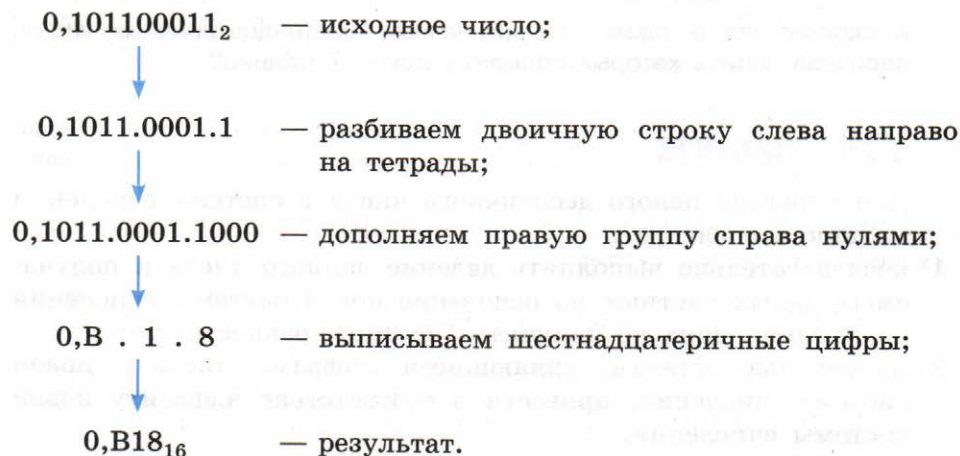


Аналогичные алгоритмы быстрого перевода существуют и для дробных чисел. Для того чтобы записать правильную двоичную дробь в системе счисления с основанием $q = 2^n$, достаточно:

- 1) двоичное число разбить слева направо на группы по n цифр в каждой;
- 2) если в последней правой группе окажется меньше n разрядов, то её надо дополнить справа нулями до нужного числа разрядов;

3) рассмотреть каждую группу как n -разрядное двоичное число и записать её соответствующей цифрой системы счисления с основанием $q = 2^n$.

Пример 10. Число $0,101100011_2$ заменим равным ему шестнадцатеричным числом.



Попытайтесь самостоятельно сформулировать алгоритм быстрого перевода произвольного двоичного числа в восьмеричную систему счисления. Примените алгоритм к числу $11101001000,11010010_2$.

Пример 11. Выясним, сколько значащих нулей в двоичной записи восьмеричного числа 1601_8 .

Для ответа на этот вопрос достаточно знать двоичные триады, соответствующие восьмеричным цифрам от 0 до 7 и выполнить «быстрый» перевод числа 1601_8 в двоичную систему счисления:

$$1601_8 = 001\ 110\ 000\ 001_2 = 1110000001_2.$$

В двоичной записи 6 значащих нулей, а первые два нуля являются незначащими и не учитываются.

Пример 12. Среди четырёхзначных шестнадцатеричных чисел, двоичная запись которых содержит ровно 7 единиц, найдём:

- 1) наименьшее число;
- 2) наибольшее число.

Наименьшее четырёхзначное шестнадцатеричное число — это $1000_{16} = 0001\ 0000\ 0000\ 0000_2$, и его двоичное представление содержит всего одну единицу. Чтобы получить наименьшее

число, удовлетворяющее условию задачи, оставшиеся шесть единиц следует разместить в самых младших разрядах. Получим $1\ 0000\ 0011\ 1111_2 = 103F_{16}$. Чтобы получить наибольшее число, удовлетворяющее условию задачи, оставшиеся шесть единиц следует разместить в самых старших разрядах. Получим $1111\ 1110\ 0000\ 0000_2 = FE00_{16}$.

А сколько всего таких четырёхзначных шестнадцатеричных чисел, двоичная запись которых содержит ровно 7 единиц?

САМОЕ ГЛАВНОЕ

Для перевода целого десятичного числа в систему счисления с основанием q следует:

- 1) последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления до тех пор, пока не получится частное, равное нулю;
- 2) полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие алфавиту новой системы счисления;
- 3) составить число в новой системе счисления, записывая его, начиная с последнего остатка.

В компьютерных науках широко используются двоичная, восьмеричная и шестнадцатеричная системы счисления, благодаря чему их называют «компьютерными». Между основаниями этих систем существует очевидная связь: $16 = 2^4$, $8 = 2^3$.

Если основание системы счисления q кратно степени двойки ($q = 2^n$), то любое число в этой системе счисления можно «быстро» перевести в двоичную систему счисления, выписав последовательно двоичные коды каждой из цифр, образующих исходное число. Замена восьмеричных цифр двоичными тройками (триадами) и шестнадцатеричных цифр двоичными четвёрками (тетрадами) позволяет осуществлять быстрый перевод между этими системами счисления, не прибегая к арифметическим операциям.

Вопросы и задания

1. Переведите целые числа из десятичной системы счисления в двоичную систему счисления:
1) 1025; 2) 512; 3) 600.

2. Переведите целое число 1147 из десятичной системы счисления в системы счисления:
 - 1) пятеричную;
 - 2) восьмеричную;
 - 3) шестнадцатеричную.
3. Переведите двоичные числа в восьмеричную систему счисления:
 - 1) 1010001001011;
 - 2) 1010,00100101.
4. Переведите двоичные числа в шестнадцатеричную систему счисления:
 - 1) 1010001001011;
 - 2) 1010,00100101.
5. Переведите числа в двоичную систему счисления:
 - 1) 266₈; 2) 266₁₆.
6. Переведите числа из восьмеричной системы счисления в шестнадцатеричную:
 - 1) 12754; 2) 1515.
7. Переведите числа из шестнадцатеричной системы счисления в восьмеричную:
 - 1) 1AE2; 2) 1C1C.
8. Сравните числа:
 - 1) 125₁₆ и 111100010101₂;
 - 2) 757₈ и 1110010101₂;
 - 3) A23₁₆ и 1232₈.
9. Сколько из чисел C , записанных в двоичной системе счисления, удовлетворяет неравенству $221_8 < C < 95_{16}$? Какие числа?
 - 1) 10010100₂; 2) 10010110₂; 3) 10010011₂; 4) 10001100₂.
10. Сколько значащих нулей в двоичной записи:
 - 1) восьмеричного числа 2501;
 - 2) шестнадцатеричного числа 12A?
11. Среди четырёхзначных восьмеричных чисел, двоичная запись которых содержит ровно 5 единиц, найдите:
 - 1) наименьшее число;
 - 2) наибольшее число.
12. Среди трёхзначных шестнадцатеричных чисел, двоичная запись которых содержит ровно 7 нулей, найдите:
 - 1) наименьшее число;
 - 2) наибольшее число.



13. Все 5-буквенные слова, составленные из букв О, П, Р, Т, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ООООО
2. ООООП
3. ООООР
4. ООООТ
5. ООПО
- ...

Какие слова находятся в этом списке на 531-м и 787-м местах?

14. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись десятичного числа 82 оканчивается на 5.

§ 12

Арифметические операции в позиционных системах счисления

Арифметические операции в позиционных системах счисления с основанием q выполняются по правилам, аналогичным правилам, действующим в десятичной системе счисления.

В начальной школе для обучения детей счёту используют таблицы сложения и умножения. Подобные таблицы можно составить для любой позиционной системы счисления.

12.1. Сложение чисел в системе счисления с основанием q

Рассмотрите примеры таблиц сложения в троичной (табл. 3.2), восьмеричной (табл. 3.4) и шестнадцатеричной (табл. 3.3) системах счисления.

Таблица 3.2

Сложение в троичной системе счисления

+	0	1	2
0	0	1	2
1	1	2	10
2	2	10	11

Таблица 3.3

Сложение в шестнадцатеричной системе счисления

+	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

Сложение в восьмеричной системе счисления

+	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

Чтобы в системе счисления с основанием q получить сумму S двух чисел A и B , надо просуммировать образующие их цифры по разрядам i справа налево:

- если $a_i + b_i < q$, то $s_i = a_i + b_i$, старший $(i + 1)$ -й разряд не изменяется;
- если $a_i + b_i \geq q$, то $s_i = a_i + b_i - q$, старший $(i + 1)$ -й разряд увеличивается на 1.

Примеры:

$$\begin{array}{r} 100202_3 \\ + 10102_3 \\ \hline 111011_3 \end{array}$$

$$\begin{array}{r} 12345_8 \\ + 2443_8 \\ \hline 15010_8 \end{array}$$

$$\begin{array}{r} FE12A9_{16} \\ + 23528_{16} \\ \hline 10047D1_{16} \end{array}$$

12.2. Вычитание чисел в системе счисления с основанием q

Чтобы в системе счисления с основанием q получить разность R двух чисел A и B , надо вычислить разности образующих их цифр по разрядам i справа налево:

- если $a_i \geq b_i$, то $r_i = a_i - b_i$, старший $(i + 1)$ -й разряд не изменяется;
- если $a_i < b_i$, то $r_i = a_i - b_i + q$, старший $(i + 1)$ -й разряд уменьшается на 1 (выполняется заём в старшем разряде).

Примеры:

$$\begin{array}{r} 100110_3 \\ - 10101_3 \\ \hline 20002_3 \end{array}$$

$$\begin{array}{r} 17345_8 \\ - 2443_8 \\ \hline 14702_8 \end{array}$$

$$\begin{array}{r} FE12A9_{16} \\ - 23528_{16} \\ \hline FBDD81_{16} \end{array}$$

12.3. Умножение чисел в системе счисления с основанием q

Рассмотрите примеры таблиц умножения в троичной (табл. 3.5), восьмеричной (табл. 3.6) и шестнадцатеричной (табл. 3.7) системах счисления.

Таблица 3.5

Умножение в троичной системе счисления

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	11

Таблица 3.6

Умножение в восьмеричной системе счисления

×	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

Таблица 3.7

Умножение в шестнадцатеричной системе счисления

×	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Рассмотрим алгоритм умножения многозначного числа на однозначное.

Чтобы в системе счисления с основанием q получить произведение M многозначного числа A и однозначного числа b , надо вычислить произведения b и цифр, образующих число A по разрядам i справа налево:

- если $a_i \cdot b < q$, то $m_i = a_i \cdot b$, старший $(i + 1)$ -й разряд не изменяется;
- если $a_i \cdot b \geq q$, то $m_i = a_i \cdot b \bmod q$, старший $(i + 1)$ -й разряд увеличивается на $a_i \cdot b \operatorname{div} q$ (где div — операция целочисленного деления).

Примеры:

$$\begin{array}{r} 11 \\ \times 202_3 \\ \hline 2_3 \\ 1111_3 \end{array}$$

$$\begin{array}{r} 123 \\ \times 246_8 \\ \hline 5_8 \\ 1476_8 \end{array}$$

$$\begin{array}{r} 6751 \\ \times AD93_{16} \\ \hline 9_{16} \\ 61A2B_{16} \end{array}$$

Умножение многозначного числа на многозначное число выполняется столбиком. При этом два множителя располагаются один под другим так, чтобы разряды чисел совпадали (находились в одном столбце).

Если один из множителей или оба множителя оканчиваются нулями, то числа записываются так, чтобы в одном столбце оказались их самые младшие разряды с цифрами, отличными от нуля. Нули переносятся в итоговое произведение, а в поле записи поэтапных произведений не заносятся.

Поэтапные (разрядные) произведения складываются по разрядам и под чертой записывается результат.

Примеры:

$$\begin{array}{r} 2020_3 \\ \times 210_3 \\ \hline 202_3 \\ + 1111_3 \\ \hline 1201200_3 \end{array}$$

$$\begin{array}{r} 2460_8 \\ \times 15_8 \\ \hline 1476_8 \\ + 246_8 \\ \hline 41560_8 \end{array}$$

12.4. Деление чисел в системе счисления с основанием q

Деление нельзя свести к поразрядным операциям над цифрами, составляющими число.

Деление чисел в системе счисления с произвольным основанием q выполняется так же, как и в десятичной системе счисления.

Пример:

$$\begin{array}{r} 2640_8 \overline{) 17_8} \\ \underline{17_8} \\ 74_8 \\ \underline{74_8} \\ 0 \end{array}$$

12.5. Двоичная арифметика

Арифметика двоичной системы счисления основывается на использовании следующих таблиц сложения, вычитания и умножения:

+	0	1
0	0	1
1	1	10

-	0	1
0	0	11
1	1	0

×	0	1
0	0	1
1	1	1

Рассмотрим несколько простых, но очень важных примеров на сложение и вычитание в двоичной системе счисления.

$$\begin{array}{r} 111111 \\ + 111111_2 \\ \hline 1000000_2 \end{array}$$

$$\begin{array}{r} 01111 \\ - 10000_2 \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 0111 \\ - 101010_2 \\ \hline 100011_2 \end{array}$$

Пример 1. Найдём количество единиц в двоичной записи числа, являющегося результатом десятичного выражения $2^{4000} + 4^{2016} + 2^{2018} - 8^{600} + 6$.

Представим все операнды исходного выражения в виде степеней двойки:

$$\begin{aligned} 4^{2016} &= (2 \cdot 2)^{2016} = (2^2)^{2016} = 2^2 \cdot 2016 = 2^{4032}, \\ 8^{600} &= (2^3)^{600} = 2^{1800}, \\ 6 &= 4 + 2 = 2^2 + 2^1. \end{aligned}$$

Исходное выражение примет вид:

$$2^{4000} + 4^{2016} + 2^{2018} - 8^{600} + 6 = 2^{4000} + 2^{4032} + 2^{2018} - 2^{1800} + 2^2 + 2^1.$$

Перепишем выражение в порядке убывания степеней:

$$2^{4032} + 2^{4000} + 2^{2018} - 2^{1800} + 2^2 + 2^1.$$

Для работы с десятичными числами вида 2^n полезно иметь в виду следующие закономерности в их двоичной записи:

$$\begin{aligned} 2^1 &= 10 = 1 + 1; \\ 2^2 &= 100 = 11 + 1; \\ 2^3 &= 1000 = 111 + 1; \\ 2^4 &= 10000 = 1111 + 1 \text{ и т. д.} \end{aligned}$$

В общем виде:

$$2^n = \underbrace{10\dots0}_n = \underbrace{1\dots1}_n + 1.$$

Для натуральных n и m таких, что $n > m$, получаем:

$$2^n + 2^m = \underbrace{10\dots0}_n + \underbrace{10\dots0}_m = \underbrace{10\dots0}_{n-m-1} \underbrace{10\dots0}_m,$$

$$2^n - 2^m = \underbrace{1\dots1}_n + 1 - (\underbrace{1\dots1}_m + 1) = \underbrace{1\dots1}_n - \underbrace{1\dots1}_m = \underbrace{1\dots1}_{n-m} \underbrace{0\dots0}_m.$$

Эти соотношения позволят нам подсчитать количество единиц в нашем выражении, не прибегая к его вычислению.

Действительно, двоичные представления чисел 2^{4032} и 2^{4000} внесут в двоичное представление суммы по одной единице. Разность $2^{2018} - 2^{1800}$ в двоичной записи представляет собой цепочку из 218 единиц и следующих за ними 1800 нулей. Слагаемые 2^2 и 2^1 дают ещё 2 единицы.

Итого: $1 + 1 + 218 + 1 + 1 = 222$.

Пример 2. Найдём количество цифр в восьмеричной записи числа, являющегося результатом десятичного выражения $2^{299} + 2^{298} + 2^{297} + 2^{296}$.

Двоичное представление исходного числа имеет вид: $11110\dots0$. Всего в этой записи 300 двоичных символов. При переводе двоичного числа в восьмеричную систему счисления каждая триада исходного числа заменяется восьмеричной цифрой. Следовательно, восьмеричное представление исходного числа состоит из 100 цифр.

САМОЕ ГЛАВНОЕ

Арифметические операции в позиционных системах счисления с основанием q выполняются по правилам, аналогичным правилам, действующим в десятичной системе счисления.

Если необходимо вычислить значение арифметического выражения, операнды которого представлены в различных системах счисления, можно:

- 1) все операнды представить в привычной нам десятичной системе счисления;
- 2) вычислить результат выражения в десятичной системе счисления;
- 3) перевести результат в требуемую систему счисления.

Для работы с десятичными числами вида 2^n полезно иметь в виду следующие закономерности в их двоичной записи:

$$2^n = \underbrace{10\dots0}_n = \underbrace{1\dots1}_n + 1.$$

Для натуральных n и m таких, что $n > m$, получаем:

$$2^n + 2^m = \underbrace{10\dots0}_n + \underbrace{10\dots0}_m = \underbrace{10\dots0}_{n-m-1} \underbrace{10\dots0}_m,$$

$$2^n - 2^m = \underbrace{1\dots1}_n + 1 - (\underbrace{1\dots1}_m + 1) = \underbrace{1\dots1}_n - \underbrace{1\dots1}_m = \underbrace{1\dots1}_{n-m} \underbrace{10\dots0}_m.$$



Вопросы и задания

1. Выполните арифметические операции над двоичными числами:

- 1) $10010011 + 101101$;
- 2) $110010,11 + 110110,11$;
- 3) $110101110 - 10111111$;
- 4) $111110 \cdot 100010$;
- 5) $11111100101 : 101011$.

Для того чтобы убедиться в правильности полученных результатов, найдите десятичные эквиваленты операндов и результата.

2. Какое число следует за каждым из данных:

- 1) 223_4 ; 2) 677_8 ; 3) 2222_3 ; 4) 1001_2 ?

Ответ для каждого числа дайте в указанной и десятичной системах счисления.

3. Какое число предшествует каждому из данных:

- 1) 222_3 ; 2) 1000_5 ; 3) 233_4 ; 4) 1001_2 ?

Ответ для каждого числа дайте в указанной и десятичной системах счисления.

4. Сумму восьмеричных чисел $17 + 1700 + 170000 + 17000000 + 1700000000$ перевели в шестнадцатеричную систему счисления. Найдите в шестнадцатеричной записи числа, равного этой сумме, пятую цифру слева.

5. Вычислите значение выражения:

- 1) $(1111101_2 + AF_{16}) : 36_8$;
- 2) $125_8 + 11101_2 \cdot A2_{16} - 1417_8$.

6. Найдите среднее арифметическое следующих чисел:

1) 10010110_2 , 1100100_2 и 110010_2 ;

2) 226_8 , 64_{16} и 62_8 .

7. В примерах на сложение восстановите неизвестные цифры, обозначенные знаком вопроса, определив вначале, в какой системе счисления эти числа записаны.

$$\begin{array}{r} 1) \quad 2 \ ? \ ? \ 2 \ 1 \\ + \ 1 \ 2 \ 3 \ ? \\ \hline \ ? \ 2 \ 0 \ 3 \end{array}$$

$$\begin{array}{r} 2) \quad 5 \ ? \ 5 \ 5 \\ + \ ? \ 3 \ 2 \ 7 \\ \hline \ ? \ 1 \ 6 \ ? \ 4 \end{array}$$

$$\begin{array}{r} 3) \quad 2 \ 1 \ ? \ 0 \ 2 \\ + \ ? \ 1 \ 2 \ 1 \ 2 \\ \hline \ ? \ ? \ ? \ 0 \ 2 \ 1 \end{array}$$

8. Даны 4 целых числа, записанные в двоичной системе счисления:

$$11000000, 11000011, 11011001, 11011111.$$

Сколько среди них чисел, больших, чем $AB_{16} + 25_8$?

9. Сколько единиц в двоичной записи числа $4^{2014} + 2^{2015} - 9$?

*10. Сколько единиц в двоичной записи числа

$$8^{4024} - 4^{1605} + 2^{1024} - 126?$$

11. Сколько цифр в восьмеричной записи числа $2^{1024} + 2^{1026}$?

12. Какая первая цифра в шестнадцатеричной записи числа $2^{1024} + 2^{1025}$?



§ 13

Представление чисел в компьютере

Самым первым видом данных, с которыми начали работать компьютеры, были числа. ЭВМ первого поколения могли производить только математические расчёты (вычисления).

Из курса информатики основной школы вы помните, что компьютеры работают с целыми и вещественными числами. Их представление в памяти осуществляется разными способами.

13.1. Представление целых чисел

Во многих задачах, решаемых на компьютере, обрабатываются целочисленные данные. Прежде всего, это задачи эконо-

мического характера, при решении которых данными служат количества акций, сотрудников, деталей, транспортных средств и др. Целые числа используются для обозначения даты и времени, для нумерации различных объектов: элементов массивов, записей в базах данных, машинных адресов и т. д. По своей природе множество целых чисел дискретно, т. к. состоит из отдельных элементов.

И хотя любое целое число можно рассматривать как вещественное, но с нулевой дробной частью, предусмотрены специальные способы представления целых чисел. Это обеспечивает: эффективное расходование памяти, повышение быстродействия, повышение точности вычислений за счёт введения операции деления нацело с остатком.

Для компьютерного представления целых чисел используется несколько различных способов, отличающихся друг от друга количеством разрядов (под целые числа обычно отводится 8, 16, 32 или 64 разряда) и наличием или отсутствием знакового разряда.

Беззнаковое представление можно использовать только для неотрицательных целых чисел.

Для получения компьютерного представления беззнакового целого числа в n -разрядной ячейке памяти достаточно перевести его в двоичную систему счисления и, при необходимости, дополнить полученный результат слева нулями до n -разрядов.

Например, десятичные числа 130 и 39 в восьмиразрядном представлении будут иметь вид:

1	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---

0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---

Понятно, что существуют ограничения на числа, которые могут быть записаны в n -разрядную ячейку памяти. Максимальное значение целого неотрицательного числа достигается в случае, когда во всех разрядах ячейки хранятся единицы. Для n -разрядного представления оно будет равно $2^n - 1$. Минимальное число соответствует n нулям, хранящимся в n разрядах памяти, и равно нулю. Далее приведены диапазоны значений для беззнаковых целых n -разрядных чисел:

Количество разрядов	Диапазон чисел
8	[0; 255]
16	[0; 65 535]
32	[0; 4 294 967 295]
64	[0; 18 446 744 073 709 551 615]

При знаковом представлении целых чисел старший разряд ячейки отводится под знак (0 — для положительных, 1 — для отрицательных чисел), а остальные разряды — под цифры числа.

Представление числа в привычной для человека форме «знак-величина», при которой старший разряд ячейки отводится под знак, а остальные разряды — под цифры числа, называется **прямым кодом**.

Например, прямые коды чисел 48 и -52 для восьмиразрядной ячейки равны:

0	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---

1	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

Минимальное отрицательное число, которое можно записать в знаковом представлении в n разрядах, равно 2^{n-1} . Максимальное положительное число, которое можно записать в знаковом представлении в n разрядах, равно $2^{n-1} - 1$. Ниже приведены диапазоны значений для знаковых представлений целых чисел в ячейках с различной разрядностью:

Количество разрядов	Диапазон чисел
8	[-128; 127]
16	[-32 768; 32 767]
32	[-2 147 483 648; 2 147 483 647]
64	[-9 223 372 036 854 775 808; 9 223 372 036 854 775 807]

В математике множество целых чисел бесконечно. Компьютер работает с ограниченным множеством целых чисел.

Прямой код положительного числа отличается от прямого кода равного по абсолютной величине отрицательного числа только содержимым знакового разряда.

В прямом коде числа можно хранить, но выполнение арифметических операций над числами в прямом коде затруднено — оно требует более сложной архитектуры центрального процессора, «умеющего» выполнять не только сложение, но и вычитание, а также «знающего» особый алгоритм обработки не имеющего «веса» знакового разряда. Этих трудностей позволяет избежать использование дополнительного кода.

Чтобы понять сущность дополнительного кода, рассмотрим работу реверсивного счётчика, последовательность показаний которого можно представить в виде замкнутого кольца из чисел (рис. 3.5).

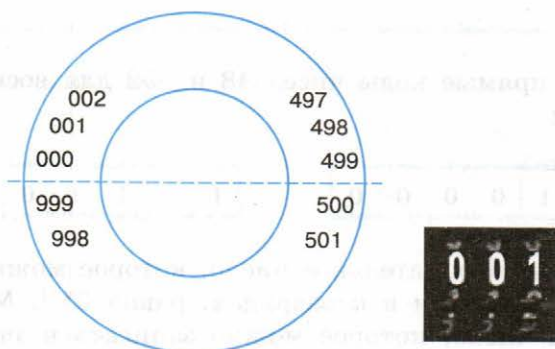


Рис. 3.5. Реверсивный счётчик

При возрастании показаний счётчика до максимального, например до 999, следующими его состояниями должны быть 1000, 1001, 1002 и т. д. Но для изображения старшей единицы в счётчике не хватает разряда, происходит переполнение разрядной сетки. Поэтому мы увидим 000, 001, 002 и т. д.

При убывании показаний счётчика после состояния 000 будут идти 999, 998, 997 и т. д. Но после достижения нуля последовательное вычитание единицы должно давать -1 , -2 , -3 и т. д.

Будем рассматривать числа 999, 998, 997 как коды чисел -1 , -2 , -3 и проверим на их примере соотношение: $y + (-y) = 0$:

$$001 + 999 = 1000;$$

$$002 + 998 = 1000;$$

$$003 + 997 = 1000.$$

С учётом того что единица переполнения теряется, мы, сложив число и код противоположного ему числа, получаем ноль!

Вот ещё несколько примеров:

$$5 - 2 = 5 + [-2] = 5 + 998 = 1003;$$

$$7 - 5 = 7 + [-5] = 7 + 995 = 1002.$$

Для устранения неоднозначности в кольце будем считать половину состояний (0–499) кодами нуля и положительных чисел, а оставшуюся половину (500–999) — кодами отрицательных чисел.

Таким образом, дополнительный код положительного числа совпадает с этим числом, а для отрицательного числа он равен дополнению его величины до числа q^n , возникающего при переполнении разрядной сетки. Здесь q — основание системы счисления, n — число разрядов в разрядной сетке.

Рассмотрим алгоритм получения дополнительного n -разрядного кода отрицательного числа:

- 1) модуль числа представить прямым кодом в n двоичных разрядах;
- 2) значения всех разрядов инвертировать (все нули заменить единицами, а единицы — нулями);
- 3) к полученному представлению, рассматриваемому как n -разрядное неотрицательное двоичное число, прибавить единицу.

Пример 1. Найдём 16-разрядный дополнительный код отрицательного числа -2017_{10} .

Прямой код $ -2017_{10} = 2017_{10}$	00000111 11100001 ₂
Инвертирование	11111000 00011110 ₂
Прибавление единицы	+ 1
Дополнительный код	11111000 00011111 ₂

Использование дополнительного кода позволяет свести операцию вычитания чисел к операции поразрядного сложения кодов этих чисел.

Пример 2. Как известно, $48 - 2017 = -1969$.

Выполним эту операцию в 16-разрядных машинных кодах.

Нам потребуются прямой код числа 48 и дополнительный код числа -2017 .

Прямой код числа 48	00000000 00110000 ₂
Дополнительный код числа -2017	11111000 00011111 ₂
Поразрядное сложение кодов	11111000 01001111 ₂

Рассмотрим полученный результат. Это отрицательное число (об этом говорит 1 в знаковом разряде), представленное в дополнительном коде. Перейдём к прямому коду модуля соответствующего числа, по которому сможем восстановить десятичное представление результата.

Прямой код можно получить из дополнительного кода, если применить к нему операцию инвертирования и прибавить единицу.

Инвертирование	00000111 10110000 ₂
Прибавление единицы	+ 1
Прямой код модуля результата	00000111 10110001 ₂

Получаем: $-11110110001_2 = -1969$.

13.2. Представление вещественных чисел

В математике множество вещественных чисел непрерывно, бесконечно и не ограничено.

Попробуйте обосновать это утверждение.

Вещественные числа записываются в естественной или в экспоненциальной форме.

В жизни мы чаще пользуемся естественной формой записи чисел, при которой: число представляется последовательностью десятичных цифр со знаком плюс или минус, знак плюс может опускаться, для разделения целой и дробной частей числа используется запятая. Например: 12,34; 0,0056; -708,9.

В экспоненциальной форме вещественное число a представляется как $a = \pm m \cdot q^p$, где m — мантисса числа, q — основание системы счисления, p — порядок числа.

Например, длину некоторого отрезка, равного 47,8 см, можно записать так:

- 1) $478 \cdot 10^{-1}$ см;
- 2) $47,8 \cdot 10^0$ см;
- 3) $4,78 \cdot 10^1$ см;
- 4) $0,478 \cdot 10^2$ см;
- 5) $0,000478 \cdot 10^5$ см.

Такое многообразие вариантов записи в экспоненциальной форме одного и того же числа не всегда удобно. Для однозначного представления вещественных чисел в компьютере используется нормализованная форма.

Нормализованная запись отличного от нуля вещественного числа¹⁾ — это запись вида $a = \pm m \cdot q^p$, где p — целое число (положительное, отрицательное или ноль), m — дробь, целая часть которой содержит одну значащую (ненулевую) цифру, т. е. $1 \leq m < q$.

Примеры нормализации чисел:

- 1) $31,415926 = 3,1415926 \cdot 10^1$;
- 2) $1000 = 1,0 \cdot 10^3$;
- 3) $0,123456789 = 1,23456789 \cdot 10^{-1}$;
- 4) $0,0000107_8 = 1,07_8 \cdot 10_8^{-5}$;
- 5) $1000,0001_2 = 1,0000001_2 \cdot 10_2^{11}$;
- 6) $AB,CDEF_{16} = A,BCDEF_{16} \cdot 10_{16}^1$.

Диапазон вещественных чисел в памяти компьютера очень широк, но, тем не менее, ограничен. Множество вещественных чисел, которые могут быть представлены в компьютере, конечно.

Поясним это на примере калькулятора, который производит вычисления в десятичной системе счисления. Пусть это будет калькулятор с десятью знаковыми местами на дисплее:

- 6 знаковых мест отводится под мантиссу (одно знаковое место отводится под знак мантиссы, четыре — под цифры мантиссы, одно — под точку, разделяющую целую и дробную части мантиссы);
- одно знаковое место отводится под символ «Е»;
- три знаковых места отводятся под порядок (одно — под знак порядка, два — под цифры порядка).



1) Стандарт IEEE 754.

У калькуляторов первая значащая цифра, с которой и начинается мантисса, изображается перед точкой.

Число 12,34 в таком калькуляторе будет представлено как +1.234E+01.

Число 12,35 будет представлено как +1.235E+01.

Как известно, между числами 12,34 и 12,35 находится бесконечное множество вещественных чисел, например: 12,341; 12,3412; 12,34123 и т. д.

Каждое из этих чисел в нашем калькуляторе будет представлено как +1.234E+01. Для последних разрядов у нас просто не хватает знакомест! Аналогичная ситуация имеет место и в компьютерном представлении вещественных чисел, независимо от того, ячейки какой разрядности там использованы.

Получается, что точно мы можем представить в компьютере лишь некоторую конечную часть множества вещественных чисел, а остальные числа — лишь приближённо.

Таким образом, множество вещественных чисел, представляемых в компьютере, дискретно, конечно и ограничено.

САМОЕ ГЛАВНОЕ

В математике множество целых чисел дискретно, бесконечно и не ограничено.

Для компьютерного представления целых чисел используется несколько различных способов, отличающихся друг от друга количеством разрядов (8, 16, 32 или 64 разряда) и наличием или отсутствием знакового разряда. В любом случае компьютерное представление целых чисел дискретно, конечно и ограничено.

В математике множество вещественных чисел непрерывно, бесконечно и не ограничено.

Для компьютерного представления вещественных чисел используется нормализованная запись вещественного числа $a = \pm m \cdot q^p$, где q — основание системы счисления, p — целое число (положительное, отрицательное или ноль), m — дробь, целая часть которой содержит одну значащую (ненулевую) цифру, т. е. $1 \leq m < q$.

Компьютерное представление вещественных чисел дискретно, конечно и ограничено.

Вопросы и задания



- Представьте в восьмиразрядном формате прямые коды десятичных чисел:
1) 64; 2) 58; 3) 72; 4) -96.
- Можно ли числа 43_{16} , 101010_2 , 129_{10} и -52_{10} сохранить в однобайтовом формате?
- Как определяется диапазон представления в компьютере целых чисел без знака? Со знаком?
- Почему множество целых чисел, представимых в памяти компьютера, дискретно, конечно и ограничено?
- Представьте в восьмиразрядном формате дополнительные коды двоичных чисел:
1) +1010; 2) -1001; 3) -11; 4) -11011.
- Найдите десятичные эквиваленты чисел, представленных в прямом коде:
1) 00000100; 2) 00001001; 3) 10000011; 4) 10000110.
- *7. Найдите десятичные эквиваленты чисел, представленных в дополнительном коде:
1) 00000100; 2) 11111001.
- Для хранения целого числа со знаком в компьютере используется два байта. Сколько единиц содержит внутреннее представление числа -101, записанного:
1) в прямом коде;
2) в дополнительном коде?
- Вычислите с помощью калькулятора (приложение Windows) в режиме «Программист» следующие примеры:
1) $111011101_2 - 1101110110_2$;
2) $1101101001_2 - 11000100100_2$.
Как вы можете объяснить полученные результаты?
- Запишите десятичные числа в нормализованной форме:
1) 217,934; 2) 75321; 3) 10,0101; 4) 200450.
- Сравните следующие числа:
1) $318,4785 \cdot 10^9$ и $3,184785 \cdot 10^{11}$;
2) $218,4785 \cdot 10^{-3}$ и $1847,85 \cdot 10^{-4}$.
- Выполните операцию сложения:
1) $0,397621 \cdot 10^3 + 0,2379 \cdot 10^1$;
2) $0,251452 \cdot 10^{-3} + 0,125111 \cdot 10^{-2}$.
- Чем ограничивается диапазон представимых в памяти компьютера вещественных чисел?



14. Почему множество вещественных чисел, представимых в памяти компьютера, дискретно, конечно и ограничено?
- *15. Попробуйте самостоятельно сформулировать основные принципы представления данных в компьютере.

§ 14

Кодирование текстовой информации

Компьютеры третьего поколения «научились» работать с текстовой информацией.

Текстовая информация по своей природе дискретна, т. к. представляется последовательностью отдельных символов.

Для компьютерного представления текстовой информации достаточно:

- 1) определить множество всех символов (алфавит), требуемых для представления текстовой информации;
- 2) выстроить все символы используемого алфавита в некоторой последовательности (присвоить каждому символу алфавита свой номер);
- 3) получить для каждого символа n -разрядный двоичный код ($n \leq 2^n$), переведя номер этого символа в двоичную систему счисления.

В памяти компьютера хранятся специальные кодовые таблицы, в которых для каждого символа указан его двоичный код. Все кодовые таблицы, используемые в любых компьютерах и любых операционных системах, подчиняются международным стандартам кодирования символов.

14.1. Кодировка ASCII и её расширения

Основой для компьютерных стандартов кодирования символов послужил код ASCII (American Standard Code for Information Interchange) — американский стандартный код для обмена информацией, разработанный в 1960-х годах в США и применявшийся для любых, в том числе и некомпьютерных, способов передачи информации (телеграф, факсимильная связь и т. д.). Этот код

7-битовый: общее количество символов составляет $2^7 = 128$, из них первые 32 символа — управляющие, а остальные — изображаемые, т. е. имеющие графическое изображение. К изображаемым символам в ASCII относятся буквы латинского алфавита (прописные и строчные), цифры, знаки препинания и арифметических операций, скобки и некоторые специальные символы. Кодировка ASCII приведена в табл. 3.8.

Таблица 3.8

Кодировка ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Хотя для кодирования символов в ASCII достаточно 7 битов, в памяти компьютера под каждый символ отводится ровно 1 байт (8 битов), при этом код символа помещается в младшие биты, а в старший бит заносится 0.

Например, 01000001 — код прописной латинской буквы «А»; с помощью шестнадцатеричных цифр его можно записать как 41.

Стандарт ASCII рассчитан на передачу только английского текста. Со временем возникла необходимость кодирования и неанглийских букв. Во многих странах для этого стали разрабатывать расширения ASCII-кодировки, в которых применялись однобайтовые коды символов. При этом первые 128 символов кодовой таблицы совпадали с кодировкой ASCII, а остальные (со 128-го по 255-й) использовались для кодирования букв национального алфавита, символов национальной валюты и т. п. Из-за несогласованности этих разработок для многих языков было создано несколько вариантов кодовых таблиц (например, для русского языка их было создано около десятка!).

Глава 3. Представление информации

Впоследствии использование кодовых таблиц было несколько упорядочено: каждой кодовой таблице было присвоено особое название и номер. Для русского языка наиболее распространёнными стали однобайтовые кодовые таблицы CP-866, Windows-1251 (табл. 3.9) и КОИ-8 (табл. 3.10). В них первые 128 символов совпадают с ASCII-кодировкой, а русские буквы размещены во второй части таблицы. Обратите внимание на то, что коды русских букв в этих кодировках различны.

Таблица 3.9

Кодировка Windows-1251

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ъ	Ѓ	,	ѓ	„	...	†	‡	€	‰	Ь	<	Ѣ	Ќ	Ѧ	Ѻ
9	ђ	'	'	“	”	•	-	—		™	ь	>	ѣ	ќ	ћ	ѻ
A		Ў	ў	Ј	ѡ	Ѓ	Ѕ	Ѕ	Ё	©	€	«	¬		®	Ї
B	°	±	І	і	г	μ	¶	·	ё	№	€	»	ј	ѕ	ѕ	ї
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Таблица 3.10

Кодировка КОИ-8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	—		г	г	L	J			T	⊥	+	■	■	■	■	■
9	▒	▒	▒		■	·	√	≈	≤	≥		Ј	°	²	·	÷
A	=		г	ё	г	г	г	г	г	г	г	г	г	г	г	г
B				Ё			т	т	т	±	⊥	⊥	†	†	†	©
C	ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
D	п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
E	Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
F	П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ

Мы выяснили, что при нажатии на алфавитно-цифровую клавишу в компьютер посылается некоторая цепочка нулей и единиц. В текстовых файлах хранятся не изображения символов, а их коды.

При выводе текста на экран монитора или принтера необходимо восстановить изображения всех символов, составляющих данный текст, причём изображения эти могут быть разнообразны и достаточно причудливы. Внешний вид выводимых на экран символов кодируется и хранится в специальных шрифтовых файлах. Современные текстовые процессоры умеют внедрять шрифты в файл. В этом случае файл содержит не только коды символов, но и описание используемых в этом документе шрифтов. Кроме того, файлы, создаваемые с помощью текстовых процессоров, включают в себя и такие данные о форматировании текста, как его размер, начертание, размеры полей, отступов, межстрочных интервалов и другую дополнительную информацию.

14.2. Стандарт Unicode

Ограниченность 8-битной кодировки, не позволяющей одновременно пользоваться несколькими языками, а также трудности, связанные с необходимостью преобразования одной кодировки в другую, привели к разработке нового кода. В 1991 году был разработан новый стандарт кодирования символов, получивший название Unicode (Юникод), позволяющий использовать в текстах любые символы любых языков мира.

Unicode — это «уникальный код для любого символа, независимо от платформы, независимо от программы, независимо от языка» (www.unicode.org).

В Unicode на кодирование символов отводится 31 бит. Первые 128 символов (коды 0–127) совпадают с таблицей ASCII. Далее размещены основные алфавиты современных языков: они полностью уместятся в первой части таблицы, их коды не превосходят $65\,536 = 2^{16}$.

Стандарт Unicode описывает алфавиты всех известных, в том числе и «мёртвых», языков. Для языков, имеющих несколько алфавитов или вариантов написания (например, японского и индийского), закодированы все варианты. В кодировку Unicode внесены все математические и иные научные символные обозначения и даже некоторые придуманные языки (например, язык эльфов из трилогии Дж. Р. Р. Толкина «Властелин колец»).

Всего современная версия Unicode позволяет закодировать более миллиона различных знаков, но реально используется чуть менее 110 000 кодовых позиций.

Для представления символов в памяти компьютера в стандарте Unicode имеется несколько кодировок.

В операционных системах семейства Windows используется кодировка UTF-16. В ней все наиболее важные символы кодируются с помощью 2 байт (16 бит), а редко используемые — с помощью 4 байт.

В операционной системе Linux применяется кодировка UTF-8, в которой символы могут занимать от 1 (символы, входящие в таблицу ASCII) до 4 байт. Если значительную часть текста составляют цифры и латинские буквы, то это позволяет в несколько раз уменьшить размер файла по сравнению с кодировкой UTF-16.

Кодировки Unicode позволяют включать в один документ символы самых разных языков, но их использование ведёт к увеличению размеров текстовых файлов.

14.3. Информационный объём текстового сообщения

Мы уже касались этого вопроса, рассматривая алфавитный подход к измерению информации.

Информационным объёмом текстового сообщения называется количество бит (байт, килобайт, мегабайт и т. д.), необходимых для записи этого сообщения путём заранее оговоренного способа двоичного кодирования.

Оценим в байтах объём текстовой информации в современном словаре иностранных слов из 740 страниц, если на одной странице размещается в среднем 60 строк по 80 символов (включая пробелы).

Будем считать, что при записи используется кодировка «один символ — один байт». Количество символов во всем словаре равно:

$$80 \cdot 60 \cdot 740 = 3\,552\,000.$$

Следовательно, объём равен

$$3\,552\,000 \text{ байт} = 3\,468,75 \text{ Кбайт} \approx 3,39 \text{ Мбайт.}$$

Если же использовать кодировку UTF-16, то объём этой же текстовой информации в байтах возрастёт в 2 раза и составит 6,78 Мбайт.

САМОЕ ГЛАВНОЕ

Текстовая информация по своей природе дискретна, т. к. представляется последовательностью отдельных символов.

В памяти компьютера хранятся специальные кодовые таблицы, в которых для каждого символа указан его двоичный код. Все кодовые таблицы, используемые в любых компьютерах и любых операционных системах, подчиняются международным стандартам кодирования символов.

Основой для компьютерных стандартов кодирования символов послужил код ASCII, рассчитанный на передачу только английского текста. Расширения ASCII — кодировки, в которых первые 128 символов кодовой таблицы совпадают с кодировкой ASCII, а остальные (со 128-го по 255-й) используются для кодирования букв национального алфавита, символов национальной валюты и т. п.

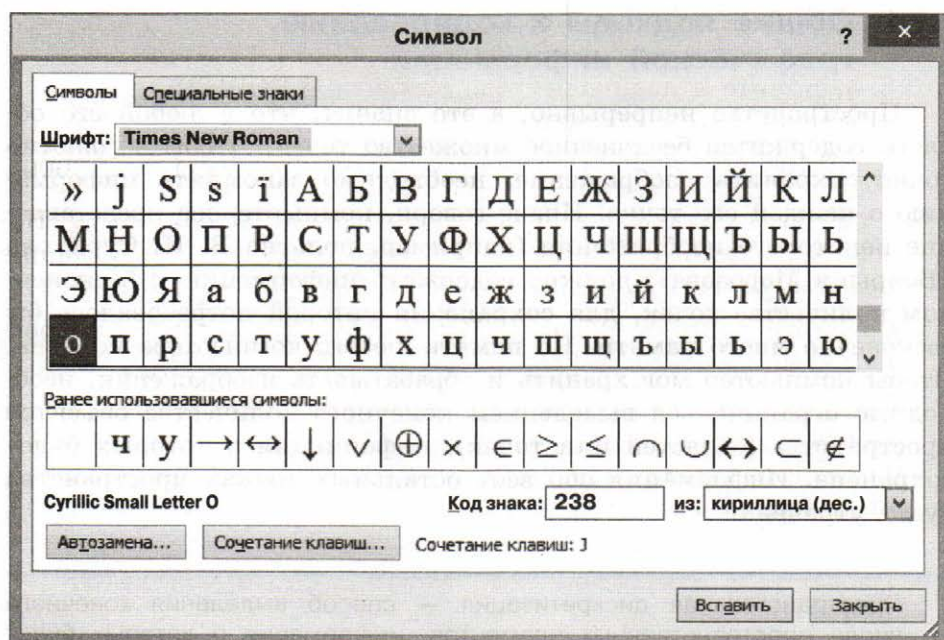
В 1991 году был разработан новый стандарт кодирования символов, получивший название Unicode (Юникод), позволяющий использовать в текстах любые символы любых языков мира. Кодировки Unicode позволяют включать в один документ символы самых разных языков, но их использование ведёт к увеличению размеров текстовых файлов.

Вопросы и задания

1. Какова основная идея представления текстовой информации в компьютере?
2. Что представляет собой кодировка ASCII? Сколько символов она включает? Какие это символы?
3. Как известно, кодовые таблицы каждому символу алфавита ставят в соответствие его двоичный код. Как, в таком случае, вы можете объяснить вид таблицы 3.8 «Кодировка ASCII»?

4. С помощью таблицы 3.8:
- 1) декодируйте сообщение 64 65 73 6B 74 6F 70;
 - 2) запишите в двоичном коде сообщение TOWER;
 - 3) декодируйте сообщение
01101100 01100001 01110000 01110100 01101111 01110000
5. Что представляют собой расширения ASCII-кодировки? Назовите основные расширения ASCII-кодировки, содержащие русские буквы.
6. Сравните подходы к расположению русских букв в кодировках Windows-1251 и КОИ-8.
7. Представьте в кодировке Windows-1251 текст «Знание — сила!»:
- 1) шестнадцатеричным кодом;
 - 2) двоичным кодом;
 - 3) десятичным кодом.
8. Представьте в кодировке КОИ-8 текст «Дело в шляпе!»:
- 1) шестнадцатеричным кодом;
 - 2) двоичным кодом;
 - 3) десятичным кодом.
9. Что является содержимым файла, созданного в современном текстовом процессоре?
10. В кодировке Unicode на каждый символ отводится 2 байта. Определите в этой кодировке информационный объем следующей строки:
Где родился, там и содился.
11. Набранный на компьютере текст содержит 2 страницы. На каждой странице 32 строки, в каждой строке 64 символа. Определите информационный объем текста в кодировке Unicode, в которой каждый символ кодируется 16 битами.
12. Текст на русском языке, первоначально записанный в 8-битовом коде Windows, был перекодирован в 16-битную кодировку Unicode. Известно, что этот текст был распечатан на 128 страницах, каждая из которых содержала 32 строки по 64 символа в каждой строке. Каков информационный объем этого текста?
13. В текстовом процессоре MS Word откройте таблицу символов (вкладка Вставка → Символ → Другие символы):





В поле **Шрифт** установите **Times New Roman**, в поле **из** — **кириллица (дес.)**.

Вводя в поле **Код знака** десятичные коды символов, декодируйте сообщение:

196	238	240	238	227	243	32
238	241	232	235	232	242	32
232	228	243	249	232	233	46


§ 15

Кодирование графической информации


Обработка и хранение графической информации требуют значительных вычислительных ресурсов, которые появились только у компьютеров четвёртого поколения.

15.1. Общие подходы к кодированию графической информации

Пространство непрерывно, а это значит, что в любой его области содержится бесконечное множество точек. Чтобы абсолютно точно сохранить изображение, необходимо запомнить информацию о каждой его точке. Иначе говоря, компьютерное представление некоторого изображения (например, полотна В. И. Сурикова «Боярыня Морозова») должно содержать информацию о бесконечном количестве точек, для сохранения которой потребовалось бы бесконечно много памяти. Но память любого компьютера конечна. Чтобы компьютер мог хранить и обрабатывать изображения, необходимо ограничиться выделением конечного количества объектов пространства (областей или точек), информация о которых будет сохранена. Информация обо всех остальных точках пространства будет утрачена.

 Пространственная дискретизация — способ выделения конечного числа пространственных элементов, информация о которых будет сохранена в памяти компьютера.

Цвет и яркость — характеристики, присущие каждому элементу (точке, области) изображения. Их можно измерять, т. е. выражать в числах. И цвет, и яркость — непрерывные величины, результаты измерения которых следует выражать вещественными числами. Но вам известно, что вещественные числа не могут быть представлены в компьютере точно.

 Квантование — процедура преобразования непрерывного диапазона всех возможных входных значений измеряемой величины в дискретный набор выходных значений.

При квантовании диапазон возможных значений измеряемой величины разбивается на несколько поддиапазонов. При измерении определяется поддиапазон, в который попадает значение, и в компьютере сохраняется только номер поддиапазона.

Дискретизация и квантование всегда приводят к потере некоторой доли информации.

15.2. Векторная и растровая графика

В зависимости от способа формирования графических изображений выделяют векторный и растровый методы кодирования графических изображений.

Векторное изображение строится из отдельных базовых объектов — графических примитивов: отрезков, многоугольников, кривых, овалов. Способ создания векторных изображений напоминает аппликацию (рис. 3.6).

Графические примитивы характеризуются цветом и толщиной контура, цветом и способом заливки внутренней области, размером и т. д. При сохранении векторного изображения в память компьютера заносится информация о составляющих его графических примитивах.

Например, для построения окружности необходимо сохранить такие исходные данные, как координаты её центра, значение радиуса, цвет и толщину контура, цвет заполнения. При этом и большая, и маленькая окружности будут описаны одним и тем же набором данных, т. е. реальные размеры объекта не оказывают никакого влияния на размер сохраняемых о нём данных.

Фактически векторное представление — это описание, в соответствии с которым происходит построение требуемого изображения. Такого рода описания представляются в компьютере как обычная текстовая информация.

Растровое графическое изображение состоит из отдельных маленьких элементов — пикселей (pixel — аббревиатура от англ. *picture element* — элемент изображения). Оно похоже на мозаику (рис. 3.7), изготовленную из одинаковых по размеру объектов (разноцветных камешков, кусочков стекла, эмали и др.).

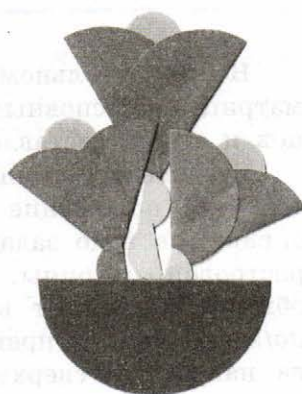


Рис. 3.6. Аппликация из бумаги



Рис. 3.7. Фрагмент мозаичного полотна на станции московского метро «Маяковская»



Растр — организованная специальным образом совокупность пикселей, представляющая изображение. Координаты, форма и размер пикселей задаются при определении растра. Изменяемым атрибутом пикселей является цвет.

В прямоугольном растре пиксели составляют прямоугольную матрицу, её основными параметрами являются количество столбцов и строк, составленных из пикселей.

Главное преимущество прямоугольных растров заключается в том, что положение каждого пикселя на изображении (или на экране) не надо задавать — его легко вычислить, зная размеры растровой матрицы, плотность размещения пикселей, которую обычно указывают в количестве точек на дюйм (*dpi*, от англ. *dots per inch*), и правила перечисления пикселей (например, слева направо и сверху вниз: сначала слева направо нумеруются все пиксели в верхней строке, затем нумерация продолжается на следующей строке, лежащей ниже, и т. д.).

Итак, мы выяснили, как происходит пространственная дискретизация, позволяющая выделить конечное число пространственных элементов, информация о которых будет сохранена в памяти компьютера.

Остаётся рассмотреть вопросы кодирования цвета каждого пространственного элемента.

15.3. Кодирование цвета



Из курса физики вам известно, что цвет — это ощущение, которое возникает у человека при воздействии на его зрительный аппарат электромагнитного излучения с длиной волны в диапазоне от 380 до 760 нм¹⁾. В табл. 3.11 показана зависимость цвета от длины волны видимого спектра.

Учёным долгое время не удавалось объяснить процесс цветовосприятия. Первые серьёзные результаты в этой области были получены Исааком Ньютоном (1643–1727), который описал составную природу белого света, выделив в его спектре семь основных (наиболее заметных) цветов — красный, оранжевый, жёлтый, зелёный, голубой, синий, фиолетовый.

¹⁾ Нанометр (нм) — единица измерения длины в Международной системе единиц (СИ), равная одной миллиардной части метра (т. е. 10^{-9} м).

Таблица 3.11

Зависимость цвета от длины волны видимого спектра

Длина волны, нм	Цвет
620–760	красный
585–620	оранжевый
575–585	жёлтый
550–575	жёлто-зелёный
510–550	зелёный
480–510	голубой (сине-зелёный)
450–480	синий
380–450	фиолетовый

Позднее, в 1756 году, выдающийся русский учёный М. В. Ломоносов (1711–1765), исследуя вопросы окрашивания стёкол, обнаружил, что для придания стеклу любого цветового оттенка достаточно использовать всего три основные краски, смешивая их в определённых пропорциях. Спустя столетие эти факты были теоретически обобщены немецким учёным Германом Грассманом (1809–1877), сформулировавшим законы синтеза цвета. Наиболее важными из них для понимания сути цвето-воспроизведения и цветового кодирования являются следующие два закона.

Закон трёхмерности: с помощью трёх линейно независимых цветов можно однозначно выразить любой цвет. Цвета считаются линейно независимыми, если никакой из них нельзя получить путём смешения остальных.

Закон непрерывности: при непрерывном изменении пропорции, в которой взяты компоненты цветовой смеси, получаемый цвет также меняется непрерывно.

На основании законов Грассмана можно сделать вывод, что любому цвету однозначно соответствует определённая точка трёхмерного пространства. Действительно, каждая цветовая модель задаёт некоторую систему координат, в которой основные цвета модели играют роль осей, а цвета можно рассматривать как точки или векторы в трёхмерном цветовом пространстве.

В компьютерной технике чаще всего используются следующие цветовые модели:

- RGB (Red — Green — Blue, красный — зелёный — синий);
- HSB (Hue — Saturation — Brightness, цветовой оттенок — насыщенность — яркость);
- CMYK (Cyan — Magenta — Yellow — black, голубой — пурпурный — жёлтый — чёрный).

15.4. Цветовая модель RGB

Для синтеза цвета на экранах компьютеров, телевизоров, смартфонов и других устройств отображения графической информации выбраны красный, зелёный и синий цвета. Смешением трёх основных цветов синтезируются все остальные цвета. Такая модель называется аддитивной (аддитивный — получаемый путём сложения, от лат. *additio* — прибавление).

Выбор этих цветов неслучаен и кроется в физиологии человеческого зрения. Действительно, в сетчатке глаза человека есть три типа колбочек, максимумы чувствительности которых приходятся на красный, зелёный и синий участки спектра (табл. 3.12).

Таблица 3.12
Спектральная чувствительность человеческого глаза

Тип колбочек	Диапазон длин волн, нм	Максимум чувствительности, нм
Красные	550–760	≈ 610
Зелёные	450–650	≈ 550
Синие	380–550	≈ 450

Модель RGB (рис. 3.8) определяет пространство цветов в виде единичного куба с осями R (красная компонента), G (зелёная компонента), B (синяя компонента).

Рассмотрим эту модель подробно.

1. Любая точка куба (r ; g ; b) определяет некоторый цвет.
2. Точка $(0; 0; 0)$ соответствует чёрному цвету; точка $(1; 1; 1)$ соответствует белому цвету; линия $(0; 0; 0)$ – $(1; 1; 1)$ описывает все градации серого от чёрного до белого.

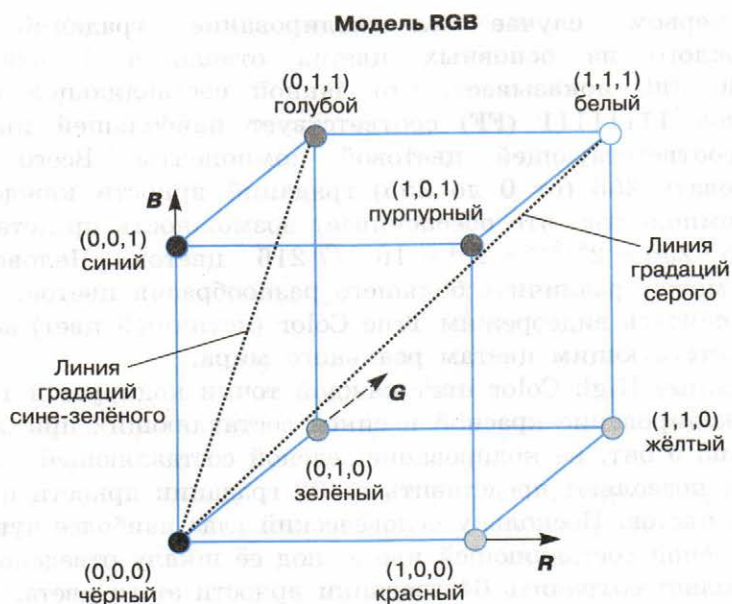


Рис. 3.8. Цветовой куб для RGB-кодирования

3. При движении по прямой от точки $(0; 0; 0)$ к точке $(r; g; b)$ получаем все градации яркости цвета $(r; g; b)$ — от самого тёмного до самого яркого.
4. На гранях куба $(r = 0)$, $(g = 0)$ и $(b = 0)$ расположены самые насыщенные цвета. Чем ближе точка к главной диагонали $(0; 0; 0)$ – $(1; 1; 1)$, тем менее насыщен соответствующий цвет. Если все три координаты точки $(r; g; b)$ ненулевые, то цвет — ненасыщенный.

Мы уже упоминали о квантовании. Благодаря ему в компьютере интенсивность цветовых компонент определяется не вещественными (из интервала $[0; 1]$), а целыми числами. Для этого «единичные» отрезки на «красной», «зелёной» и «синей» осях разбивают на несколько поддиапазонов, каждый из которых получает свой номер. Цвет пикселя изображения, которое будет выведено на экран, задаётся тремя целыми неотрицательными числами — номерами поддиапазонов, в которые «попадает» этот цвет. Качество цветопередачи зависит от количества выделяемых поддиапазонов.

В современных компьютерах для сохранения информации о цвете каждой точки цветного изображения в модели RGB обычно отводится 3 (режим True Color) или 2 (режим High Color) байта.

В первом случае на кодирование градаций яркости каждого из основных цветов отводится 1 байт: код 00000000 (00) показывает, что данной составляющей нет вообще, код 11111111 (FF) соответствует наибольшей интенсивности соответствующей цветовой компоненты. Всего можно закодировать 256 (от 0 до 255) градаций яркости каждой цветовой компоненты, что обеспечивает возможность представления $256 \cdot 256 \cdot 256 = 2^{8+8+8} = 2^{24} = 16\,777\,216$ цветов. Человеческий глаз не может различить большего разнообразия цветов. Это позволяет считать видеорежим True Color (истинный цвет) абсолютным соответствующим цветам реального мира.

В режиме High Color цвет каждой точки кодируется 16 битами. На кодирование красной и синей составляющих при этом отводится по 5 бит, на кодирование зелёной составляющей — 6 бит. Пять бит позволяют представить по 32 градации яркости красного и синего цветов. Поскольку человеческий глаз наиболее чувствителен к зелёной составляющей цвета, под её шкалу отведено 6 бит, что позволяет сохранить 64 градации яркости этого цвета. Всего в рассматриваемом режиме можно представить $2^{5+6+5} = 2^{16} = 65\,536$ цветов. Цвета изображений, сохранённых в режиме High Color также вполне реалистичны и «комфортны» для человеческого глаза. При этом размеры соответствующих файлов в 1,5 раза меньше, чем при использовании режима True Color.

Глубина цвета (i) — количество бит, используемое для хранения и представления цвета при кодировании одного пикселя растровой графики или видеоизображения.

Палитра (N) — количество цветов, которые могут быть использованы для воспроизведения изображения.

Справедливо соотношение: $N = 2^i$.

Пример. Имеется канал связи, обеспечивающий передачу информации со скоростью 16 000 бит/с. Выясним, сколько секунд потребуется для передачи по этому каналу 256-цветного растрового изображения размером 800×600 пикселей.

Зная палитру — количество цветов, используемых для воспроизведения изображения, вычислим глубину цвета — количество бит для представления цвета одного пикселя.

$$256 = 2^i, \quad i = 8 \text{ (бит)}.$$

Размер файла с изображением равен: $800 \cdot 600 \cdot 8$ (бит).

Вычислим время передачи данного файла:

$$t = \frac{800 \cdot 600 \cdot 8}{16000} = 240 \text{ (с).}$$

15.5. Цветовая модель HSB

Модель RGB проста и понятна. Её применяют всегда, когда изображение подготавливают для воспроизведения на экране. Если изображение (например, цифровая фотография) проходит компьютерную обработку в графическом редакторе, то его тоже следует представить в этой модели.

В тех графических редакторах, которые ориентированы не на обработку готовых изображений, а на их создание своими руками, удобнее применять цветовую модель HSB. Человеку гораздо проще не синтезировать цвет из отдельных составляющих, а выбрать его, ориентируясь на более естественные параметры: цветовой оттенок (Что это за цвет?), насыщенность (Насколько цвет насыщенный?), яркость (Насколько цвет светлый или тёмный?). Именно эти три параметра и стали основой для модели HSB (Hue — Saturation — Brightness).

Цветовой оттенок (Hue) — один из цветов спектра. Конкретный цветовой оттенок кодируется либо величиной угла, либо длиной дуги на цветовом круге (рис. 3.9).



Рис. 3.9. Круговое расположение цветов

Насыщенность цвета (Saturation) характеризуется степенью разбавления его белым цветом. Например, если ярко-красную (насыщенную) краску разбавить белой, то её цветовой оттенок останется прежним, изменится только насыщенность. Чем больше добавляется белого цвета, тем меньше насыщенность конкретного цветового оттенка, а с увеличением насыщенности цвет становится «сочнее».

Яркость цвета (Brightness) зависит от добавления к нему чёрного цвета — чем больше чёрного цвета, тем меньше яркость.

Пространство цветов модели HSB может быть представлено в форме вложенных концентрических конусов с общей вершиной и общей осью симметрии (рис. 3.10).

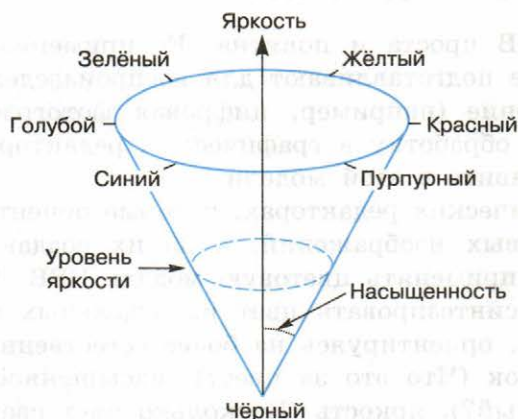


Рис. 3.10. Цветовая модель HSB

По окружности в основании конуса расположены цветовые оттенки. Цвета с одинаковым оттенком расположены в полуплоскости, проходящей через ось симметрии конуса.

Величина угла между осью конуса и образующей определяет насыщенность цвета. Насыщенность цвета возрастает с увеличением этого угла (т. е. с отдалением от оси конуса). Цвета с одинаковой насыщенностью расположены на конической поверхности с определённым углом при основании.

Ось конуса — это ось яркости. Вершина конуса соответствует чёрному цвету. Яркость цвета возрастает — с приближением к его основанию. Цвета с одинаковой яркостью расположены по кругу — сечению конуса плоскостью, перпендикулярной его оси.

15.6. Цветовая модель CMYK

Когда мы смотрим на изображение на экране монитора, мы видим излучаемый свет, а когда рассматриваем картинку на бумаге, то видим свет отражённый. Именно поэтому если смешать краски красного и зелёного цветов, то получится тёмно-коричневая краска, а не жёлтая, как это предполагается в RGB-модели.

Если под микроскопом рассмотреть цветные иллюстрации в книге, то можно увидеть, что они напечатаны очень маленькими, частично перекрывающимися цветными точками — офсетам. Офсетам хорошо видны на границах цветной печати и в местах с бледной краской.

Для подготовки печатных изображений используется цветовая модель СМУК (Cyan — Magenta — Yellow — black), базовыми цветами которой являются голубой, пурпурный и жёлтый цвета. Основные цвета этой модели подобраны так, чтобы соответствующие краски поглощали свет в достаточно узкой области спектра: голубая краска сильно поглощает красный цвет, пурпурная — зелёный, а жёлтая — синий.

Можно сказать, что цветовые компоненты модели СМУК получаются в результате вычитания основных цветов из белого:

- голубой = белый — красный = зелёный + синий (0; 255; 255);
- пурпурный = белый — зелёный = красный + синий (255; 0; 255);
- жёлтый = белый — синий = красный + зелёный (255; 255; 0).

Эти три цвета называются дополнительными, потому что они дополняют основные цвета до белого. Сама же модель СМУК называется субтрактивной (вычитающей) моделью.

Пространство цветовой модели СМУК также можно представить единичным кубом, где плотность закрашки (или яркость базовых цветов) — это вещественные числа в диапазоне от 0 до 1 (рис. 3.11).

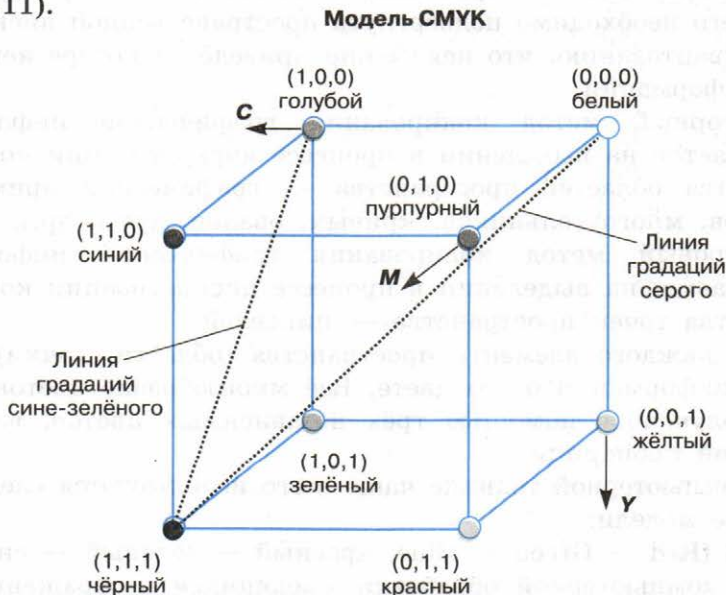


Рис. 3.11. Цветовой куб для СМУК-кодирования

Рассмотрим особенности модели СМУК:

- любая точка куба $(c; m; y)$ определяет некоторый цвет;
- точка $(0; 0; 0)$ соответствует белому цвету, точка $(1; 1; 1)$ — чёрному, а линия $(0; 0; 0)$ – $(1; 1; 1)$ описывает все оттенки серого цвета от белого до чёрного;
- при движении по прямой от $(0; 0; 0)$ к точке $(c; m; y)$ получаем все градации яркости цвета $(c; m; y)$ — от самой яркой до самой тёмной.

В идеальном случае голубого, пурпурного и жёлтого цветов было бы достаточно для формирования на бумаге любого цвета. Однако реально существующие краски не идеальны, они не могут поглотить цветовые компоненты полностью. Если нанести все три краски на бумагу, то вместо чёрного получится тёмно-серый цвет. Поэтому, чтобы скорректировать цветовую гамму, используют четвёртую краску — чёрную. Именно поэтому в название цветовой модели СМУК добавлена буква К (black).

САМОЕ ГЛАВНОЕ

Графическая информация, так же как числовая и текстовая, хранится в памяти компьютера в двоичном коде. Для представления графического изображения в памяти компьютера предварительно его необходимо подвергнуть пространственной дискретизации и квантованию, что неизбежно приведёт к потере некоторой доли информации.

Векторный метод кодирования графической информации основывается на выделении в процессе дискретизации конечного количества областей пространства — графических примитивов (отрезков, многоугольников, кривых, овалов, дуг и др.).

Растровый метод кодирования графической информации основывается на выделении в процессе дискретизации конечного количества точек пространства — пикселей.

Для каждого элемента пространства (области, точки) сохраняется информация о его цвете. Всё многообразие цветов может быть получено с помощью трёх независимых цветов, взятых в некоторой пропорции.

В компьютерной технике чаще всего используются следующие цветовые модели:

- RGB (Red — Green — Blue, красный — зелёный — синий) — для компьютерной обработки имеющихся изображений, при подготовке изображений для воспроизведения на экране;

- HSB (Hue — Saturation — Brightness, цветовой оттенок — насыщенность — яркость) — при создании изображений с помощью инструментов графического редактора;
- CMYK (Cyan — Magenta — Yellow — black, голубой — пурпурный — жёлтый — чёрный) — для подготовки печатных изображений.

Для сохранения информации о цвете каждой точки цветного изображения в модели RGB отводится 24 (режим True Color) или 16 бит (режим High Color).

Количество бит, используемое для хранения и представления цвета при кодировании одного пикселя растровой графики или видеоизображения, называется глубиной цвета (i). Палитра (N) — количество цветов, которые могут быть использованы для воспроизведения изображения. Справедливо соотношение: $N = 2^i$.

В режиме High Color может быть представлено 65 536 разных цветов, а в режиме True Color — 16 777 216. Такие цветовые палитры вполне реалистичны и «комфортны» для человеческого глаза.

Вопросы и задания

1. Почему при кодировании графических изображений всегда происходит потеря некоторого количества информации?
2. В чём суть векторного кодирования информации?
3. В чём суть растрового кодирования информации?
4. Какова физическая природа света?
5. Сформулируйте законы, наиболее важные для понимания сути цветовоспроизведения и цветового кодирования.
6. В чём состоит суть цветовой модели RGB?
7. Определите требуемый объём видеопамати при заданных разрешении монитора и глубине цвета.

Разрешение монитора	Глубина цвета		
	8	16	24
1024 × 768			
1280 × 720			
1600 × 900			
1920 × 1080			

8. Для хранения растрового изображения размером 128×128 пикселей отвели 16 Кбайт памяти. Каково максимально возможное число цветов в палитре изображения?
9. Определите объём видеопамати компьютера, который необходим для реализации графического режима монитора True Color с разрешающей способностью 1024×768 точек.
10. В цветовой модели RGB для кодирования одного пикселя используется 3 байта. Фотографию размером 2048×1536 пикселей сохранили в виде несжатого файла с использованием RGB-кодирования. Определите размер файла.
11. Укажите минимальный объём памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 64×64 пикселя, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно.
12. Вы хотите работать с разрешением монитора 1920×1080 пикселей, используя 16 777 216 цветов. В магазине продаются видеокарты с памятью 512 Кбайт, 2 Мбайта, 4 Мбайта и 64 Мбайта. Какую из них можно купить для вашей работы?
13. В процессе преобразования растрового графического изображения количество цветов уменьшилось с 16 777 216 до 256. Во сколько раз уменьшится объём занимаемой им памяти?
14. Пусть используется режим High Color. Назовите цвет, который задаётся кодом:
1) 1111100000011111; 2) 0111101111101111.
15. Для кодирования цвета фона интернет-страницы используется атрибут `bgcolor="#XXXXXX"`, где XXXXXX — шестнадцатизначное значение интенсивности цветовых компонент в 24-битной RGB-модели.
Назовите цвет страниц, заданных тегами:
1) `<body bgcolor="#FFFFFF">`;
2) `<body bgcolor="#00FF00">`;
3) `<body bgcolor="#0000FF">`;
4) `<body bgcolor="#EEEE00">`;
5) `<body bgcolor="#A5A5A5">`.
16. Выясните, каким образом распределено количество битов по красной, зелёной и синей составляющим в 8-битной цветовой схеме. С чем связано такое распределение?

17. Назовите факторы, существенные для построения моделей цветопередачи RGB и HSB. Какими характеристиками цвета оперирует модель цветопередачи HSB?
18. Исследуйте окно «Изменение палитры» в графическом редакторе Paint. Какие модели конструирования цвета там представлены?
19. Краски каких цветов используются в цветном принтере? Почему для печати на цветном принтере нельзя использовать краски красного, зелёного и синего цветов?
20. Почему модель RGB считается аддитивной, а модель CMYK — субтрактивной цветовой моделью?

§ 16

Кодирование звуковой информации

16.1. Звук и его характеристики

Звук — это распространяющиеся в воздухе, воде или другой среде волны с непрерывно меняющейся амплитудой и частотой (рис. 3.12).

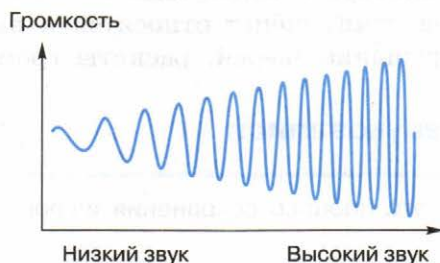


Рис. 3.12. Звуковая волна

Амплитуду звуковых колебаний называют звуковым давлением или силой звука. Эта величина характеризует воспринимаемую громкость звука. Абсолютную величину звукового давления измеряют в единицах давления — паскалях (Па). Самые слабые, едва различимые звуки имеют амплитуду около 20 мкПа ($2 \cdot 10^{-5}$ Па, так называемый порог слышимости). Самые сильные звуки, не

выводящие слуховые органы из строя, могут иметь амплитуду до 200 Па (так называемый болевой порог).

На практике вместо абсолютной используют относительную силу (уровень) звука, измеряемую в децибелах (дБ). Вот некоторые значения уровня звука:

Порог слышимости	0 дБ
Шелест листьев, дыхание	10–20 дБ
Шёпот	20–30 дБ
Разговор средней громкости	50–60 дБ
Шумная улица	70–80 дБ
Громкая музыка	100–120 дБ
Самолёт на взлёте	120–130 дБ
Болевой порог	140 дБ

Частота определяется как количество колебаний в секунду и выражается в герцах (Гц). Чем больше частота, тем выше звук, и наоборот. Человек способен слышать звук в широком частотном диапазоне, но важное для жизни значение имеют только звуки от 125 до 8000 Гц. Например, звуковые волны в диапазоне 500–4000 Гц соответствуют человеческому голосу. Звучание детского голоса, пение птиц, шёпот относятся к высоким частотам. Звук контрабаса, рычание зверей, раскаты грома — к низким.

16.2. Понятие звукозаписи

Звукозапись — это процесс сохранения информации о параметрах звуковых волн.



Способы записи звука разделяются на аналоговые и цифровые. При аналоговой записи на носителе размещается непрерывный «слепок» звуковой волны. Так, на грампластинке пропечатывается непрерывная канавка, изгибы которой повторяют амплитуду и частоту звука.

16.3. Оцифровка звука

Чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму. Для этого его подвергают временной дискретизации и квантованию: параметры звукового сигнала измеряются не непрерывно, а через определённые промежутки времени (временная дискретизация); результаты измерений записываются в цифровом виде с ограниченной точностью (квантование).

Вообще говоря, в компьютер приходит не сам звук, а электрический сигнал, снимаемый с какого-либо устройства: например, микрофон преобразует звуковое давление в электрические колебания, которые в дальнейшем и обрабатываются.

Если записывается стереозвук (ведётся двухканальная запись), то оцифровке подвергается не один электрический сигнал, а сразу два и, следовательно, количество сохраняемой цифровой информации удваивается.

Сущность временной дискретизации заключается в том, что аналоговый звуковой сигнал разбивается на отдельные маленькие временные участки и для каждого такого участка устанавливается определённая величина интенсивности звука (рис. 3.13). Другими словами, через какие-то промежутки времени мы измеряем уровень аналогового сигнала. Количество таких измерений за одну секунду называется частотой дискретизации.

Частота дискретизации — это количество измерений громкости звука за одну секунду.

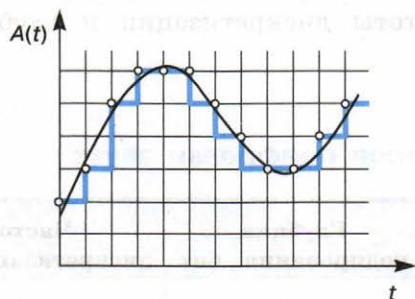


Рис. 3.13. Временная дискретизация звукового сигнала ($A(t)$ — амплитуда, t — время)

Частота дискретизации измеряется в герцах (Гц) и килогерцах (кГц). $1 \text{ кГц} = 1000 \text{ Гц}$. Частота дискретизации, равная 100 Гц, означает, что за одну секунду проводилось 100 измерений громкости звука.

Качество звукозаписи зависит не только от частоты дискретизации, но также и от глубины кодирования звука.

Глубина кодирования звука или разрешение — это количество информации, которое необходимо для кодирования дискретных уровней громкости цифрового звука.

В результате измерений звукового сигнала (см. рис. 3.13) на каждой его «ступеньке» будет получено некоторое значение громкости, при этом все результаты измерений будут лежать в некотором диапазоне.

Пусть под запись одного результата измерения громкости в памяти компьютера отведено n бит. Вы знаете, что это позволяет закодировать ровно 2^n разных результатов измерений. Так, при $n = 8$ можно закодировать 256 разных результатов измерений громкости звука. Поэтому весь диапазон, в котором могут находиться результаты измерений громкости звука, можно разбить на 256 разных поддиапазонов — уровней громкости звука, каждому из которых присвоить свой уникальный код. После этого каждый имеющийся результат измерений громкости звука можно соотнести с некоторым поддиапазоном, в который он попадает, и кодировать его номером (кодом) соответствующего уровня громкости.

В зависимости от ситуации на практике используются разные значения частоты дискретизации и глубины кодирования (табл. 3.13).

Таблица 3.13

Примеры параметров оцифровки звука

Название	Глубина кодирования, бит	Частота дискретизации, кГц	Число каналов
Телефонная связь	8	8	1
CD	16	44,1	2

Пример. Оценим объём звукового стереоаудиофайла с глубиной кодирования 16 бит и частотой дискретизации 44,1 кГц, который хранит звуковой фрагмент длительностью звучания 15 секунд.

Объём такого звукового фрагмента равен:

$$\begin{aligned} & 2 \text{ (канала)} \cdot 16 \text{ бит} \cdot 44\,100 \text{ Гц} \cdot 15 \text{ с} = \\ & = 2\,646\,000 \text{ байт} \approx 2\,584 \text{ Кбайта}. \end{aligned}$$

Увеличивая частоту дискретизации и глубину кодирования, можно более точно сохранить и впоследствии восстановить форму звукового сигнала. При этом объём сохраняемых данных будет увеличиваться.

Важно понимать, каких параметров оцифровки достаточно, чтобы сохраняемый звук был достаточно близок к исходному, а содержащий его файл имел минимально возможный объём. В начале 30-х годов прошлого века было установлено, что это возможно, если частота временной дискретизации будет в два раза выше максимальной частоты измеряемого сигнала.

В 1928 году американский учёный Гарри Найквист высказал утверждение, что частота дискретизации должна быть в два или более раза выше максимальной частоты измеряемого сигнала. В 1933 году наш соотечественник В. А. Котельников и независимо от него американец Клод Шеннон в 1949 году сформулировали и доказали теорему, более сильную чем утверждение Найквиста, о том, при каких условиях и как по дискретным значениям можно восстановить форму непрерывного сигнала.

САМОЕ ГЛАВНОЕ

Звук — это распространяющиеся в воздухе, воде или другой среде волны с непрерывно меняющейся амплитудой и частотой.

Чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть преобразован в цифровую дискретную форму. Для этого его подвергают временной дискретизации и квантованию: параметры звукового сигнала измеряются не непрерывно, а через определённые промежутки времени (временная дискретизация); результаты измерений записываются в цифровом виде с ограниченной точностью (квантование).

Таким образом, при оцифровке звука искажение сохраняемого сигнала происходит дважды: во-первых, при дискретизации теряется информация об истинном изменении звука между изме-

рениями, а во-вторых, при квантовании сохраняются не точные, а близкие к ним дискретные значения.

Объём оцифрованного звукового фрагмента в битах находится как произведение частоты дискретизации в Гц, глубины кодирования звука в битах, длительности звучания записи в секундах и количества каналов.



Вопросы и задания

1. Каким образом происходит преобразование непрерывного звукового сигнала в дискретный цифровой код?
2. Как частота дискретизации и глубина кодирования влияют на качество цифрового звука?
3. Производится четырёхканальная (квадро) звукозапись с частотой дискретизации 32 кГц и 32-битным разрешением. Запись длится 4 минуты, её результаты заносятся в файл, сжатие данных не производится. Определите приблизительно размер полученного файла (в мегабайтах). В качестве ответа укажите ближайшее к размеру файла целое число, кратное 10.
4. Музыкальный фрагмент был записан в формате моно, оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла — 49 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате стерео (двухканальная запись) и оцифрован с разрешением в 4 раза выше и частотой дискретизации в 3,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите в мегабайтах размер файла, полученного при повторной записи.
5. Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 32 секунды. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 3 раза выше и частотой дискретизации в 3 раза выше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б. Пропускная способность канала связи с городом Б в 2 раза выше, чем канала связи с городом А. Сколько секунд длилась передача файла в город Б?

- Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 96 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 4 раза выше и частотой дискретизации в 3 раза ниже, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б за 16 секунд. Во сколько раз пропускная способность канала связи с городом Б больше пропускной способности канала связи с городом А?
- В сети Интернет найдите информацию о записи музыкальных произведений в формате MIDI. Почему запись звука в этом формате считают аналогичной векторному методу кодирования графических изображений?

Дополнительные материалы к главе смотрите в авторской мастерской.



Глава 4

ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ И АЛГЕБРЫ ЛОГИКИ

§ 17

Некоторые сведения из теории множеств

17.1. Понятие множества

С понятием множества вы познакомились на уроках математики ещё в начальной школе, а затем работали с ним при изучении математики и информатики в основной школе.

Множество — это совокупность объектов произвольной природы, которая рассматривается как единое целое.

Примерами множеств могут служить: множество всех учеников вашего класса, множество всех жителей Санкт-Петербурга, множество всех натуральных чисел, множество всех решений некоторого уравнения и т. п.

Множества принято обозначать прописными буквами латинского алфавита (A , B , C , ...). Объекты, входящие в состав множества, называются его элементами.

Множество можно задать следующими способами:

- 1) перечислением всех его элементов;
- 2) характеристическим свойством его элементов.

В первом случае внутри фигурных скобок перечисляются все объекты, составляющие множество. Каждый объект, входящий в множество, указывается в фигурных скобках лишь один раз.

Например, запись $M = \{1, 3, 5, 7, 9\}$ означает, что множество M состоит из чисел 1, 3, 5, 7 и 9. Точно такой же смысл будет иметь запись $M = \{3, 1, 5, 9, 7\}$. Иначе говоря, порядок расположения элементов в фигурных скобках значения не имеет. Важно точно указать, какие именно объекты являются элементами множества.

Например:

- число 5 является элементом множества M : $5 \in M$ ¹⁾;
- число 4 не является элементом множества M : $4 \notin M$.

Это же множество можно задать с помощью характеристического свойства образующих его элементов — такого свойства, которым обладает каждый элемент, принадлежащий множеству, и не обладает ни один элемент, который ему не принадлежит. В нашем примере можно говорить о множестве натуральных однозначных нечётных чисел.

В рассматриваемом множестве M содержится 5 элементов. Это обозначают так: $|M| = 5$. Можно составить множество, содержащее любое число элементов. Например, множество всех корней уравнения $x^2 - 4x - 5 = 0$ конечно (два элемента), а множество всех точек прямой бесконечно. Множество, не содержащее ни одного элемента, называется пустым и обозначается символом \emptyset .

Первый способ задания множеств применим только для конечных множеств, да и то при условии, что число элементов множества невелико. Вторым способом можно задавать как конечные, так и бесконечные множества.

Из некоторых элементов множества M можно составить новое множество, например P : $P = \{1, 3, 5\}$.

Если каждый элемент множества P принадлежит множеству M , то говорят, что P есть подмножество M , и записывают: $P \subset M$.

Само множество M является своим подмножеством, т. к. каждый элемент M принадлежит множеству M . Пустое множество также является подмножеством M .

Работая с объектами какой-то определённой природы, всегда можно выделить «самое большое» или универсальное множество, содержащее все возможные подмножества. Пусть A — множество чётных чисел, B — множество натуральных чисел, C — множество чисел, кратных пяти. Тогда самым большим множеством, содержащим в себе множества A , B и C , а также другие подобные множества, будет множество целых чисел. Универсальное множество будем обозначать буквой U .

Для наглядного изображения множеств используются круги Эйлера (рис. 4.1). Точки внутри круга считаются элементами множества.

1) Символ \in называется знаком принадлежности.

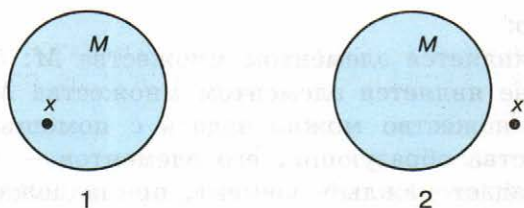


Рис. 4.1. Графическое изображение множеств: 1) $x \in M$, 2) $x \notin M$

17.2. Операции над множествами

Над множествами, как и над числами, производят некоторые операции.

Пересечением двух множеств X и Y называется множество их общих элементов.

Пересечение множеств обозначают с помощью знака \cap : $X \cap Y$. На рисунке 4.2 закрашено множество $X \cap Y$.

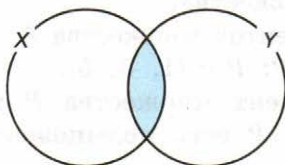


Рис. 4.2. Графическое изображение множества $X \cap Y$

Пусть множества X и Y состоят из букв:

$$X = \{\text{ш; к, о, л, а}\};$$

$$Y = \{\text{у, р, о, к}\}.$$

Эти множества имеют общие элементы: к, о.

$$X \cap Y = \{\text{к, о}\}.$$

Множества M и X не имеют общих элементов, их пересечение — пустое множество:

$$M \cap X = \emptyset.$$

Пересечение множеств M и P есть множество P , а пересечение множеств M и M есть множество M :

$$M \cap P = P;$$

$$M \cap M = M.$$

Объединением двух множеств X и Y называется множество, состоящее из всех элементов этих множеств и не содержащее никаких других элементов.

Объединение множеств обозначают с помощью знака \cup : $X \cup Y$.

На рисунке 4.3 закрашено множество $X \cup Y$.

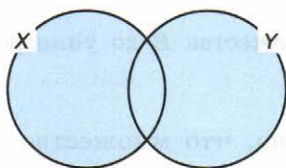


Рис. 4.3. Графическое изображение множества $X \cup Y$

Для наших примеров:

$$\begin{aligned} X \cup Y &= \{\text{ш, к, о, л, а, у, р}\}; \\ M \cup X &= \{1, 3, 5, 7, 9, \text{ш, к, о, л, а}\}; \\ M \cup P &= M; \quad M \cup M = M. \end{aligned}$$

Подумайте, возможно ли равенство: $A \cup B = A \cap B$.

Пересечение и объединение выполняются для любой пары множеств. Третья операция — дополнение — имеет смысл не для всех множеств, а только тогда, когда второе множество является подмножеством первого.

Пусть множество P является подмножеством множества M . **Дополнением** P до M называется множество, состоящее из тех элементов M , которые не вошли в P .

Дополнение P до M обозначают \bar{P} : $\bar{P} = \{7, 9\}$.

Дополнение M до M есть пустое множество, дополнение пустого множества до M есть M : $\bar{M} = \emptyset$; $\bar{\emptyset} = M$.

Особый интерес представляет дополнение некоторого множества B до универсального множества U . Например, если B — это множество точек, принадлежащих некоторому отрезку, то его дополнением \bar{B} до универсального множества U , которым в данном случае является множество всех точек числовой прямой, является множество точек, не принадлежащих данному отрезку.

В общем случае можем записать: $B \cup \bar{B} = U$ (рис. 4.4)

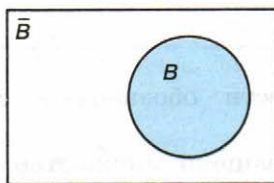


Рис. 4.4. Дополнение множества B до универсального множества

На рисунке 4.5 видно, что множество $A \cup B$ будет совпадать с универсальным, если A будет совпадать с множеством \bar{B} или содержать его в качестве подмножества. В первом случае, т. е. при $A = \bar{B}$, мы имеем дело с минимальным множеством A, таким что $A \cup B = U$.

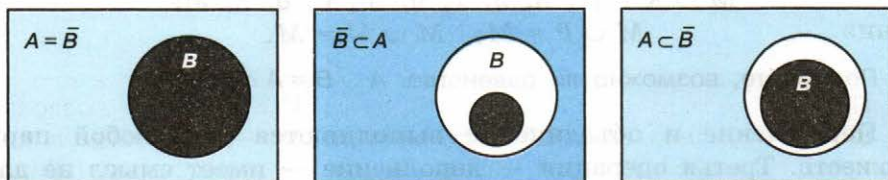


Рис. 4.5. Выбор такого множества A, что $A \cup B = U$



Каким должно быть множество A для того, чтобы множество $\bar{A} \cup B$ совпадало с универсальным множеством?

Для ответа на этот вопрос воспользуйтесь рисунком 4.6.

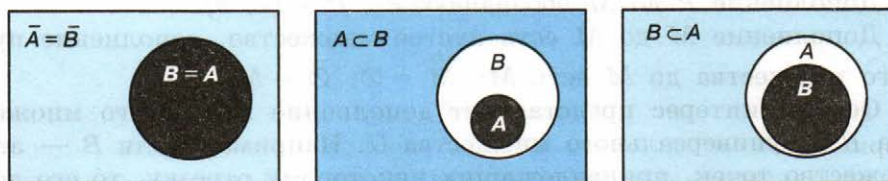


Рис. 4.6. Выбор такого множества A, что $\bar{A} \cup B = U$

17.3. Мощность множества

Мощностью конечного множества называется число его элементов. Мощность множества X обозначается $|X|$.

В рассмотренных выше примерах $|X| = 5$, $|M| = 5$.

Число элементов объединения двух непересекающихся множеств равно сумме чисел элементов этих множеств. Так, в объединении множеств M и X содержится 10 элементов: $|M \cup X| = 10$.

Если же множества пересекаются, то число элементов объединения находится сложнее. Так, X состоит из 5 элементов, множество Y — из 4, а их объединение — из 7. Сложение чисел 5 и 4 даёт нам число 9. Но в эту сумму дважды вошло число элементов пересечения. Чтобы получить правильный результат, надо к числу элементов X прибавить число элементов Y и из суммы вычесть число элементов пересечения. Полученная формула подходит для любых двух множеств: $|X \cup Y| = |X| + |Y| - |X \cap Y|$. Это частный случай так называемого принципа включений-исключений.

Принципом включений-исключений называется формула, позволяющая вычислить мощность объединения (пересечения) множеств, если известны их мощности и мощности всех их пересечений (объединений).

Для случая объединения трёх множеств формула имеет вид:

$$|X \cup Y \cup Z| = |X| + |Y| + |Z| - |X \cap Y| - |X \cap Z| - |Y \cap Z| + |X \cap Y \cap Z|.$$

Аналогичные формулы справедливы и для пересечения множеств:

$$|X \cap Y| = |X| + |Y| - |X \cup Y|;$$

$$|X \cap Y \cap Z| = |X| + |Y| + |Z| - |X \cup Y| - |X \cup Z| - |Y \cup Z| + |X \cup Y \cup Z|.$$

Пример. В зимний оздоровительный лагерь отправляется 100 старшеклассников. Почти все они увлекаются сноубордом, коньками или лыжами. При этом многие из них занимаются не одним, а двумя и даже тремя видами спорта. Организаторы выяснили, что всего кататься на сноуборде умеют 30 ребят, на лыжах — 28, на коньках — 42. Всего умением кататься на лыжах и сноуборде

из них могут похвастаться 8 ребят, на лыжах и коньках — 10, на сноуборде и коньках — 5, но только трое из них владеют всеми тремя видами спорта. Сколько ребят не умеет кататься ни на сноуборде, ни на лыжах, ни на коньках?

Обозначим через S , L и K множества сноубордистов, лыжников и любителей коньков соответственно. Тогда $|S| = 30$, $|L| = 28$ и $|K| = 42$. При этом $|S \cap L| = 8$, $|K \cap L| = 10$, $|S \cap K| = 5$, $|S \cap L \cap K| = 3$.

Объединение множеств S , L и K — это множество ребят, увлекающихся хотя бы каким-то видом спорта.

По формуле включений-исключений находим:

$$|S \cup L \cup K| = 30 + 28 + 42 - 8 - 10 - 5 + 3 = 80.$$

Таким образом, из 100 старшеклассников 20 не умеют кататься ни на сноуборде, ни на лыжах, ни на коньках.

САМОЕ ГЛАВНОЕ

Множество — это совокупность объектов произвольной природы, которая рассматривается как единое целое.

Пересечением двух множеств X и Y называется множество их общих элементов.

Объединением двух множеств X и Y называется множество, состоящее из всех элементов этих множеств и не содержащее никаких других элементов.

Пусть множество P является подмножеством множества M . Дополнением P до M называется множество, состоящее из тех элементов M , которые не вошли в P .

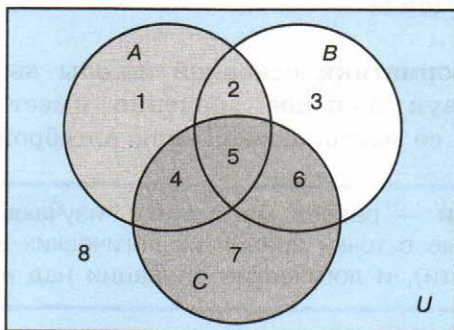
Мощностью конечного множества называется число его элементов.

Формула включений-исключений позволяет вычислить мощность объединения (пересечения) множеств, если известны их мощности и мощности всех их пересечений (объединений).

Вопросы и задания

1. Если множество X — это множество натуральных чисел, делящихся нацело на 2, а Y — множество натуральных чисел, делящихся нацело на 3, то что будет:
 - 1) пересечением этих множеств;
 - 2) объединением этих множеств?

2. Пусть множество X — это множество натуральных чисел, делящихся нацело на 18, а Y — множество натуральных чисел, делящихся нацело на 14. Укажите наименьшее число, входящее:
- 1) в пересечение этих множеств;
 - 2) в объединение этих множеств?
3. Пусть A , B и C — некоторые множества, обозначенные кругами, U — универсальное множество.



С помощью операций объединения, пересечения и дополнения до универсального множества выразите через A , B и C следующие множества:

- | | |
|---|-------------------------------|
| 1) $1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6$; | 4) $2 \cup 4 \cup 5 \cup 6$; |
| 2) $2 \cup 5$; | 5) $1 \cup 2 \cup 3$; |
| 3) 5 ; | 6) 8 . |
4. В первую смену в лагере «Дубки» отдыхали: 30 отличников, 28 победителей олимпиад и 42 спортсмена. При этом 10 человек были и отличниками, и победителями олимпиад, 5 — отличниками и спортсменами, 8 — спортсменами и победителями олимпиад, 3 — и отличниками, и спортсменами, и победителями олимпиад. Сколько ребят отдыхало в лагере?
5. Старшеклассники заполняли анкету с вопросами об экзаменах по выбору. Оказалось, что выбрали они информатику, физику и обществознание. В классе 38 учеников. Обществознание выбрал 21 ученик, причём трое из них выбрали ещё и информатику, а шестеро — ещё и физику. Один ученик выбрал все три предмета. Всего информатику выбрали 13 учеников, пятеро из которых указали в анкете два предмета. Надо определить, сколько же учеников выбрали физику.

*6. Из 100 человек 85 знают английский язык, 80 — испанский, 75 — немецкий. Сколько человек знают все три языка?

§ 18

Алгебра логики

Из курса информатики основной школы вы знаете, что для компьютерных наук большое значение имеет математическая логика, а точнее, её часть, называемая алгеброй логики.

Алгебра логики — раздел математики, изучающий высказывания, рассматриваемые с точки зрения их логических значений (истинности или ложности), и логические операции над ними.



Джордж Буль (1815–1864) — английский математик, основоположник алгебры логики. Дж. Буль изучал логику мышления математическими методами и разработал алгебраические методы решения традиционных логических задач. В 1854 году он опубликовал работу, в которой изложил суть алгебры логики, основанной на трёх операциях: and, or, not. Долгое время алгебра логики была известна достаточно узкому классу специалистов. В 1938 году Клод Шеннон применил алгебру логики для описания процесса функционирования релейно-контактных и электронно-ламповых схем.

18.1. Логические высказывания и переменные

Высказывание — это предложение, в отношении которого можно сказать, истинно оно или ложно.

Например, высказывание «Джордж Буль — основоположник алгебры логики» истинно, а высказывание « $2 + 2 = 5$ » ложно.

Что вы можете сказать об истинности или ложности предложения «Данное высказывание — ложь»?

Из имеющихся высказываний можно строить новые высказывания. Для этого используются логические связки — слова и словосочетания «не», «и», «или», «если ..., то», «тогда и только тогда» и др.

Высказывания, образованные из других высказываний, называются составными (сложными). Высказывание, никакая часть которого не является высказыванием, называется элементарным (простым).

Например, из двух простых высказываний «Алгебра логики является основой строения логических схем компьютеров» и «Алгебра логики служит математической основой решения сложных логических задач» можно получить составное высказывание «Алгебра логики является основой строения логических схем компьютеров и служит математической основой решения сложных логических задач».

Обоснование истинности или ложности элементарных высказываний не является задачей алгебры логики. Эти вопросы решаются теми науками, к сфере которых относятся элементарные высказывания. Такое сужение интересов позволяет обозначать высказывания символическими именами (например, A , B , C). Так, если обозначить элементарное высказывание «Джордж Буль — основоположник алгебры логики» именем A , а элементарное высказывание « $2 + 2 = 5$ » именем B , то составное высказывание «Джордж Буль — основоположник алгебры логики, и $2 + 2 = 5$ » можно записать как « A и B ». Здесь A , B — логические переменные, «и» — логическая связка.

Логическая переменная — это переменная, которая обозначает любое высказывание и может принимать логические значения «истина» или «ложь».

Для логических значений «истина» и «ложь» могут использоваться следующие обозначения:

Истина	Ложь
И	Л
true	false
да	нет
1	0

Истинность или ложность составных высказываний зависит от истинности или ложности образующих их высказываний и определённой трактовки связок (логических операций над высказываниями).

18.2. Логические операции

Логическая операция полностью может быть описана таблицей истинности, указывающей, какие значения принимает составное высказывание при всех возможных значениях образующих его элементарных высказываний.

Из курса информатики основной школы вам известны логические операции отрицание, конъюнкция и дизъюнкция. Их таблицы истинности представлены ниже.

Конъюнкция

A	B	A и B
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкция

A	A	A или B
0	0	0
0	1	1
1	0	1
1	1	1

Отрицание

A	\bar{A}
0	1
1	0

Логическая операция, ставящая в соответствие двум высказываниям новое, являющееся истинным тогда и только тогда, когда оба исходных высказывания истинны, называется **конъюнкцией** или **логическим умножением**.

Логическая операция, ставящая в соответствие двум высказываниям новое, являющееся ложным тогда и только тогда, когда оба исходных высказывания ложны, называется **дизъюнкцией** или **логическим сложением**.

Логическая операция, которая каждому высказыванию ставит в соответствие новое высказывание, значение которого противоположно исходному, называется **отрицанием** или **инверсией**.

При построении отрицания простого высказывания:

- используется оборот «неверно, что» или к сказуемому добавляется частица «не»;
- в высказывании, содержащем слово «все», это слово заменяется на «некоторые» и наоборот.

Рассмотрим несколько новых логических операций.

Логическая операция, ставящая в соответствие двум высказываниям новое, являющееся ложным тогда и только тогда, когда первое высказывание (посылка) истинно, а второе (следствие) — ложно, называется **импликацией** или **логическим следованием**.

Операция импликации обозначается символом \rightarrow и задаётся следующей таблицей истинности:

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

В разговорной речи импликации соответствуют предложения, содержащие связку «если ..., то». Эту связку мы используем тогда, когда хотим показать наличие причинно-следственной связи, иначе говоря, зависимость одного события от другого. Например, пусть некоторый человек сказал: «Если завтра будет хорошая погода, то я пойду гулять». Ясно, что человек окажется лжецом лишь в том случае, если погода действительно будет хорошей, а гулять он не пойдёт. Если же погода будет плохой, то, независимо от того, пойдёт он гулять или нет, во лжи его нельзя обвинить: обещание пойти гулять он давал лишь при условии, что погода будет хорошей.

Результат операции импликации, как и других логических операций, определяется истинностью или ложностью логических переменных, а не наличием причинно-следственных связей между высказываниями. Например, абсурдное с житейской точки зрения высказывание «Если $2 > 3$, то существуют ведьмы» является истинным с точки зрения алгебры логики.

Логическая операция, ставящая в соответствие двум высказываниям новое, являющееся истинным тогда и только тогда, когда только одно из двух высказываний истинно, называется **строгой (исключающей) дизъюнкцией**.

Строгая дизъюнкция обозначается символом \oplus и задаётся следующей таблицей истинности:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

В русском языке строгой (разделительной) дизъюнкции соответствует связка «либо». В отличие от обычной дизъюнкции (связка «или») в высказывании, содержащем строгую дизъюнкцию, мы утверждаем, что произойдёт только одно событие.

Например, высказывая утверждение «На сегодняшнем матче Петя сидит на трибуне А либо на трибуне В», мы считаем, что Петя сидит либо только на трибуне А, либо только на трибуне В, и что сидеть одновременно на двух трибунах Петя не может.

Логическая операция, ставящая в соответствие двум высказываниям новое, являющееся истинным, когда оба исходных высказывания истинны или оба исходных высказывания ложны, называется **эквиваленцией** или **равнозначностью**.

В логике эквиваленция обозначается символом \leftrightarrow и задаётся следующей таблицей истинности:

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

В разговорной речи для выражения взаимной обусловленности используется связка «тогда и только тогда, когда», а в математике — «необходимо и достаточно».

Рассмотрим высказывание «Денис пойдёт в бассейн тогда и только тогда, когда он выучит уроки».

Это высказывание истинно (договорённость соблюдается), если истинны оба элементарных высказывания («Денис пойдёт в бассейн», «Денис выучит уроки»). Высказывание истинно (договорённость не нарушается) и в том случае, если оба элементарных высказывания ложны («Денис не пойдёт в бассейн», «Денис не выучит уроки»). Если же одно из двух высказываний ложно («Денис пойдёт в бассейн, хотя и не выучит уроки», «Денис выучит уроки, но не пойдёт в бассейн»), то договорённость нарушается, и составное высказывание становится ложным.

А сейчас посмотрите внимательно на таблицы истинности строгой дизъюнкции и эквиваленции: если на некотором наборе логических переменных результатом строгой дизъюнкции является истина, то на этом же наборе результатом эквиваленции всегда будет ложь, и наоборот. Можно сделать выводы:

- операция эквиваленции есть отрицание операции строгой дизъюнкции ($A \leftrightarrow B = \overline{A \oplus B}$);
- операция строгой дизъюнкции есть отрицание операции эквиваленции ($A \oplus B = \overline{A \leftrightarrow B}$).

На сегодняшний день в алгебре логики не существует унифицированной символики для обозначения логических операций. В таблице 4.1 представлены логические операции и их наиболее распространённые обозначения, используемые как в алгебре логики, так и в некоторых языках программирования. Здесь же введены речевые обороты, соответствующие логическим операциям.

Таблица 4.1

Логические операции и их обозначения

Операция	Обозначение	Речевой оборот
Отрицание (инверсия, логическое НЕ)	$\neg A$, \bar{A} , НЕ A , not A	«Не», «неверно, что»
Конъюнкция (логическое умножение, логическое И)	$A \wedge B$, $A \& B$, $A \cdot B$, AB , A И B , A and B	«И», «как ..., так и», «вместе с», «но», «хотя», «а»
Дизъюнкция (логическое сложение, логическое ИЛИ)	$A \vee B$, $A + B$, $A B$, A ИЛИ B , A or B	«Или», «или ..., или ...», или оба вместе»

Операция	Обозначение	Речевой оборот
Строгая дизъюнкция (исключающая дизъюнкция, исключающее ИЛИ)	$A \oplus B$, $A \text{ хог } B$	«Либо ..., либо», «только ... или только»
Импликация (логическое следование)	$A \rightarrow B$, $A \Rightarrow B$	«Если ..., то», «из ... следует», «влечёт»
Эквиваленция (эквивалентность, равнозначность)	$A \leftrightarrow B$, $A \Leftrightarrow B$, $A \equiv B$	«Эквивалентно», «равносильно», «необходимо и достаточно», «тогда и только тогда, когда»

Операция отрицания выполняется над одним операндом. Такие операции называются одноместными или унарными. Все остальные логические операции, представленные в таблице 4.1, выполняются над двумя операндами и называются двуместными или бинарными.

18.3. Логические выражения

Составное логическое высказывание можно представить в виде логического выражения (формулы), состоящего из логических констант (0, 1), логических переменных, знаков логических операций и скобок.

Для логического выражения справедливо:

- 1) всякая логическая переменная, а также логические константы (0, 1) есть логическое выражение;
- 2) если A — логическое выражение, то и \bar{A} — логическое выражение;
- 3) если A и B — выражения, то, связанные любой бинарной операцией, они также представляют собой логическое выражение.

При преобразовании или вычислении значения логического выражения логические операции выполняются в соответствии с их приоритетом:

- 1) отрицание;
- 2) конъюнкция;
- 3) дизъюнкция, строгая дизъюнкция;
- 4) импликация, эквиваленция.

Операции одного приоритета выполняются в порядке их следования, слева направо. Как и в арифметике, скобки меняют порядок выполнения операций.

Пример 1. Выясним, какие из приведённых слов удовлетворяют логическому условию (первая буква согласная \rightarrow вторая буква согласная) $\&$ (последняя буква гласная \rightarrow предпоследняя буква гласная):

- 1) ОЗОН;
- 2) ИГРА;
- 3) МАФИЯ;
- 4) ТРЕНАЖ.

Вычислим значение логического выражения для каждого из данных слов:

- 1) $(0 \rightarrow 1) \& (0 \rightarrow 1) = 1 \& 1 = 1$;
- 2) $(0 \rightarrow 1) \& (1 \rightarrow 0) = 1 \& 0 = 0$;
- 3) $(1 \rightarrow 0) \& (1 \rightarrow 1) = 0 \& 1 = 0$;
- 4) $(1 \rightarrow 1) \& (0 \rightarrow 1) = 1 \& 1 = 1$.

Итак, заданному условию удовлетворяют первое и четвёртое слова.

Решение логического уравнения — это один или несколько наборов значений логических переменных, при которых логическое уравнение становится истинным выражением.

Пример 2. Решим логическое уравнение

$$(A \rightarrow C) \vee ((\overline{B \vee C}) \& A) \vee D = 0.$$

Дизъюнкция ложна в том и только в том случае, когда ложно каждое из образующих её высказываний. Иными словами, наше уравнение соответствует системе уравнений:

$$\begin{cases} A \rightarrow C = 0; \\ (\overline{B \vee C}) \& A = 0; \\ D = 0. \end{cases}$$

Таким образом, значение переменной D уже найдено.

Импликация равна нулю в единственном случае — когда из истины следует ложь. Иначе говоря, в нашем случае: $A = 1$ и $C = 0$.

Подставим найденные значения переменных в уравнение $(\overline{B \vee C}) \& A = 0$. Получим: $(\overline{B \vee 0}) \& 1 = 0$ или $\overline{B} = 0$, т. е. $B = 1$.

Ответ: $A = 1$, $B = 1$, $C = 0$, $D = 0$.

Логические уравнения могут иметь не одно, а несколько и даже очень много решений. Зачастую требуется, не выписывая все решения уравнения, указать их количество.

Пример 3. Выясним, сколько различных решений имеет логическое уравнение $(A \& B \& \bar{C}) \vee (\bar{B} \& C \& D) = 1$.

Дизъюнкция истинна, если истинно хотя бы одно из образующих её высказываний. Решение данного логического уравнения равносильно совокупности, состоящей из двух уравнений:

$$\begin{cases} A \& B \& \bar{C} = 1; \\ \bar{B} \& C \& D = 1. \end{cases}$$

Первое равенство будет выполняться только при $A = 1$, $B = 1$ и $C = 0$. Поскольку D в этом уравнении не задействовано, оно может принимать любое из двух значений (0 или 1). Таким образом, всего первое уравнение имеет два решения.

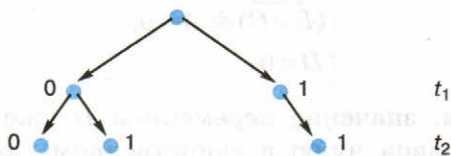
Самостоятельно выясните, сколько решений имеет второе уравнение (из совокупности двух уравнений).

Сколько решений имеет исходное уравнение?

Пример 4. Выясним, сколько решений имеет очень простое с виду логическое уравнение $x_1 \& x_2 \rightarrow x_3 \& x_4 = 1$.

Введём замену переменных. Пусть $t_1 = x_1 \& x_2$, $t_2 = x_3 \& x_4$. Тогда исходное уравнение примет вид: $t_1 \rightarrow t_2 = 1$.

На t_1 никаких ограничений нет, эта переменная может принимать значения 0 и 1. Импликация равна 0 только в случае, когда из истины (1) следует ложь (0). Исключим этот вариант. Построим дерево решений, представив на нём значения переменных t_1 и t_2 , при которых $t_1 \rightarrow t_2 = 1$.



Получаем для t_1 и t_2 три набора значений: 00, 01, 11. Первая двоичная цифра в каждом из этих трёх наборов — результат выражения $x_1 \& x_2$, вторая — $x_3 \& x_4$. Рассмотрим первый набор: существует три набора x_1 и x_2 таких, что $x_1 \& x_2 = 0$, другими

словами, первый 0 мы можем получить тремя способами. Вторым 0 в этом наборе мы также можем получить тремя способами.

Из курсов информатики и математики основной школы вам известно одно из основных правил комбинаторики — правило умножения. Согласно ему, если элемент A можно выбрать n способами, и при любом выборе A элемент B можно выбрать m способами, то пару (A, B) можно выбрать $n \cdot m$ способами.

Согласно правилу умножения, пару 00 можно получить $3 \cdot 3 = 9$ способами.

Что касается пары 01, то первый 0 мы можем получить тремя способами, а для получения 1 существует единственный вариант ($x_3 \& x_4 = 1$ при $x_3 = 1$ и $x_4 = 1$). Следовательно, есть ещё три набора переменных x_1, x_2, x_3, x_4 , являющихся решением исходного уравнения.

Самостоятельно доведите решение этой задачи до конца.



18.4. Предикаты и их множества истинности

Равенства, неравенства и другие предложения, содержащие переменные, высказываниями не являются, но они становятся высказываниями при замене переменной каким-нибудь конкретным значением. Например, предложение $x < 12$ становится истинным высказыванием при $x = 5$ ($5 < 12$ — истина) и ложным при $x = 15$ ($15 < 12$ — ложь). Предложения такого рода называют высказывательными формами или предикатами.

Предикат — это утверждение, содержащее одну или несколько переменных.



Выделим некоторый предикат $P(x)$ и рассмотрим множество всевозможных объектов I , к которым он относится, — область определения предиката. Можно выделить такое подмножество P множества I , что на всех его элементах предикат $P(x)$ будет превращаться в истинное высказывание. Определённое таким образом P называется множеством истинности предиката $P(x)$.

Рассмотрим множество учеников некоторого класса. Известно, что в этом классе два отличника — Иван и Саша. Предикат «Он отличник» будет истинным высказыванием только по отношению к этим двум ученикам и ложным по отношению ко всем остальным.

Предикаты позволяют задать множество, не перечисляя всех его элементов. Например, множество истинности предиката $P(x) = (x < 0)$ — множество отрицательных чисел; множество истинности предиката $P(x, y) = (x^2 + y^2 = 1)$ — множество точек окружности единичного радиуса с центром в начале координат. Следует отметить, что многие задания, выполняемые вами на уроках математики, прямо связаны с предикатами. Например, стандартное задание «Решить квадратное уравнение $x^2 - 3x + 2 = 0$ » фактически означает требование найти множество истинности предиката $P(x) = (x^2 - 3x + 2 = 0)$.

Из имеющихся предикатов с помощью логических операций можно строить новые предикаты.

Пусть A и B соответственно являются множествами истинности предикатов $A(x)$ и $B(x)$. Тогда пересечение множеств A и B будет являться множеством истинности для предиката $A(x) \& B(x)$, а объединение множеств A и B будет множеством истинности для предиката $A(x) \vee B(x)$.

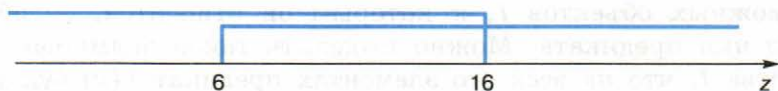
Пример 5. Найдём все целые числа z , превращающие предикат

$$P(z) = (z > 5) \& (z - 2 < 15)$$

в истинное высказывание. Другими словами, требуется найти множество истинности предиката $P(z)$, заданного на множестве целых чисел \mathbb{Z} .

Предикат $P(z)$ состоит из двух предикатов, соединённых операцией конъюнкции: $P(z) = A(z) \& B(z)$. Рассмотрим каждый из них в отдельности.

Множеством истинности предиката $A(z) = (z > 5)$ являются целые числа 6, 7, 8 и т. д. Множеством истинности предиката $B(z) = (z - 2 < 15)$ являются все целые числа, меньшие 17.



Множество истинности исходного предиката — пересечение (общие элементы) множеств истинности образующих его предикатов:

$$P = A \cap B = \{6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}.$$

Его мощность $|P| = 11$.

Пример 6. Рассмотрим предикат $(50 < x^2) \rightarrow (50 > (x + 1)^2)$, определённый на множестве целых чисел. Найдём множество истинности этого предиката.

Зачастую задания такого рода формулируют несколько иначе. Например, так: «Найдите все целые числа x , для которых истинно высказывание $(50 < x^2) \rightarrow (50 > (x + 1)^2)$ ».

Проанализируем отдельно каждый из элементарных предикатов $(50 < x^2)$ и $(50 > (x + 1)^2)$, решив соответствующие неравенства:

$50 < x^2$ истинно для всех целых $x \in]-\infty; -8] \cup [8; +\infty[$;

$50 > (x + 1)^2$ истинно для всех целых $x \in [-8; 6]$.

Определим значение исходного предиката на каждом из полученных подмножеств, причём отдельно рассмотрим значение $x = -8$ (оно попадает в два подмножества) и значение $x = 7$ (оно не попадает ни в одно подмножество):

$x \in Z$	$50 < x^2$	$50 > (x + 1)^2$	$(50 < x^2) \rightarrow (50 > (x + 1)^2)$
$]-\infty; -9]$	1	0	0
-8	1	1	1
$[-7; 6]$	0	1	1
7	0	0	1
$[8; +\infty[$	1	0	0

Итак, множеством истинности исходного предиката являются целые числа, принадлежащие отрезку $[-8; 7]$. Наименьшим элементом этого множества является число -8 , наибольшим — число 7 ; мощность множества равна 16 .

САМОЕ ГЛАВНОЕ

Высказывание — это предложение, в отношении которого можно сказать, истинно оно или ложно. Высказывания, образованные из других высказываний, называются составными (сложными). Высказывание, никакая часть которого не является высказыванием, называется элементарным (простым). Истинность или ложность составных высказываний зависит

от истинности или ложности образующих их высказываний и определённой трактовки связок (логических операций над высказываниями).

Логическая операция полностью может быть описана таблицей истинности, указывающей, какие значения принимает составное высказывание при всех возможных значениях образующих его элементарных высказываний.

Логические переменные		Логические операции					
		Отрицание	Конъюнкция	Дизъюнкция	Импликация	Строгая дизъюнкция	Эквиваленция
A	B	\bar{A}	$A \& B$	$A \vee B$	$A \rightarrow B$	$A \oplus B$	$A \leftrightarrow B$
0	0	1	0	0	1	0	1
0	1	1	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	1	1	1	0	1

Составное логическое высказывание можно представить в виде логического выражения (формулы), состоящего из логических констант (0, 1), логических переменных, знаков логических операций и скобок.

Логические операции имеют следующий приоритет:

- 1) отрицание;
- 2) конъюнкция;
- 3) дизъюнкция, строгая дизъюнкция;
- 4) импликация, эквиваленция.

Операции одного приоритета выполняются в порядке их следования, слева направо. Скобки меняют порядок выполнения операций.

Предикат — это утверждение, содержащее одну или несколько переменных. Из имеющихся предикатов с помощью логических операций можно строить новые предикаты.

Вопросы и задания



1. Из данных предложений выберите те, которые являются высказываниями. Обоснуйте свой выбор.

- 1) Как пройти в библиотеку?
- 2) Коля спросил: «Который час?»
- 3) Картины Пикассо слишком абстрактны.
- 4) Компьютеры могут быть построены только на основе двоичной системы счисления.

2. Из каждых трёх выберите два высказывания, являющихся отрицаниями друг друга:

- 1) « $1999 < 2000$ », « $1999 > 2000$ », « $1999 \leq 2000$ »;
- 2) «Петя решил все задания контрольной работы», «Петя не решил все задания контрольной работы», «Петя решил не все задания контрольной работы»;
- 3) «Луна — спутник Земли», «Неверно, что Луна — спутник Земли», «Неверно, что Луна не является спутником Земли»;
- 4) «Прямая a не параллельна прямой c », «Прямая a перпендикулярна прямой c », «Прямые a и c не пересекаются (считаем, что прямые a и c лежат в одной плоскости);
- 5) «Мишень поражена первым выстрелом», «Мишень поражена не первым выстрелом», «Неверно, что мишень поражена не первым выстрелом».

3. Рассмотрите следующие элементарные высказывания: A = «Река Днепр впадает в Чёрное море», B = «45 — простое число», C = «Вена — столица Австрии», D = «0 — натуральное число».

Определите, какие из них истинные, а какие ложные.

Составьте сложные высказывания, применяя каждый раз только одну из пяти логических операций (\neg , $\&$, \vee , \rightarrow , \leftrightarrow) к высказываниям A , B , C и D . Сколько новых высказываний можно получить с помощью отрицания (инверсии)? Конъюнкции? Дизъюнкции? Импликации? Эквиваленции?

Сколько всего новых высказываний можно получить? Сколько среди них будет истинных?

4. Представьте каждую поговорку в виде сложного логического высказывания, построенного на основе простых высказываний. Ответ обоснуйте при помощи таблиц истинности.

- 1) На вкус и цвет товарищей нет.
- 2) Если долго мучиться, что-нибудь получится.

- 3) Не зная броду, не суйся в воду.
 4) Тяжело в ученье, легко в бою.
 5) То не беда, что во ржи лебеда, то беда, что ни ржи, ни лебеды.
 6) Где тонко, там и рвётся.
 7) Или грудь в крестах, или голова в кустах.
 8) За двумя зайцами погонишься — ни одного не поймаешь.
 9) И волки сыты, и овцы целы.
5. Подберите вместо A , B , C , D такие высказывания, чтобы полученные сложные высказывания имели смысл:
- 1) если (A или B и C), то D ;
 - 2) если (не A и не B), то (C или D);
 - 3) (A или B) тогда и только тогда, когда (C и не D).
6. Вычислите:
- 1) $1 \vee X \& 0$;
 - 2) $X \& X \& 1$;
 - 3) $0 \& X \vee 0$;
 - 4) $0 \vee X \& X$.
7. Сколько из приведённых чисел Z удовлетворяют логическому условию: $((Z \text{ кратно } 4) \vee (Z \text{ кратно } 5)) \rightarrow (Z \text{ кратно } 6)$?
- 1) 4; 2) 6; 3) 7; 4) 12.
8. Найдите все целые числа Z , для которых истинно высказывание:
- 1) $\overline{(Z > 5)} \& (Z^2 < 100)$;
 - 2) $(Z > 5) \rightarrow (Z > 10)$.
9. Какие из высказываний A , B , C должны быть истинны и какие ложны, чтобы были ложны следующие высказывания?
- 1) $\overline{((A \vee B) \& B)} \rightarrow C$;
 - 2) $A \& \overline{B} \leftrightarrow 1$.
10. Даны три числа в различных системах счисления:
 $A = 23_{10}$, $B = 23_8$, $C = 1A_{16}$.
 Переведите A , B и C в двоичную систему счисления и выполните поразрядно логические операции $(A \vee B) \& C$. Ответ дайте в десятичной системе счисления.
11. Логическое отрицание восьмиразрядного двоичного числа, записанное в десятичной системе счисления, равно 217. Определите исходное число в десятичной системе счисления.
12. Определите логическое произведение и логическую сумму всех двоичных чисел в диапазоне от 16_{10} до 22_{10} , включая

- границы. Ответ запишите в восьмеричной системе счисления.
13. Сколько различных решений имеет логическое уравнение?
- 1) $(A \vee B \vee C) \& (\bar{B} \& \bar{C} \& D) = 1$;
 - 2) $(A \vee B \vee C) \vee (\bar{B} \& C \& D) = 0$;
 - 3) $(A \rightarrow C) \vee (B \& A) \vee (D \rightarrow B \& C) = 0$;
 - 4) $(A \& B \& C) \rightarrow (\bar{C} \& D) = 1$.
14. Сколько решений имеет логическое уравнение $x_1 \& x_2 \vee x_3 \& x_4 = 1$?
15. Изобразите в декартовой прямоугольной системе координат множества истинности для следующих предикатов:
- 1) $P(x, y) = (y \geq x) \& (y + x \geq 0) \& (y \leq 1)$;
 - 2) $P(x, y) = (|x| \leq 1) \& (|y| \leq 1)$;
 - 3) $P(x, y) = (x^2 + y^2 \leq 4) \& (x^2 + y^2 \geq 1)$.
16. Предикат $((8x - 6) < 75) \rightarrow (x(x - 1) > 65)$ определен на множестве целых чисел. Найдите его множество истинности. Укажите наибольшее целое число x , при котором предикат превращается в ложное высказывание.

§ 19

Таблицы истинности

19.1. Построение таблиц истинности

Таблицу значений, которые принимает логическое выражение при всех сочетаниях значений (наборах) входящих в него переменных, называют таблицей истинности логического выражения.

Для того чтобы построить таблицу истинности логического выражения, достаточно:

- 1) определить число строк таблицы $m = 2^n$, где n — число переменных в логическом выражении;
- 2) определить число столбцов таблицы как сумму чисел логических переменных и логических операций в логическом выражении;

- 3) установить последовательность выполнения логических операций с учётом скобок и приоритетов операций;
- 4) заполнить строку с заголовками столбцов таблицы истинности, занеся в неё имена логических переменных и номера выполняемых логических операций;
- 5) выписать наборы входных переменных с учётом того, что они представляют собой ряд целых n -разрядных двоичных чисел от 0 до $2^n - 1$;
- 6) провести заполнение таблицы истинности по столбцам, выполняя логические операции.



Пример 1. Построим таблицу истинности для логического выражения

$$A \& B \vee \bar{A} \& \bar{B}.$$

В этом выражении две логические переменные и пять логических операций. Всего в таблице истинности будет пять строк (2^2 плюс строка заголовков) и 7 столбцов.

Начнём заполнять таблицу истинности с учётом следующего порядка выполнения логических операций: сначала выполняются операции отрицания (в порядке следования), затем операции конъюнкции (в порядке следования), последней выполняется дизъюнкция.

$$\begin{array}{cccccc}
 \boxed{3} & \boxed{5} & \boxed{1} & \boxed{4} & \boxed{2} & \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 A \& B \vee \bar{A} \& \bar{B}
 \end{array}$$

A	B	1	2	3	4	5
0	0	1	1	0	1	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	0	1



Обратите внимание на последний столбец, содержащий конечный результат. Какой из рассмотренных логических операций он соответствует?

Логические выражения, зависящие от одних и тех же логических переменных, называются равносильными или эквивалентными, если для всех наборов входящих в них переменных значения выражений в таблицах истинности совпадают.

Таблица истинности, построенная в предыдущем примере, доказывает равносильность выражений $A \& B \vee \bar{A} \& \bar{B}$ и $A \leftrightarrow B$.

Можно записать: $A \& B \vee \bar{A} \& \bar{B} = A \leftrightarrow B$.

С помощью таблиц истинности докажите равносильность выражений $A \rightarrow B$ и $\bar{A} \vee B$.

Функцию от n переменных, аргументы которой и сама функция принимают только два значения — 0 и 1, называют **логической функцией**. Таблица истинности может рассматриваться как способ задания логической функции.

19.2. Анализ таблиц истинности

Рассмотрим несколько примеров.

Пример 2. Известен фрагмент таблицы истинности для логического выражения F , содержащего логические переменные A , B и C .

A	B	C	F
1	0	1	0
1	1	0	1
1	1	1	1

Сколько из приведённых ниже логических выражений соответствуют этому фрагменту?

- $(A \vee C) \& B$;
- $(A \vee B) \& (C \rightarrow A)$;
- $(A \& B \vee C) \& (B \rightarrow A \& C)$;
- $(A \rightarrow B) \vee (C \vee A \rightarrow B)$.

Ответить на поставленный вопрос можно, вычислив значение каждого логического выражения на каждом заданном наборе переменных и сравнив его с имеющимся значением F .

- 1) Логическое выражение $(A \vee C) \& B$ соответствует данному фрагменту таблицы истинности:

A	B	C	(A \vee C) & B	F
1	0	1	(1 \vee 1) & 0 = 1 & 0 = 0	0
1	1	0	(1 \vee 0) & 1 = 1 & 1 = 1	1
1	1	1	(1 \vee 1) & 1 = 1 & 1 = 1	1

- 2) Логическое выражение $(A \vee B) \& (C \rightarrow A)$ не соответствует данному фрагменту таблицы истинности, т. к. уже на первом наборе значение рассматриваемого логического выражения не совпадает со значением F . Проведение дальнейших вычислений не имеет смысла.

A	B	C	(A \vee B) & (C \rightarrow A)	F
1	0	1	(1 \vee 0) & (1 \rightarrow 1) = 1 & 1 = 1	0
1	1	0		1
1	1	1		1

- 3) Логическое выражение $(A \& B \vee C) \& (B \rightarrow A \& C)$ не соответствует данному фрагменту таблицы истинности:

A	B	C	(A & B \vee C) & (B \rightarrow A & C)	F
1	0	1	(1 & 0 \vee 1) & (0 \rightarrow 1 & 1) = 1 & 1 = 1	0
1	1	0		1
1	1	1		1

- 4) Логическое выражение $(A \rightarrow B) \vee (C \vee A \rightarrow B)$ соответствует данному фрагменту таблицы истинности:

A	B	C	(A \rightarrow B) \vee (C \vee A \rightarrow B)	F
1	0	1	(1 \rightarrow 0) \vee (1 \vee 1 \rightarrow 0) = 0	0
1	1	0	(1 \rightarrow 1) \vee (0 \vee 1 \rightarrow 1) = 1	1
1	1	1	(1 \rightarrow 1) \vee (1 \vee 1 \rightarrow 1) = 1	1

Итак, имеется два логических выражения, соответствующих заданному фрагменту таблицы истинности.

Можно ли утверждать, что в результате решения задачи мы нашли логическое выражение F ?

Пример 3. Логическая функция F задаётся выражением:

$$(x \vee \bar{y} \vee \bar{z}) \& (\bar{x} \vee y).$$

Ниже приведён фрагмент таблицы истинности, содержащий все наборы переменных, на которых F истинна.

?	?	?	F
0	0	0	1
0	1	0	1
0	1	1	1
1	0	0	1
1	1	1	1

Определим, какому столбцу таблицы истинности функции F соответствует каждая из переменных x , y , z .

В исходном логическом выражении задействовано три логические переменные. Полная таблица истинности для этого выражения должна состоять из 8 (2^3) строк.

Наборам переменных, на которых логическое выражение истинно, соответствуют десятичные числа 0, 2, 3, 4 и 7.

Следовательно, наборам переменных, на которых логическое выражение ложно, должны соответствовать десятичные числа 1, 5 и 6 (их двоичные коды 001, 101 и 110). Построим по этим данным вторую часть таблицы истинности:

?	?	?	F
0	0	1	0
1	0	1	0
1	1	0	0

Теперь выясним, при каких значениях x , y , z логическое выражение ложно: $(x \vee \bar{y} \vee \bar{z}) \& (\bar{x} \vee y) = 0$. Логическое произ-



ведение ложно, если хотя бы один из операндов равен нулю. Таким образом, мы имеем две дизъюнкции, каждая из которых должна быть ложной. Это возможно только в случае равенства нулю каждого из операндов, входящих в дизъюнкцию. Подберём подходящие значения x , y и z , заполняя следующую таблицу:

	x	y	z	F
$x \vee \bar{y} \vee \bar{z}$				0
$\bar{x} \vee y$				0

Первая дизъюнкция равна нулю на наборе 011. Для равенства нулю второй дизъюнкции требуется, чтобы $x = 1$, $y = 0$, а z может быть и 0, и 1.

	x	y	z	F
$x \vee \bar{y} \vee \bar{z}$	0	1	1	0
$\bar{x} \vee y$	1	0	0	0
	1	0	1	0

Сравним эту таблицу с восстановленным нами фрагментом исходной таблицы истинности, предварительно подсчитав, сколько раз каждая переменная принимает единичное значение.

Восстановленный фрагмент

?	?	?	F
0	0	1	0
1	0	1	0
1	1	0	0

2 1 2

Таблица со значениями x , y и z

	x	y	z	F
$x \vee \bar{y} \vee \bar{z}$	0	1	1	0
$\bar{x} \vee y$	1	0	0	0
	1	0	1	0

2 1 2

Переменная y принимает единичное значение только один раз. Следовательно, ей соответствует второй столбец исходной таблицы. Из таблицы со значениями x , y и z следует, что при $y = 1$: $x = 0$, а $z = 1$. Следовательно, переменной z соответствует первый столбец, а переменной x — третий столбец исходной таблицы.

Убедиться в правильности полученного ответа можно, полностью заполнив следующую таблицу:

z	y	x	$A = x \vee \bar{y} \vee \bar{z}$	$B = \bar{x} \vee y$	$A \& B$	F
0	0	0	1	1	1	1
0	0	1	1	0	0	0
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	0	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1

САМОЕ ГЛАВНОЕ

Таблицу значений, которые принимает логическое выражение при всех сочетаниях значений (наборах) входящих в него переменных, называют таблицей истинности логического выражения.

Истинность логического выражения можно доказать путём построения его таблицы истинности.

Функцию от n переменных, аргументы которой и сама функция принимают только два значения — 0 и 1, называют логической функцией. Таблица истинности может рассматриваться как способ задания логической функции.

Вопросы и задания

1. Что представляет собой таблица истинности?
2. Составлена таблица истинности для логического выражения, содержащего n переменных. Известно m — количество строк, в которых выражение принимает значение 0. Требуется выяснить, в скольких случаях логическое выражение примет значение 1 при следующих значениях n и m :
 - 1) $n = 6, m = 15$;
 - 2) $n = 7, m = 100$;
 - 3) $n = 10, m = 500$.

3. Постройте таблицы истинности для следующих логических выражений:

- 1) $(A \rightarrow B) \leftrightarrow (A \& B)$;
- 2) $(A \rightarrow B) \rightarrow ((A \rightarrow \bar{B}) \rightarrow \bar{A})$;
- 3) $(A \rightarrow (C \rightarrow B)) \rightarrow (B \vee C)$.

4. Рассмотрите два составных высказывания:

- F_1 = «Если одно слагаемое делится на 3 и сумма делится на 3, то и другое слагаемое делится на 3»;
- F_2 = «Если одно слагаемое делится на 3, а другое слагаемое не делится на 3, то сумма не делится на 3».

Формализуйте эти высказывания, построьте таблицы истинности для каждого из полученных выражений и убедитесь, что результирующие столбцы совпадают.

5. Логическое выражение, являющееся истинным при любом наборе входящих в него переменных, называется тождественно истинным. Убедитесь, что следующие логические выражения являются тождественно истинными:

- 1) $A \rightarrow (B \rightarrow A)$;
- 2) $(A \rightarrow \bar{B}) \rightarrow (B \rightarrow \bar{A})$;
- 3) $(A \& C \rightarrow B) \rightarrow (C \rightarrow (A \vee B \rightarrow B \& C))$.

6. Какое из приведённых логических выражений равносильно выражению $(A \rightarrow C) \& (B \rightarrow C)$?

- 1) $A \& B \rightarrow C$;
- 2) $A \rightarrow B \rightarrow C$;
- 3) $A \vee B \rightarrow C$;
- 4) $A \leftrightarrow B \rightarrow C$.

7. Известен фрагмент таблицы истинности для логического выражения F , содержащего логические переменные A , B и C .

A	B	C	F
0	1	0	0
1	0	0	1
1	0	1	0
1	1	1	1

Какое из приведённых далее логических выражений соответствует этому фрагменту?

- 1) $A \& C \vee (B \rightarrow A)$;
- 2) $(A \vee B) \& (C \rightarrow A)$;
- 3) $(A \& B \vee C) \& (B \rightarrow A \& C)$;
- 4) $(\overline{B \rightarrow A}) \vee (C \vee A \rightarrow B)$;

5) ни одна из указанных формул.

8. Логическая функция F задаётся выражением

$$(A \& B \& \overline{C}) \vee (A \& B \& C) \vee (A \& \overline{B} \& \overline{C}).$$

Ниже приведён фрагмент таблицы истинности, содержащий все наборы переменных, на которых F ложна.

?	?	?	F
0	0	0	0
0	0	1	0
0	1	1	0
1	0	0	0
1	0	1	0

Какому столбцу таблицы истинности функции F соответствует каждая из переменных A , B , C ?

§ 20

Преобразование логических выражений

Способ определения истинности логического выражения путём построения его таблицы истинности становится неудобным при увеличении количества логических переменных, т. к. за счёт существенного увеличения числа строк таблицы становятся громоздкими. В таких случаях выполняются преобразования логических выражений в равносильные. Для этого используют свойства логических операций, которые иначе называют законами алгебры логики.

20.1. Основные законы алгебры логики

Приведём основные законы алгебры логики.

1. **Переместительные (коммутативные) законы:**
 $A \& B = B \& A;$
 $A \vee B = B \vee A.$
2. **Сочетательные (ассоциативные) законы:**
 $(A \& B) \& C = A \& (B \& C);$
 $(A \vee B) \vee C = A \vee (B \vee C).$
3. **Распределительные (дистрибутивные) законы:**
 $A \& (B \vee C) = (A \& B) \vee (A \& C);$
 $A \vee (B \& C) = (A \vee B) \& (A \vee C).$
4. **Законы идемпотентности (отсутствия степеней и коэффициентов):**
 $A \& A = A;$
 $A \vee A = A.$
5. **Закон противоречия:**
 $A \& \bar{A} = 0.$
6. **Закон исключённого третьего:**
 $A \vee \bar{A} = 1.$
7. **Закон двойного отрицания:**
 $\overline{\bar{A}} = A.$
8. **Законы работы с константами:**
 $A \vee 1 = 1; A \vee 0 = A;$
 $A \& 1 = A; A \& 0 = 0.$
9. **Законы де Моргана:**
 $\overline{A \& B} = \bar{A} \vee \bar{B};$
 $\overline{A \vee B} = \bar{A} \& \bar{B}.$
10. **Законы поглощения:**
 $A \& (A \vee B) = A;$
 $A \vee (A \& B) = A.$

Справедливость законов можно доказать построением таблиц истинности.

Пример 1. Упростим логическое выражение

$$A \& B \& C \vee A \& B \& \bar{C}.$$

Последовательно применим дистрибутивный закон и закон исключённого третьего:

$$A \& B \& C \vee A \& B \& \bar{C} = A \& B \& (C \vee \bar{C}) = A \& B \& 1 = A \& B.$$

Пример 2. Упростим логическое выражение

$$(A \vee B) \& (A \vee B \vee C) \& (A \vee B \vee \bar{C}).$$

$$(A \vee B) \& (A \vee B \vee C) \& (A \vee B \vee \bar{C}) = (A \vee B) \& (0 \vee C \vee \bar{C}) = \\ = (A \vee B) \& 1 = A \vee B.$$

Аналогичные законы выполняются для операций объединения, пересечения и дополнения множеств. Например:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C),$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C),$$

$$\overline{A \cap B} = \bar{A} \cup \bar{B};$$

$$\overline{A \cup B} = \bar{A} \cap \bar{B}.$$

Пробуйте самостоятельно доказать один из этих законов с помощью кругов Эйлера.

Пример 3. На числовой прямой даны отрезки $B = [2; 12]$ и $C = [7; 18]$. Каким должен быть отрезок A , чтобы предикат $(x \in A) \vee ((x \in B) \rightarrow (x \in C))$ становился истинным высказыванием при любых значениях x .

Преобразуем исходное выражение, избавившись от импликации:

$$(x \in A) \vee ((x \in B) \rightarrow (x \in C)) = (x \in A) \vee ((\overline{x \in B}) \vee (x \in C)) = \\ = (x \in A) \vee (\overline{x \in B}) \vee (x \in C).$$

A , B и C — множества. Для них можем записать:

$$A \cup \bar{B} \cup C = U.$$

Известно, что $A \cup \bar{A} = U$.

Будем считать, что $\bar{A} = \bar{B} \cup C$. Тогда $A = \overline{\bar{B} \cup C} = B \cap \bar{C}$, причём это минимально возможное множество A .

Множество B — это отрезок $[2; 12]$.

Множество \bar{C} — это промежутки $]-\infty; 7[$ и $]18; +\infty[$.

Изобразим это графически:



Пересечением этих множеств будет служить промежуток $[2; 7[$. В качестве ответа мы можем взять этот промежуток, а также любой другой, его включающий.

Чему равна минимальная длина отрезка A ? Укажите ещё несколько вариантов множества A .

Пример 4. Для какого наименьшего неотрицательного целого десятичного числа a выражение

$$(x \& 28 \neq 0 \vee x \& 45 \neq 0) \rightarrow (x \& 17 = 0 \rightarrow x \& a \neq 0)$$

тождественно истинно (т. е. принимает значение 1 при любом неотрицательном целом значении десятичной переменной x)? Здесь $\&$ — поразрядная конъюнкция двух неотрицательных целых десятичных чисел.

Прежде всего, вспомним, что представляет собой поразрядная конъюнкция двух целых десятичных чисел, например 27 и 22.

$$27 = 11011_2, \quad 22 = 10110_2.$$

$$27 \& 22 = 11011 \& 10110 = 10010_2 = 18_{10}.$$

$$\begin{array}{r} 11011 \\ \& 10110 \\ \hline 10010 \end{array}$$

Обратите внимание на то, что если в некотором бите хотя бы одного сомножителя есть 0, то 0 есть и в этом бите результата, а 1 в результате получается только тогда, когда в соответствующих битах каждого сомножителя есть 1.

Введём обозначения:

$$M(x) = (x \& 28 \neq 0), \quad N(x) = (x \& 45 \neq 0),$$

$$K(x) = (x \& 17 = 0), \quad A(x) = (x \& a \neq 0).$$

Перепишем исходное выражение в наших обозначениях:

$$(M(x) \vee N(x)) \rightarrow (K(x) \rightarrow A(x)).$$

Преобразуем полученное выражение:

$$\begin{aligned} (M(x) \vee N(x)) \rightarrow (K(x) \rightarrow A(x)) &= \overline{M(x) \vee N(x)} \vee \overline{K(x)} \vee A(x) = \\ &= \overline{(M(x) \vee N(x)) \& K(x)} \vee A(x). \end{aligned}$$

Рассмотрим предикат $M(x) = (x \& 28 \neq 0)$. В числе $28 = 11100_2$ 4-й, 3-й и 2-й биты содержат единицы, а 1-й и 0-й — нули. Следовательно, множеством истинности этого предиката являются такие числа x , у которых хотя бы один из битов с номерами 4, 3 или 2 содержит единицу. Если и 4-й, и 3-й, и 2-й биты числа x нулевые, то высказывание $x \& 28 \neq 0$ будет ложным.

Рассмотрим предикат $N(x) = (x \& 45 \neq 0)$. В числе $45 = 101101_2$ 5-й, 3-й, 2-й и 0-й биты содержат единицы, 4-й и 1-й — нули.

Следовательно, множеством истинности этого предиката являются такие числа x , у которых хотя бы один из битов с номерами 5, 3, 2 или 0 содержит единицу. Если и 5-й, и 3-й, и 2-й, и 0-й биты числа x нулевые, то высказывание $x \& 45 \neq 0$ будет ложным.

Рассмотрим предикат $K(x) = (x \& 17 = 0)$. В числе $17 = 10001_2$ 3-й, 2-й и 1-й биты содержат нули, 4-й и 0-й — единицы. Побитовая конъюнкция 17 и x будет равна нулю, если в числе x 4-й и 0-й биты будут содержать нули. Множество истинности этого предиката — все x с нулями в 4-м и 0-м битах.

По условию задачи надо, чтобы $(M(x) \vee N(x)) \& K(x) \vee A(x) = 1$. Запишем это выражение для рассмотренных множеств истинности:

$$\overline{(M \cup N) \cap K} \cup A = U.$$

Так как $A \cup \overline{A} = U$, примем $A = (M \cup N) \cap K$.

Объединением множеств M и N являются все двоичные числа, у которых хотя бы один из битов с номерами 5, 4, 3, 2, 0 содержит единицу. Пересечением этого множества с множеством K будут все двоичные числа, у которых биты с номерами 4 и 0 будут заняты нулями, т. е. такие двоичные числа, у которых хотя бы один из битов с номерами 5, 3, 2 содержит 1. Все эти числа образуют множество A .

Искомое число a должно быть таким, чтобы при любом неотрицательном целом значении переменной x : $x \& a \neq 0$, и кроме того, оно должно быть минимальным из возможных. Этим условиям удовлетворяет число 101100_2 . Действительно, единицы в нём стоят в тех и только в тех битах, которые нужны для выполнения условия $x \& a \neq 0$.

Итак, требуемое число 101100_2 или 44_{10} .

Приведите пример такого десятичного числа a , что для него $x \& a \neq 0$ при любом неотрицательном целом значении десятичной переменной x , но само число a не является минимально возможным.

Пример 5. Выясним, сколько решений имеет следующая система из двух уравнений:

$$\begin{cases} (x_1 \rightarrow x_2) \& (x_2 \rightarrow x_3) \& (x_3 \rightarrow x_4) = 1; \\ (\bar{y}_1 \vee y_2) \& (\bar{y}_2 \vee y_3) \& (\bar{y}_3 \vee y_4) = 1. \end{cases}$$



Рассмотрим первое уравнение:

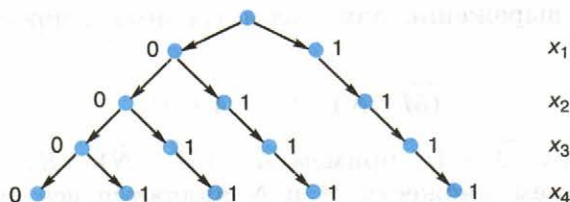
$$(x_1 \rightarrow x_2) \& (x_2 \rightarrow x_3) \& (x_3 \rightarrow x_4) = 1.$$

Конъюнкция истинна тогда и только тогда, когда истинны все образующие её высказывания. Следовательно, каждая из трёх входящих в конъюнкцию импликаций должна быть равна 1.

Начнем строить дерево решений, представив на нём значения переменных x_1 и x_2 при которых $x_1 \rightarrow x_2 = 1$.

Продолжим строить дерево решений. Значения переменной x_3 будем выбирать такими, чтобы при имеющихся x_2 импликация $x_2 \rightarrow x_3$ оставалась истинной.

То же самое сделаем для переменной x_4 .



На дереве видно, что рассматриваемое нами уравнение имеет 5 решений — 5 разных наборов значений логических переменных x_1, x_2, x_3, x_4 , при которых выполняется равенство:

$$(x_1 \rightarrow x_2) \& (x_2 \rightarrow x_3) \& (x_3 \rightarrow x_4) = 1.$$

Так как $a \rightarrow b = \bar{a} \vee b$, второе уравнение системы:

$$(\bar{y}_1 \vee y_2) \& (\bar{y}_2 \vee y_3) \& (\bar{y}_3 \vee y_4) = 1$$

можно преобразовать к виду:

$$(y_1 \rightarrow y_2) \& (y_2 \rightarrow y_3) \& (y_3 \rightarrow y_4) = 1.$$

Следовательно, как и первое уравнение, это уравнение имеет 5 решений. Представим их в табличной форме:

y_1	y_2	y_3	y_4
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1

Решение исходной системы логических уравнений — это множество различных наборов значений логических переменных $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$ таких, что при подстановке каждого из них в систему оба уравнения превращаются в истинные равенства.

Начнём строить такие наборы или двоичные цепочки. Их началом может служить любой из пяти наборов — решений первого уравнения, а концом — любой из пяти наборов — решений второго уравнения. Например, на основе одного из решений первого уравнения можно построить следующие пять решений системы:

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4
0	0	0	0	0	0	0	0
				0	0	0	1
				0	0	1	1
				0	1	1	1
				1	1	1	1

Всего мы можем построить $5 \cdot 5 = 25$ решений системы.

Вспомните, как называется теорема комбинаторики, которую мы применили для подсчёта количества решений системы.



20.2. Логические функции

Значение любого логического выражения определяется значениями входящих в него логических переменных. Тем самым логическое выражение может рассматриваться как способ задания логической функции.

Совокупность значений n аргументов удобно интерпретировать как строку нулей и единиц длины n . Существует ровно 2^n различных двоичных строк длины n . Так как на каждой такой строке некая функция может принимать значение 0 или 1, общее количество различных булевых функций от n аргументов равно 2^{2^n} .

Для $n = 2$ существует 16 различных логических функций.



Рассмотрим их подробнее.

A	B	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇	F ₈	F ₉	F ₁₀	F ₁₁	F ₁₂	F ₁₃	F ₁₄	F ₁₅	F ₁₆
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$F_1(A, B) = 0$ — константа «ложь»;

$F_2(A, B) = A \& B$ — конъюнкция;

$F_3(A, B) = \overline{A \rightarrow B}$ — отрицание импликации;

$F_4(A, B) = A$ — функция, равная первому аргументу;

$F_5(A, B) = B \rightarrow A$ — отрицание обратной импликации;

$F_6(A, B) = B$ — функция, равная второму аргументу;

$F_7(A, B) = A \oplus B$ — строгая дизъюнкция;

$F_8(A, B) = A \vee B$ — дизъюнкция;

$F_9(A, B) = A \downarrow B$ — стрелка Пирса (отрицание дизъюнкции, ИЛИ-НЕ);

$F_{10}(A, B) = A \leftrightarrow B$ — эквиваленция;

$F_{11}(A, B) = \overline{B}$ — отрицание второго аргумента;

$F_{12}(A, B) = B \rightarrow A$ — обратная импликация;

$F_{13}(A, B) = \overline{A}$ — отрицание первого аргумента;

$F_{14}(A, B) = A \rightarrow B$ — импликация;

$F_{15}(A, B) = A | B$ — штрих Шеффера (отрицание конъюнкции, И-НЕ);

$F_{16}(A, B) = 1$ — константа «истина».

С увеличением числа аргументов количество логических функций резко возрастает. Так, для трёх переменных существует 256 различных логических функций! Но изучать их все нет никакой необходимости. Дело в том, что путём преобразований функция любого количества переменных может быть выражена через функции только двух переменных. Более того, можно использовать не все, а лишь некоторые логические функции двух переменных. Например:

- 1) F_2 и F_{11} (конъюнкция и отрицание второго аргумента);
- 2) F_8 и F_{13} (дизъюнкция и отрицание первого аргумента);
- 3) F_9 (стрелка Пирса, отрицание дизъюнкции);
- 4) F_{15} (штрих Шеффера, отрицание конъюнкции).

Два последних примера говорят о том, что при желании всю алгебру логики можно свести к одной функции! Но чаще всего логические функции записываются в виде логического выражения через отрицание, конъюнкцию и дизъюнкцию.

20.3. Составление логического выражения по таблице истинности и его упрощение

Ранее мы выяснили, что для любого логического выражения можно составить таблицу истинности. Справедливо и обратное: для всякой таблицы истинности можно составить соответствующее ей логическое выражение.

Алгоритм составления логического выражения по таблице истинности достаточно прост. Для этого надо:

- 1) отметить в таблице истинности наборы переменных, при которых значение логического выражения равно единице;
- 2) для каждого отмеченного набора записать конъюнкцию всех переменных следующим образом: если значение некоторой переменной в этом наборе равно 1, то в конъюнкцию включаем саму переменную, в противном случае — её отрицание;
- 3) все полученные конъюнкции связать операциями дизъюнкции.

Пример 6. Имеется следующая таблица истинности:

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0



После выполнения двух первых шагов алгоритма получим:

A	B	C	F	
0	1	0	1	$\bar{A} \& B \& \bar{C}$
0	1	1	1	$\bar{A} \& B \& C$
1	1	0	1	$A \& B \& \bar{C}$

После выполнения третьего шага получаем логическое выражение:

$$\bar{A} \& B \& \bar{C} \vee \bar{A} \& B \& C \vee A \& B \& \bar{C}.$$

Попробуем упростить полученное логическое выражение. Прежде всего, вынесем за скобки B — общий множитель, имеющийся у всех трёх слагаемых, затем — множитель \bar{A} , а далее используем законы алгебры логики.

$$\begin{aligned} & \bar{A} \& B \& \bar{C} \vee \bar{A} \& B \& C \vee A \& B \& \bar{C} = \\ & = B \& (\bar{A} \& \bar{C} \vee \bar{A} \& C \vee A \& \bar{C}) = \\ & = B \& (\bar{A} \& (\bar{C} \vee C) \vee A \& \bar{C}) = B \& (\bar{A} \& 1 \vee A \& \bar{C}) = \\ & = B \& (\bar{A} \vee A \& \bar{C}) = B \& (\bar{A} \vee A) \& (\bar{A} \vee \bar{C}) = \\ & = B \& 1 \& (\bar{A} \vee \bar{C}) = B \& (\bar{A} \vee \bar{C}) = B \& \overline{A \& C}. \end{aligned}$$

САМОЕ ГЛАВНОЕ





Способ определения истинности логического выражения путём построения его таблицы истинности становится неудобным при увеличении количества логических переменных, т. к. за счёт существенного увеличения числа строк таблицы становятся громоздкими. В таких случаях выполняются преобразования логических выражений в равносильные. Для этого используют свойства логических операций, которые иначе называют законами алгебры логики. Аналогичные законы имеют место и в алгебре множеств.

Логическая функция может быть задана с помощью таблицы истинности или аналитически, т. е. с помощью логического выражения.

Для всякой таблицы истинности можно составить соответствующее ей логическое выражение.

Вопросы и задания



- Какие из рассмотренных законов алгебры логики аналогичны законам алгебры чисел, а какие нет?
- Докажите второй закон де Моргана с помощью таблиц истинности.
- Путём преобразования докажите равносильность следующих высказываний:
 - $\overline{(A \& \bar{B}) \vee (B \& \bar{C})}$ и $(\bar{A} \& \bar{B}) \vee (\bar{A} \& C) \vee (B \& C)$;
 - $(A \& B) \vee \overline{(A \& \bar{C})}$ и $(A \& B) \vee A \vee \bar{C}$.
- Упростите логические формулы:
 - $(A \& B \& \bar{C}) \vee (A \& B \& C) \vee (A \& B)$;
 - $(A \& B \vee A \& B \& \bar{C} \vee B \& \bar{C} \vee C) \& (\bar{C} \vee A \& C \vee \bar{A} \& B \& \bar{C})$.
- Найдите X , если $\overline{(X \vee A)} \vee \overline{(X \vee \bar{A})} = B$.
- На числовой прямой даны два отрезка: $P = [10; 25]$ и $Q = [20; 55]$. Укажите наибольшую возможную длину такого отрезка A , что выражение $(x \in A) \rightarrow ((x \in P) \vee (x \in Q))$ истинно при любом значении переменной x . 
- Элементами множеств A , P и Q являются натуральные числа, причём $P = \{2, 4, 6, 8, 10, 12\}$ и $Q = \{2, 6, 12, 18, 24\}$. Известно, что выражение $(x \in Q) \rightarrow ((x \in A) \rightarrow (x \in P))$ истинно при любом значении переменной x . Определите наименьшее возможное количество элементов множества A . 
- На числовой прямой даны два отрезка: $M = [10; 60]$ и $N = [40; 80]$. Укажите наименьшую возможную длину такого отрезка A , что выражение $(x \in M) \rightarrow (((x \in N) \& (x \in A)) \rightarrow (x \in M))$ истинно при любом значении переменной x . 
- Для какого наименьшего неотрицательного целого десятичного числа A формула $x \& 25 \neq 0 \rightarrow (x \& 17 = 0 \rightarrow x \& A \neq 0)$ тождественно истинна, т. е. принимает значение 1 при любом неотрицательном целом значении десятичной переменной x ? (Здесь $\&$ — поразрядная конъюнкция двух неотрицательных целых десятичных чисел.) 

*10. Определите наибольшее натуральное десятичное число A , при котором выражение $((x \& 46 = 0) \vee (x \& 18 = 0)) \rightarrow \rightarrow ((x \& 115 \neq 0) \vee (x \& A = 0))$ тождественно истинно, т. е. принимает значение 1 при любом натуральном значении десятичной переменной x . (Здесь $\&$ — поразрядная конъюнкция двух неотрицательных целых десятичных чисел.)

11. Сколько различных решений имеет система уравнений:

$$1) \begin{cases} x_1 \cdot x_2 \rightarrow x_3 \cdot x_4 = 1; \\ \bar{x}_3 + \bar{x}_4 + x_5 \cdot x_6 = 1. \end{cases}$$

$$2) \begin{cases} \bar{x}_1 + \bar{x}_2 + x_3 \cdot x_4 = 1; \\ \bar{x}_3 + \bar{x}_4 + x_5 \cdot x_6 = 1; \\ \bar{x}_5 + \bar{x}_6 + x_7 \cdot x_8 = 1; \\ \bar{x}_7 + \bar{x}_8 + x_9 \cdot x_{10} = 1. \end{cases}$$

12. Сколько существует различных логических функций от четырёх переменных?

13. По заданной таблице истинности составьте логические выражения для функций F_1, F_2 .

A	B	F_1	F_2
0	0	1	0
0	1	1	1
1	0	0	0
1	1	0	1

14. По известным таблицам истинности запишите аналитическое представление импликации, эквиваленции и строгой дизъюнкции.

15. Логические функции штрих Шеффера и стрелка Пирса называются так в честь математиков, исследовавших их свойства. Подготовьте краткую биографическую справку об одном из этих учёных.

16. По заданной таблице истинности составьте логические выражения для функций F_1, F_2 .

A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

17. Запишите логическое выражение для логической функции $F(A, B, C)$, равной 1 на наборах 011, 101, 110, 111. Попытайтесь упростить полученное выражение.

§ 21

Элементы схемотехники.
Логические схемы

Любое устройство компьютера, выполняющее арифметические или логические операции, может рассматриваться как преобразователь двоичной информации: значения входных переменных для него — последовательность нулей и единиц, а значение выходной функции — новая двоичная последовательность. Необходимые преобразования информации в блоках компьютера производятся логическими устройствами двух типов: комбинационными схемами и цифровыми автоматами с памятью.

В комбинационной схеме набор выходных сигналов в любой момент времени полностью определяется набором входных сигналов.

В цифровых автоматах с памятью набор выходных сигналов зависит не только от набора входных сигналов, но и от внутреннего состояния данного устройства. Такие устройства всегда имеют память.

Схемотехника — научно-техническое направление, занимающееся проектированием, созданием и отладкой электронных схем и электронных устройств различного назначения.

21.1. Логические элементы

Логический элемент — это устройство с n входами и одним выходом, которое преобразует входные двоичные сигналы в двоичный сигнал на выходе.

Работу любого логического элемента математически удобно описать как логическую функцию, которая упорядоченному набору из нулей и единиц ставит в соответствие значение, также равное нулю или единице.

В схемотехнике широко используются логические элементы, представленные в таблице 4.2.

Таблица 4.2

Условные обозначения типовых логических элементов

Наименование элемента	Условное обозначение	Название функции и её формула
И		Конъюнкция $F = A \& B$
ИЛИ		Дизъюнкция $F = A \vee B$
НЕ		Инверсия $F = \bar{A}$
И-НЕ		Штрих Шеффера $F = \overline{A \& B}$
ИЛИ-НЕ		Стрелка Пирса $F = \overline{A \vee B}$

Логический элемент И (конъюнктор) реализует операцию логического умножения. Единица на выходе этого элемента появится тогда и только тогда, когда на всех входах будут единицы.

Опишите подобным образом логические элементы ИЛИ (дизъюнктор), НЕ (инвертор), И-НЕ, ИЛИ-НЕ.

Однотипность сигналов на входах и выходах позволяет подавать сигнал, вырабатываемый одним элементом, на вход другого элемента. Это позволяет из двухвходовых элементов «собирать» многовходовые элементы (рис 4.7), а также синтезировать произвольные комбинационные схемы, соединяя в цепочки отдельные логические элементы.

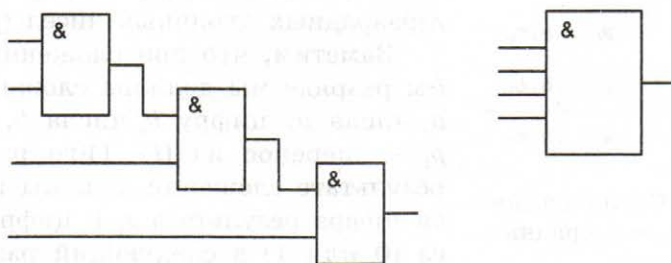


Рис. 4.7. Схема и обозначение четырёхвходового конъюктора

Пример. По заданной логической функции $F(A, B) = \overline{A} \& B \vee A \& \overline{B}$ построим комбинационную схему (рис. 4.8).

Построение начнём с логической операции, которая должна выполняться последней. В данном случае такой операцией является логическое сложение, следовательно, на выходе логической схемы должен быть дизъюнктор. На него сигналы подаются с двух конъюкторов, на которые в свою очередь подаются один входной сигнал нормальный и один инвертированный (с инверторов).

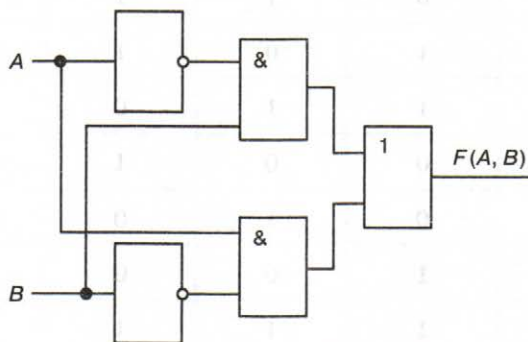


Рис. 4.8. Комбинационная схема функции $F(A, B) = \overline{A} \& B \vee A \& \overline{B}$



21.2. Сумматор

Из отдельных логических элементов можно составить устройства, производящие арифметические операции над двоичными числами.

Электронная логическая схема, выполняющая суммирование двоичных чисел, называется сумматором.

$$\begin{array}{r}
 \\
 a = a_n a_{n-1} \dots a_i \dots a_1 a_0 \\
 + b = b_n b_{n-1} \dots b_i \dots b_1 b_0 \\
 \hline
 s_{n+1} s_n s_{n-1} \dots s_i \dots s_1 s_0
 \end{array}$$

Рис. 4.9. Схема сложения двух n -разрядных двоичных чисел

Вспомним схему сложения двух n -разрядных двоичных чисел (рис. 4.9).

Заметим, что при сложении цифр в i -м разряде мы должны сложить цифру a_i числа a , цифру b_i числа b , а также p_i — перенос из $(i-1)$ -го разряда. В результате сложения должны получить цифру результата s_i и цифру переноса (0 или 1) в следующий разряд p_{i+1} .

Основываясь на этих рассуждениях, построим таблицу истинности для функций, которые в зависимости от цифр a_i , b_i и p_i получают цифры s_i и p_{i+1} .

Входы			Выходы	
a_i	b_i	p_i	s_i	p_{i+1}
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

Вам известен алгоритм построения логического выражения по таблице истинности. Воспользуемся им и запишем выражение для функции p_{i+1} :

$$p_{i+1} = \bar{a}_i \& b_i \& p_i \vee a_i \& \bar{b}_i \& p_i \vee a_i \& b_i \& \bar{p}_i \vee a_i \& b_i \& p_i.$$

Попытаемся упростить это выражение, воспользовавшись тем, что $A \vee A = A$. Основываясь на этом законе, включим в имеющуюся дизъюнкцию ещё два слагаемых вида $a_i \& b_i \& p_i$, причём на основании коммутативного и ассоциативного законов преобразуем полученное выражение к виду:

$$\begin{aligned} p_{i+1} &= (\bar{a}_i \& b_i \& p_i \vee a_i \& b_i \& p_i) \vee (a_i \& \bar{b}_i \& p_i \vee a_i \& b_i \& p_i) \vee \\ &\vee (a_i \& b_i \& \bar{p}_i \vee a_i \& b_i \& p_i) = [\text{по законам дистрибутивности}] = \\ &= b_i \& p_i \& (\bar{a}_i \vee a_i) \vee a_i \& p_i \& (\bar{b}_i \vee b_i) \vee a_i \& b_i \& (\bar{p}_i \vee p_i) = \\ &= [\text{по закону исключённого третьего}] = b_i \& p_i \vee a_i \& p_i \vee a_i \& b_i. \end{aligned}$$

Полученное выражение означает, что функция p_{i+1} принимает значение 1 только для таких комбинаций входных переменных, когда хотя бы две переменные имеют единичные значения. Обратите внимание на то, что такой вывод можно сделать и в результате анализа таблицы истинности.

По таблице истинности можем записать выражение для s_i :

$$s_i = \bar{a}_i \& \bar{b}_i \& p_i \vee \bar{a}_i \& b_i \& \bar{p}_i \vee a_i \& \bar{b}_i \& \bar{p}_i \vee a_i \& b_i \& p_i.$$

Его также можно попытаться преобразовать к более короткому виду. Но можно пойти другим путём и провести более тщательный анализ таблицы истинности для функции s_i .

Из таблицы видно, что значение s_i равно 1, если все входные сигналы равны 1. Этому соответствует выражение $a_i \& b_i \& p_i = 1$.

Или значение s_i равно 1, если в комбинации входных сигналов есть единственная 1, т. е. единица среди переменных есть, но нет одновременно двух переменных, значения которых равны 1. Это можно записать так:

$$(a_i \vee b_i \vee p_i) \& \bar{p}_{i+1} = 1.$$

Следовательно, s_i можно записать так:

$$s_i = (a_i \vee b_i \vee p_i) \& \bar{p}_{i+1} \vee a_i \& b_i \& p_i.$$

Можно попытаться самостоятельно провести преобразование логического выражения, полученного по таблице истинности для s_i к итоговому виду. Но, чтобы убедиться в равносильности этих двух выражений, достаточно построить таблицу истинности для второго из них.



Полученные выражения позволяют реализовать одноразрядный двоичный сумматор схемой, представленной на рисунке 4.10.

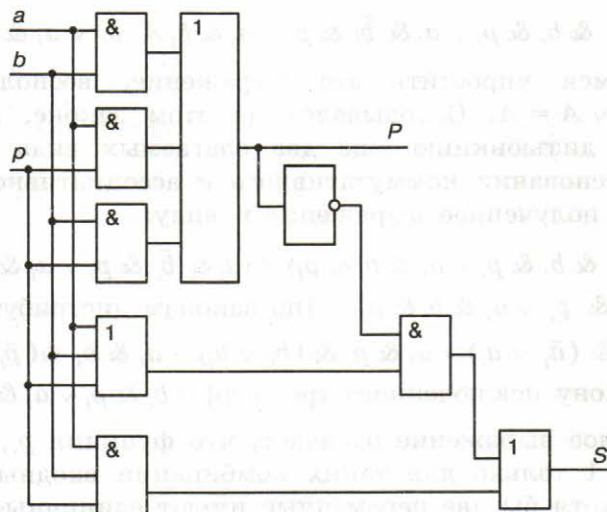
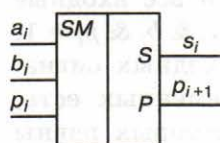


Рис. 4.10. Схема одноразрядного сумматора

Выразить s_i и p_{i+1} можно и другими формулами. Например, самое короткое выражение для s_i имеет вид: $s_i = a_i \oplus b_i \oplus p_i$, что позволяет построить сумматор, используя другие логические элементы.



Сложение n -разрядных двоичных чисел осуществляется с помощью комбинации одноразрядных сумматоров (условное обозначение одноразрядных сумматоров приведено на рисунке слева).

21.3. Триггер



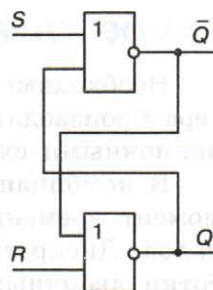
Триггер (от англ. *trigger* — защёлка, спусковой крючок) — логический элемент, способный хранить один разряд двоичного числа.

Триггер был изобретён в 1918 году М. А. Бонч-Бруевичем.



Михаил Александрович Бонч-Бруевич (1888–1940) — русский и советский радиотехник, основатель отечественной радиоламповой промышленности. Работал в области радиовещания и дальней связи на коротких волнах. В 1918 году М. А. Бонч-Бруевич предложил схему переключающего устройства, имеющего два устойчивых рабочих состояния, под названием «катодное реле». Это устройство впоследствии было названо триггером.

Самый простой триггер — RS. Он состоит из двух логических элементов ИЛИ-НЕ, входы и выходы которых соединены кольцом: выход первого соединён со входом второго и выход второго — со входом первого. Схема RS-триггера представлена на рисунке 4.11.



Триггер имеет два входа: S (от англ. *set* — установка) и R (от англ. *reset* — сброс) и два выхода: Q (прямой) и \bar{Q} (инверсный). Принцип его работы иллюстрирует следующая таблица истинности:

Рис. 4.11. Логическая схема RS-триггера

Режим работы триггера	Входы		Состояние триггера Q
	R	S	
Хранение предыдущего состояния	0	0	Q
Установка триггера в 1	0	1	1
Установка триггера в 0	1	0	0
Запрещённое состояние	1	1	Недопустимо

Если на входы поступают сигналы $R = 0$ и $S = 0$, то триггер находится в режиме хранения — на выходах Q и \bar{Q} сохраняются установленные ранее значения.

Если на установочный вход S на короткое время поступает сигнал 1, то триггер переходит в состояние 1 и после того, как сигнал на входе S станет равен 0, триггер будет сохранять это состояние, т. е. будет хранить 1.

При подаче 1 на вход R триггер перейдёт в состояние 0.

Подача на оба входа S и R логической единицы может привести к неоднозначному результату, поэтому такая комбинация входных сигналов запрещена.

Триггер используется для хранения информации в оперативной памяти компьютера, а также во внутренних регистрах процессора. Для хранения одного байта информации необходимо 8 триггеров, для килобайта — $8 \cdot 1024$ триггеров. Оперативная память современных компьютеров содержит миллионы триггеров.

В целом же компьютер состоит из огромного числа логических устройств, образующих все его узлы и память.

САМОЕ ГЛАВНОЕ

Необходимые преобразования информации в блоках компьютера производятся логическими устройствами двух типов: комбинационными схемами и цифровыми автоматами с памятью.

В комбинационной схеме набор выходных сигналов в любой момент времени полностью определяется набором входных сигналов. Дискретный преобразователь, который выдаёт после обработки двоичных сигналов значение одной из логических операций, называется логическим элементом. Электронная логическая схема, выполняющая суммирование двоичных чисел, называется сумматором.

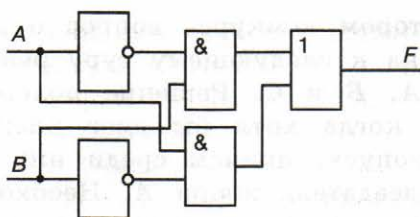
В цифровых автоматах с памятью набор выходных сигналов зависит не только от набора входных сигналов, но и от внутреннего состояния данного устройства. Такие устройства всегда имеют память. Триггер — логический элемент, способный хранить один разряд двоичного числа. Оперативная память современных компьютеров содержит миллионы триггеров.

В целом же компьютер состоит из огромного числа логических устройств, образующих все его узлы и память.

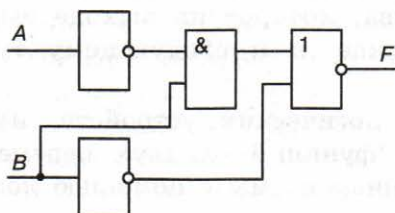


Вопросы и задания

1. Что такое логический элемент? Перечислите базовые логические элементы?
2. По логическому выражению $(A \vee \bar{B}) \& \bar{C} \& (B \vee \bar{A}) \& \bar{C}$ требуется разработать логическое устройство. Какие логические элементы необходимы для его создания?
3. Найдите значение выходного сигнала в приведенной схеме, если:
1) $A = 0$ и $B = 0$; 3) $A = 1$ и $B = 0$;
2) $A = 0$ и $B = 1$; 4) $A = 1$ и $B = 1$.



4. Определите логическое выражение преобразования, выполняемого схемой:



5. Постройте логические схемы для следующих функций:

1) $F = \overline{(A \& B \& C)} \vee B \& C \vee \bar{A}$;

2) $F = B \vee (C \& \bar{A}) \vee (A \& B)$.

6. Постройте схему устройства, выполняющего преобразование информации в соответствии с данной таблицей истинности:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

7. Пусть в некотором конкурсе вопрос о допуске того или иного участника к следующему туру решается тремя членами жюри: A , B и C . Решение положительно тогда и только тогда, когда хотя бы двое членов жюри высказываются за допуск, причём среди них обязательно должен быть председатель жюри A . Необходимо разработать устройство для голосования, в котором каждый член жюри нажимает на одну из двух кнопок — «За» или «Против», а результат голосования всех трёх членов жюри определяется по тому, загорится (участник допускается) или нет (участник не допускается) сигнальная лампочка. Составьте схему устройства, которое на выходе выдавало бы 1, если участник допускается к следующему туру, и 0, если не допускается.
8. Существует 16 логических устройств, имеющих два входа (16 логических функций от двух переменных). Реализуйте их комбинационные схемы с помощью логических элементов И, ИЛИ, НЕ.
9. Если при суммировании не учитывается признак переноса, то соответствующая логическая схема называется *полусумматором*. По имеющейся таблице истинности постройте логическую схему полусумматора.

Входы		Выходы	
a_i	b_i	s_i	P_{i+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

10. Что такое триггер? В чём основное отличие триггера от таких логических элементов, как инвертор или конъюнктор?
11. Подготовьте краткую биографическую справку о нашем выдающемся соотечественнике М. А. Бонч-Бруевиче. В чём заключается его вклад в развитие вычислительной техники?

§ 22

Логические задачи
и способы их решения

Исходными данными в логических задачах являются высказывания. При этом высказывания и взаимосвязи между ними бывают так сложны, что разобраться в них без использования специальных методов бывает достаточно трудно.

22.1. Метод рассуждений

Основная идея этого метода состоит в том, чтобы последовательно анализировать всю информацию, имеющуюся в задаче, и делать на этой основе выводы.

Пример 1. На одной улице стоят в ряд 4 дома, в каждом из них живёт по одному человеку. Их зовут Василий, Семён, Геннадий и Иван. Известно, что все они имеют разные профессии: скрипач, столяр, охотник и врач. Известно, что:

- 1) столяр живёт правее охотника;
- 2) врач живёт левее охотника;
- 3) скрипач живёт с краю;
- 4) скрипач живёт рядом с врачом;
- 5) Семён не скрипач и не живёт рядом со скрипачом;
- 6) Иван живёт рядом с охотником;
- 7) Василий живёт правее врача;
- 8) Василий живёт через дом от Ивана.

Определим, кто где живёт.

Изобразим дома прямоугольниками и пронумеруем их:



Известно, что скрипач живёт с краю (3). Следовательно, он может жить в доме 1 или в доме 4.



Глава 4. Теория множеств и алгебра логики

Скрипач живёт рядом с врачом (4), т. е. врач может жить правее (дом 2) или левее (дом 3) скрипача.

1	2	3	4
скрипач?	врач?	врач?	скрипач?

Но врач живёт левее охотника (2), следовательно, скрипач не может жить в доме 4, т. к. в противном случае получится, что врач, живущий с ним рядом, живёт правее охотника, а это противоречит условию (2). Таким образом, скрипач живёт в доме 1, а врач — рядом с ним, в доме 2.

1	2	3	4
скрипач	врач		

Так как врач живёт левее охотника (2), а столяр — правее охотника (1), то охотнику достаётся дом 3, а столяру — дом 4.

1	2	3	4
скрипач	врач	охотник	столяр

Так как Семён не скрипач и не живёт рядом со скрипачом (5), то он может жить в доме 3 или в доме 4.

1	2	3	4
скрипач	врач	охотник	столяр
Не Семён	Не Семён	Семён?	Семён?

Так как Иван живёт рядом с охотником (6), то он может жить в доме 2 или 4.

1	2	3	4
скрипач	врач	охотник	столяр
Не Семён Не Иван	Не Семён Иван?	Семён? Не Иван	Семён? Иван?

Так как Василий живёт правее врача (7), то он может жить в доме 3 или 4.

1	2	3	4
скрипач	врач	охотник	столяр
Не Семён Не Иван Не Василий	Не Семён Иван? Не Василий	Семён? Не Иван Василий?	Семён? Иван? Василий?

Подводим итоги с учётом того, что Василий живёт через дом от Ивана (8): в доме 1 может жить только Геннадий, в доме 2 — Иван, в доме 4 — Василий, в доме 3 — Семён.

1	2	3	4
скрипач	врач	охотник	столяр
Геннадий	Иван	Семён	Василий

Как видите, далеко не самая сложная задача потребовала достаточно серьёзных рассуждений. Этот метод, как правило, применяется для решения простых задач.

22.2. Задачи о рыцарях и лжецах

Задачи о рыцарях и лжецах — это такой класс логических задач, в которых фигурируют персонажи:

- рыцарь — человек, всегда говорящий правду;
- лжец — человек, всегда говорящий ложь;
- обычный человек — человек, который в одних ситуациях может говорить правду, а в других — лгать.

Решение подобных задач сводится к перебору вариантов и исключению тех из них, которые приводят к противоречию.

Пример 2. Двое жителей острова *A* и *B* разговаривали между собой в саду. Проходивший мимо незнакомец спросил у *A*: «Вы рыцарь или лжец?». Тот ответил, но так неразборчиво, что незнакомец не смог ничего понять. Тогда незнакомец спросил у *B*: «Что сказал *A*?». «*A* сказал, что он лжец», — ответил *B*. Может ли незнакомец доверять ответу *B*? Мог ли *A* сказать, что он лжец?

Если A — рыцарь, то он скажет правду и сообщит, что он рыцарь.

Если A — лжец, то он скроет правду и сообщит, что он рыцарь.

Это значит, что B , утверждающий, что « A сказал, что он лжец» заведомо лжёт; он — лжец. Определить же, кем является A , в данной ситуации невозможно.

Пример 3. Рядом стоят два города: город Лжецов (L) и город Правдивых (P). В городе Лжецов живут лжецы, а в городе Правдивых — правдивые люди. Лжецы всегда лгут, а правдивые — всегда говорят правду. Лжецы и правдивые ходят друг к другу в гости.

Вы попали в один из городов, а в какой не знаете. Вам нужно у первого встречного, задав простой вопрос, узнать, в каком вы городе. Ответом на вопрос может быть только «Да» или «Нет».

Нужен простой вопрос, ответ на который точно известен вашему респонденту. Например: «Вы находитесь в своём городе?».

Надо задать вопрос и проанализировать варианты ответов с учетом того, кто их мог дать.

Самостоятельно разберитесь с решением задачи, рассмотрев блок-схему на рис. 4.12.

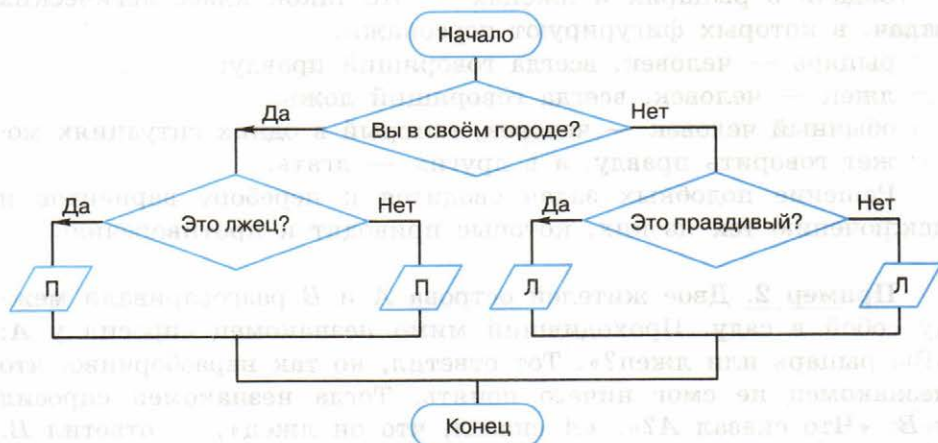


Рис. 4.12. Блок-схема для анализа ответов

Пример 4. Перед нами три человека: A , B и C . Один из них рыцарь, другой — лжец, третий — нормальный человек. При этом неизвестно, кто есть кто. Эти люди утверждают следующее:

- 1) A : я нормальный человек;
- 2) B : это правда;
- 3) C : я не нормальный человек.

Кто такие A , B и C ?

Для решения этой задачи следует рассмотреть все возможные варианты распределения ролей.

Начнём с A . Он может быть рыцарем (Р), лжецом (Л) или нормальным человеком (Н). Если A — рыцарь, то B может быть лжецом или нормальным человеком и т. д. Представим все варианты распределения ролей в таблице:

A	B	C
Р	Л	Н
Р	Н	Л
Л	Р	Н
Л	Н	Р
Н	Р	Л
Н	Л	Р

Проанализируем имеющиеся три утверждения, считая, что роли между A , B и C распределены в соответствии с первой строкой таблицы.

Итак, A утверждает, что он нормальный человек (1). Но, согласно первой строке таблицы, — он рыцарь, который не может так о себе сказать. Получено противоречие. Следовательно, первая строка не удовлетворяет условию задачи.

Самостоятельно проанализируйте оставшиеся строки таблицы и дайте ответ на вопрос, поставленный в задаче.

22.3. Задачи на сопоставление. Табличный метод

Многие логические задачи связаны с рассмотрением нескольких конечных множеств и связей между их элементами. Для решения таких задач зачастую прибегают к помощи таблиц или графов. От того, насколько удачно выбрана их структура, во многом зависит успешность решения задачи.



Пример 5. В летнем лагере в одной палатке жили Алёша, Боря, Витя и Гриша. Все они разного возраста, учатся в разных классах (с 7-го по 10-й) и занимаются в разных кружках: математическом, авиамodelьном, шахматном и фотокружке. Выяснилось, что фотограф старше Гриши, Алёша старше Вити, а шахматист старше Алёши. В воскресенье Алёша с фотографом играли в теннис, а Гриша в то же время проиграл авиамodelисту в городки.

Определим, кто в каком кружке занимается.

В этой задаче речь идёт о высказывательной форме (предикате) вида «Ученик x занимается в кружке y ». Требуется определить такие значения x и y , чтобы высказывательная форма превратилась в истинное высказывание.

Составим таблицу:

$x \backslash y$	Математика	Авиамоделирование	Шахматы	Фотография
Алёша				
Боря				
Витя				
Гриша				

Рассмотрим условия:

- 1) фотограф старше Гриши;
- 2) Алёша старше Вити, а шахматист старше Алёши;
- 3) в воскресенье Алёша с фотографом играли в теннис, а Гриша в то же время проиграл авиамodelисту в городки.

Можем сделать выводы: Гриша — не фотограф (1); шахматист — не Алёша и не Витя (2); Алёша — не фотограф и не авиамodelист, Гриша — не фотограф и не авиамodelист (3). Отметим это в таблице:

$x \backslash y$	Математика	Авиамоделирование	Шахматы	Фотография
Алёша		0	0	0
Боря				
Витя			0	
Гриша		0		0

Имеющейся информации достаточно для того, чтобы утверждать, что Алёша занимается математикой, а Гриша — шахматами:

$x \backslash y$	Математика	Авиамоделирование	Шахматы	Фотография
Алёша	1	0	0	0
Боря	0		0	
Витя	0		0	
Гриша	0	0	1	0

Из того, что Гриша — шахматист, и условий (1) и (2) следует, что мы можем расположить учеников по возрасту (в порядке возрастания): Витя — Алёша — шахматист Гриша — фотограф. Следовательно, Боря — фотограф. Этого достаточно, чтобы окончательно заполнить таблицу:

$x \backslash y$	Математика	Авиамоделирование	Шахматы	Фотография
Алёша	1	0	0	0
Боря	0	0	0	1
Витя	0	1	0	0
Гриша	0	0	1	0

Итак, Алёша занимается в математическом кружке, Боря — в фотокружке, Витя — в авиамodelьном кружке, Гриша — в шахматном кружке.

Самостоятельно сделайте вывод о том, кто из ребят в каком классе учится.



22.4. Использование таблиц истинности для решения логических задач

Аппарат алгебры логики позволяет применять к широкому классу логических задач универсальные методы, основанные на формализации условий задачи.

Одним из таких методов является построение таблицы истинности по условию задачи и её анализ. Для этого следует:

- 1) выделить из условия задачи элементарные (простые) высказывания и обозначить их буквами;
- 2) записать условие задачи на языке алгебры логики, соединив простые высказывания в составные с помощью логических операций;
- 3) построить таблицу истинности для полученных логических выражений;
- 4) выбрать решение — набор логических переменных (элементарных высказываний), при котором значения логических выражений соответствуют условиям задачи;
- 5) убедиться, что полученное решение удовлетворяет всем условиям задачи.

Пример 6. Три подразделения A , B , C торговой фирмы стремились получить по итогам года максимальную прибыль. Экономисты высказали следующие предположения:

- 1) если A получит максимальную прибыль, то максимальную прибыль получают B и C ;
- 2) A и C получают или не получают максимальную прибыль одновременно;
- 3) необходимым условием получения максимальной прибыли подразделением C является получение максимальной прибыли подразделением B .

По завершении года оказалось, что одно из трёх предположений ложно, а остальные два истинны.

Выясним, какие из названных подразделений получили максимальную прибыль.

Рассмотрим элементарные высказывания:

- A — « A получит максимальную прибыль»;
- B — « B получит максимальную прибыль»;
- C — « C получит максимальную прибыль».

Запишем на языке алгебры логики прогнозы, высказанные экономистами:

- 1) $F_1 = A \rightarrow B \& C$;
- 2) $F_2 = A \& C \vee \bar{A} \& \bar{C}$;
- 3) $F_3 = C \rightarrow B$.

Составим таблицу истинности для F_1 , F_2 , F_3 .

A	B	C	F_1	F_2	F_3
0	0	0	1	1	1
0	0	1	1	0	0
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Теперь вспомним, что из трёх прогнозов F_1 , F_2 , F_3 один оказался ложным, а два других — истинными. Эта ситуация соответствует четвёртой строке таблицы.

Таким образом, максимальную прибыль получили подразделения B и C .

22.5. Решение логических задач путём упрощения логических выражений

Следующий формальный способ решения логических задач состоит в том, чтобы:

- 1) выделить из условия задачи элементарные (простые) высказывания и обозначить их буквами;
- 2) записать условие задачи на языке алгебры логики, соединив простые высказывания в составные с помощью логических операций;
- 3) составить единое логическое выражение, учитывающее все требования задачи;
- 4) используя законы алгебры логики, упростить полученное выражение и вычислить его значение;
- 5) выбрать решение — набор логических переменных (элементарных высказываний), при котором построенное логическое выражение является истинным;
- 6) убедиться, что полученное решение удовлетворяет всем условиям задачи.

Пример 7. На вопрос, кто из трёх учащихся изучал логику, был получен ответ: «Если изучал первый, то изучал и второй, но неверно, что если изучал третий, то изучал и второй». Кто из учащихся изучал логику?

Обозначим через A , B , C простые высказывания:

- A = «Первый ученик изучал логику»;
- B = «Второй ученик изучал логику»;
- C = «Третий ученик изучал логику».

Из условия задачи следует истинность высказывания:

$$(A \rightarrow B) \& \overline{(C \rightarrow B)}.$$

Упростим получившееся высказывание:

$$\begin{aligned} (A \rightarrow B) \& \overline{(C \rightarrow B)} &= (\overline{A} \vee B) \& \overline{(\overline{C} \vee B)} = \\ &= (\overline{A} \vee B) \& C \& \overline{B} = \overline{A} \& C \& \overline{B} \vee B \& C \& \overline{B} = \\ &= \overline{A} \& C \& \overline{B} \end{aligned}$$

Получившееся высказывание будет истинным только в случае, если C — истина, а A и B — ложь. А это значит, что логику изучал только третий ученик, а первый и второй не изучали.

САМОЕ ГЛАВНОЕ

Исходными данными в логических задачах являются высказывания. При этом высказывания и взаимосвязи между ними бывают так сложны, что разобраться в них без использования специальных методов бывает достаточно трудно.

Основная идея метода рассуждений состоит в том, чтобы последовательно анализировать всю информацию, имеющуюся в задаче, и делать на этой основе выводы.

Многие логические задачи связаны с рассмотрением нескольких конечных множеств и связей между их элементами. Для решения таких задач зачастую прибегают к помощи таблиц или графов. От того, насколько удачно выбрана их структура, во многом зависит успешность решения задачи.

Аппарат алгебры логики позволяет применять к широкому классу логических задач универсальные методы, основанные на формализации условий задачи. К ним относятся методы: 1) построения таблицы истинности по условию задачи и её анализ; 2) составления и упрощения логического выражения.

Вопросы и задания



1. Вы встретили 10 островитян, стоящих по кругу. Каждый из них произнёс фразу: «Следующие 4 человека, стоящие после меня по часовой стрелке, лжецы». Сколько среди них лжецов?
2. Однажды некий путешественник гостил на острове рыцарей и лжецов. Там ему встретились два местных жителя. Путешественник спросил одного из них: «Кто-нибудь из вас рыцарь?» Его вопрос не остался без ответа, и он узнал то, что хотел. Кем был островитянин, к которому путешественник обратился с вопросом, — рыцарем или лжецом? Кем был другой островитянин?
3. В старинном индийском храме восседали три богини: Правда, Ложь и Мудрость. Правда говорит только правду, Ложь всегда лжёт, а Мудрость может сказать правду или солгать. Паломник, посетивший храм, спросил у богини слева: «Кто сидит рядом с тобой?» «Правда», — ответила та. Тогда он спросил у средней: «Кто ты?» «Мудрость», — отвечала она. Наконец он спросил у той, что справа: «Кто твоя соседка?» «Ложь», — ответила богиня. И после этого паломник точно знал, кто есть кто. Определите, на каком месте сидит каждая из богинь.
4. В симфонический оркестр приняли на работу трёх музыкантов — Борисова, Сергеева и Васечкина, умеющих играть на скрипке, флейте, альте, кларнете, гобое и трубе. Каждый из музыкантов владеет двумя инструментами.
Известно, что:
 - 1) Сергеев — самый высокий;
 - 2) играющий на скрипке меньше ростом играющего на флейте;
 - 3) играющие на скрипке и флейте и Борисов любят пиццу;
 - 4) когда между альтистом и трубачом возникает ссора, Сергеев мирит их;
 - 5) Борисов не умеет играть ни на трубе, ни на гобое.Выясните, на каких инструментах играет каждый из музыкантов.
5. В педагогическом институте Аркадьева, Бабанова, Корсакова, Дашков, Ильин и Флёров преподают экономическую географию, английский язык, немецкий язык, историю, французский язык, математику.

Известно, что:

- 1) преподаватель немецкого языка и преподаватель математики в студенческие годы занимались художественной гимнастикой;
- 2) Ильин старше Флёрова, но стаж работы у него меньше, чем у преподавателя экономической географии;
- 3) будучи студентками, Аркадьева и Бабанова учились вместе в одном университете. Все остальные окончили педагогический институт;
- 4) Флёров — сын преподавателя французского языка, но студентом у него не был;
- 5) преподаватель французского языка — самый старший из всех по возрасту и у него самый большой стаж работы. Он работает в педагогическом институте с тех пор, как окончил его. Преподаватели математики и истории — его бывшие студенты;
- 6) Аркадьева старше преподавателя немецкого языка.
Кто какой предмет преподаёт?

6. На вопрос «Кто из девушек собирается прийти на день рождения к Саше?» был получен уклончивый ответ: «Если Марина придёт на день рождения, то Надя тоже придёт, а Таня не придёт. Если Надя придёт, то Таня придёт в том и только в том случае, если не придёт Марина». Можно ли по этой информации точно установить, кто из девушек придёт к Саше, а кто нет?

7. В бюро переводов приняли на работу троих сотрудников: Диму, Сашу и Юру. Каждый из них знает ровно два иностранных языка из следующего набора: немецкий, японский, шведский, китайский, французский и греческий.

Известно, что:

- 1) ни Дима, ни Юра не знают японского;
- 2) переводчик с шведского старше переводчика с немецкого;
- 3) переводчик с китайского, переводчик с французского и Саша родом из одного города;
- 4) переводчик с греческого, переводчик с немецкого и Юра учились втроём в одном институте;
- 5) Дима — самый молодой из всех троих, и он не знает греческого;
- 6) Юра знает два европейских языка.

Укажите имена переводчика с шведского языка и переводчика с китайского языка.

8. Ребята знали, что у четырёх подруг — Маши, Кати, Вали и Наташи — дни рождения приходятся на разное время года, но не могли точно вспомнить, у кого на какое. Попытка вспомнить закончилась следующими утверждениями:

- 1) у Вали день рождения зимой, а у Кати — летом;
- 2) у Кати день рождения осенью, а у Маши — весной;
- 3) весной празднует день рождения Наташа, а Валя отмечает его летом.

Позже выяснилось, что в каждом утверждении только одно из двух высказываний истинно. В какое время года день рождения у каждой из девушек?

9. В санатории на берегу моря отдыхают отец O , мать M , сын S и две дочери D_1 и D_2 . До завтрака члены семьи часто купаются в море, причём известно, что если отец утром отправляется купаться, то с ним обязательно идут мать и сын; если сын идёт купаться, то его сестра D_1 отправляется вместе с ним; вторая дочь D_2 купается тогда и только тогда, когда купается мать; каждое утро купается по крайней мере один из родителей. Если в воскресенье утром купалась в море лишь одна из дочерей, то кто из членов семьи в это утро ходил на море?
10. В нарушении правил обмена валюты подозреваются четыре работника банка — Антипов (A), Борисов (B), Цветков (C) и Дмитриев (D). Известно, что:
- 1) если A нарушил, то и B нарушил правила обмена валюты;
 - 2) если B нарушил, то и C нарушил или A не нарушал;
 - 3) если D не нарушал, то A нарушил, а C не нарушал;
 - 4) если D нарушил, то и A нарушил.

Кто из подозреваемых нарушил правила обмена валюты?


Дополнительные материалы к главе смотрите в авторской мастерской.



Глава 5

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ И ОБРАБОТКИ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ

Традиционно понятие «технология» трактуется как совокупность методов и инструментов для достижения желаемого результата в некоторой области человеческой деятельности.

 Информационные технологии (ИТ) — это совокупность методов, производственных процессов, программно-технических и лингвистических средств, объединённых с целью сбора, обработки, хранения, распространения, отображения и использования информации, представленной в цифровой форме.

Синонимом этого понятия в русском языке выступает понятие «информационно-коммуникационные технологии» (ИКТ). Используемые англоязычные аббревиатуры — ИТ, ИСТ.

В отличие от технологий «материальных» и исходным материалом, и результатом применения информационных технологий всегда являются данные. Информационная технология — это процедура автоматизированного преобразования данных, формирования на их основе новых данных. Целью разработки и применения всех информационных технологий является максимальная автоматизация тех информационных процессов, которые ранее требовали ручного человеческого труда, зачастую рутинного, предполагавшего значительные временные затраты.

Информационные технологии как отдельная отрасль деятельности получили наибольшее развитие с появлением и распространением компьютеров — универсальных автоматических средств обработки данных.

Существует большое количество оснований для классификации информационных технологий. Их делят на индивидуальные

и коллективные, локальные и сетевые, технологии управления данными и процессами, защиты информации, разработки программного обеспечения и т. д.

В курсе информатики основной школы информационные технологии были представлены (классифицированы) по видам обрабатываемой информации. Вы знакомы и, скорее всего, используете в учебной деятельности, технологии обработки текста и графической информации, мультимедийные технологии, электронные таблицы, базы данных. Знание базовых принципов обработки информации, владение наиболее распространёнными технологиями — необходимый навык для любого современного человека.

§ 23

Текстовые документы

23.1. Виды текстовых документов

В различных словарях можно найти следующие толкования понятия «текст»:

- 1) упорядоченный набор слов, предназначенный для того, чтобы выразить некий смысл;
- 2) всякая записанная речь (литературное произведение, сочинение, документ и т. п., а также часть, отрывок из них);
- 3) последовательность языковых и иных знаков, образующая единое целое, служащее объектом изучения.

С позиции информатики, текст — это последовательность знаков некоторого алфавита.

Вам известно, что в памяти компьютера тексты представляются в двоичном коде: 1) за каждым символом алфавита закрепляется определённый двоичный код; 2) в двоичном коде представляется и информация о типе и размере используемого шрифта, положении строк, полей, отступов и прочая дополнительная информация.

Практически в любой профессиональной деятельности работник сталкивается с необходимостью подготовки текстовых документов

различного назначения и объёма: от заявления о приёме на работу до составления отчёта по результатам проделанной работы.

Можно выделить следующие виды текстовых документов:

- художественный текст;
- научный текст;
- деловой документ;
- рекламный документ;
- личный документ.

Для каждой из перечисленных разновидностей текстовых документов существует определённый набор правил, которых следует придерживаться при работе над ними. Личное письмо отличается по стилистике от официального документа, а художественное произведение — от научного текста. Различаются также словари наиболее употребляемых слов и терминов для перечисленных разновидностей документов.

Дайте характеристику каждому из видов текстовых документов — художественному тексту, научному тексту, деловому документу, рекламному документу, личному документу.

23.2. Виды программного обеспечения для обработки текстовой информации

Существует множество программных продуктов, предназначенных для работы с текстовой информацией. Представим классификацию этой разновидности прикладного программного обеспечения по его назначению.

Текстовые редакторы — это программы, которые помогают подготовить текст простой структуры, но не обладают необходимыми средствами оформления его для печати. Типичный пример — редактор Блокнот (в ОС Windows).

Текстовые процессоры — более сложные программные комплексы, позволяющие выполнить оформление текста, точно задать его расположение, включить в него графические материалы. Примеры — Microsoft Word, OpenOffice Writer.

Специальные программные средства для подготовки научных текстов, содержащих математические, химические или другие формулы, сложные схемы и специфические обозначения, используемые в научных, учебных и технических публикациях и документах. При подготовке научных, технических и учебных текстов часто используется свободно доступная система подготовки публикаций $T_E X$.

Издательские системы — комплексы программных средств, позволяющих выполнить весь цикл допечатной подготовки издания: импорт или набор текста, его оформление и расположение на листах, вставку иллюстраций и сложных объектов, и в итоге — вывод издания на печать. Примерами таких программ могут быть пакеты Adobe InDesign, Scribus, QuarkXPress. Процесс и результат создания страниц издания называют вёрсткой, а точную копию самого издания — оригинал-макетом. Использование издательских систем позволило значительно сократить срок подготовки печатных изданий, снизить трудоёмкость этого процесса, значительно расширить творческие возможности дизайнеров печатных изданий.

Электронные переводчики и словари предназначены для автоматического перевода текстов с одного языка на другой, проверки правописания текстов на разных языках. Особым видом словарей являются тезаурусы — словари, в которых слова связываются на основе каких-либо лексических отношений (например, слова, являющиеся синонимами, антонимами и т. п.). Примеры — PROMT, ABVYU Lingvo.

Системы оптического распознавания текстов (например, ABVYU FineReader) предназначены для преобразования отсканированного графического изображения текстового документа в текстовый формат.

Кроме того, программы для работы с текстовой информацией интегрированы в системы программирования, а также являются частью HTML-редакторов, предназначенных для создания веб-страниц.

23.3. Создание текстовых документов на компьютере

При подготовке текстовых документов на компьютере используются три основные группы операций: ввод, редактирование, форматирование.

Операции ввода позволяют сформировать содержимое и первоначальный вид текстового документа и сохранить его в памяти компьютера. Ввод может осуществляться не только набором с помощью клавиатуры, но и путём сканирования бумажного оригинала и последующего перевода документа из графического формата в текстовый (распознавания).



Напомним основные правила ввода текстовых документов с помощью клавиатуры.

- Между словами ставится только один пробел.
- Переход на новую строку в процессе набора текста происходит автоматически, не требуя ввода специального символа.
- Окончание абзаца маркируется нажатием клавиши Enter, позволяющей перейти на новую строку — первую строку нового абзаца.
- Перед знаками препинания (такими, как ; : . , ! ?) пробел не ставится. Перед тире вводится пробел. После любого знака препинания вводится один пробел или символ конца абзаца.
- Знак «дефис» в словах вводится без пробелов.
- После открывающих и перед закрывающими скобками ({ } () []) и кавычками пробел не вводится.
- Для ввода римских цифр используются прописные латинские буквы I, V, X, L, C, D, M.
- Знак «неразрывный (нерастяжимый) пробел» препятствует символам, между которыми он поставлен, располагаться на разных строчках, и сохраняется фиксированным при любом выравнивании абзаца (не может увеличиваться, в отличие от обычного пробела). Этот знак очень удобно применять при вводе дат (которые не принято располагать на двух строчках), фамилий с инициалами и т. п. Например: А. С. Пушкин. Ставится знак «неразрывный пробел» с помощью одновременного нажатия комбинации клавиш Ctrl + Shift + пробел.

При вводе и редактировании текста полезно включать режим отображения скрытых символов — символов, которые вводятся пользователем при наборе текста, но при печати не выводятся на бумагу, а на экране отображаются только при включении соответствующего режима (табл. 5.1). Режим отображения скрытых символов даёт возможность лучше понять структуру документа.

Для автоматизации ввода в современных текстовых процессорах существуют инструменты **Автозамена** и **Автотекст**.

Бывает, что при вводе текста с клавиатуры пользователь допускает опечатки: вместо нужной клавиши нажимает соседнюю, пропускает букву, меняет две буквы местами. Такие опечатки исправляются автоматически инструментом **Автозамена**, имеющим встроенный словарь наиболее типичных опечаток и ошибочных написаний.

Таблица 5.1

Примеры скрытых символов

Название символа	Клавиша (комбинация клавиш)	Изображение
Пробел	Пробел	•
Конец абзаца	Enter	¶
Неразрывный пробел	Ctrl + Shift + пробел	◦
Разрыв строки	Shift + Enter	↵
Разрыв страницы	Ctrl + Enter Разрыв страницы

Для быстрого ввода стандартных фраз по нескольким первым буквам можно использовать инструмент **Автотекст**. Он автоматически предлагает вставить короткую фразу из списка элементов автотекста, как только будут набраны несколько первых букв этой фразы.

Операции редактирования (правки) позволяют изменить уже существующий электронный документ путём добавления или удаления фрагментов, перестановки частей документа, слияния нескольких файлов, разбиения единого документа на несколько более мелких и т. д. (рис. 5.1)



Рис. 5.1. Основные операции редактирования документов

На протяжении многих веков для внесения изменений в текст нужно было заново переписывать его. Основное преимущество компьютерной технологии создания текстовых документов заключается именно в удобстве его редактирования. Возможность быстро исправлять ошибки является одной из основных причин повсеместного перевода подготовки текстовой информации с бумажной на компьютерную основу.

Ввод и редактирование при работе над текстом часто выполняются параллельно. При вводе и редактировании формируется содержание текстового документа.

Совокупность значений свойств объекта называют форматом объекта, а изменение этих значений — форматированием объекта.

Операции форматирования позволяют точно определить, как будет выглядеть текст на экране монитора или на бумаге после печати на принтере. Операции форматирования могут применяться как к отдельным объектам текстового документа (табл. 5.2), так и ко всему документу в целом. В первом случае говорят о прямом форматировании, во втором — о стилевом.

Таблица 5.2

Объекты текстового документа и их свойства

Объект	Свойства объекта
Символ	Шрифт, размер, цвет, начертание, видоизменение и др.
Слово	Межзнаковый интервал и др.
Абзац	Отступ от границ левого и правого полей, отступ первой строки, выравнивание, междустрочный интервал, расстояние между абзацами и др.
Страница	Размеры страницы, её ориентация, размеры полей, колонтитулы и др.
Список	Вид списка (маркированный, нумерованный, многоуровневый) и др.
Таблица	Количество строк и столбцов, их высота и ширина, границы, заливка и др.
Графическое изображение	Тип, размер, положение, обтекание текстом и др.

Такие действия по оформлению документа, как выравнивание абзацев, установка абзацных отступов и интервалов между абзацами, строками в абзацах и символами в словах и т. п., выполняются специальными средствами текстовых процессоров, а не вставкой пробелов и пустых строк.

Для облегчения анализа и последующего преобразования текста очень важно соблюдать основные правила его ввода, редактирования и форматирования.

В современных текстовых процессорах есть специальные инструменты, обеспечивающие автоматическую нумерацию страниц, таблиц и рисунков.

При работе с большими текстами, как правило, применяют **стилевое форматирование**. Смысл этой операции заключается в том, что структурным элементам, несущим одну и ту же функциональную нагрузку (например, заголовкам одного уровня, основному тексту, примерам и т. д.), назначается определённый стиль форматирования — набор параметров форматирования (шрифт, его начертание и размер, абзацные отступы, междустрочный интервал и др.).

Стиль — это имеющий имя набор значений свойств объектов каждого типа, входящих в текстовый документ.

В заключение приведём основные правила оформления текстов:

- основной текст документа желательно оформлять в одном формате, другой формат использовать для выделения заголовков, отдельных смысловых фрагментов;
- количество разных цветов и шрифтов в документе не должно превышать трёх;
- размер символов и междустрочный интервал следует подбирать такими, чтобы текст читался легко, без дополнительной нагрузки для глаз;
- цветовая гамма должна отвечать назначению документа — поздравительная открытка может иметь яркую, насыщенную окраску, а простое письмо стоит оформлять в более спокойных тонах;
- однотипную информацию целесообразно представлять в списках и таблицах;

- графические изображения в документе (рисунки, диаграммы, схемы) должны дополнять содержание текста, разъяснять или иллюстрировать его отдельные моменты; графические изображения следует оформлять в едином стиле;
- на всех страницах документа рекомендуется делать одинаковый фон и поля, если иное не требуется содержанием документа.

23.4. Средства автоматизации процесса создания документов

Мы рассмотрели основные операции ввода, редактирования и форматирования документов. Многие из них в той или иной мере направлены на автоматизацию процесса создания текстовых документов. Ещё больше возможностей в этом направлении обеспечивает использование шаблонов, макросов и средств, обеспечивающих работу со структурными компонентами документа.

Многие типовые документы должны иметь стандартный вид, который определяет, что и где размещается в создаваемом тексте, например: кому адресован документ, от кого он, дата создания документа и другие реквизиты.

Требования к оформлению, структуре и содержанию многих документов устанавливаются стандартами. Все они находятся в открытом доступе в Интернете.

Найдите в Интернете и познакомьтесь со следующими стандартами:

- ГОСТ Р 6.30-2003 «Унифицированные системы документации. Унифицированная система организационно-распорядительной документации. Требования к оформлению документов»;
- ГОСТ 2.105-95 «ЕСКД. Общие требования к текстовым документам»;
- ГОСТ Р 7.0.5-2008 «СИБИД. Библиографическая ссылка. Общие требования и правила составления»;
- ГОСТ 7.32-2001 «СИБИД. Отчёт о научно-исследовательской работе. Структура и правила оформления».

Какие из них имеют статус государственных, а какие — межгосударственных? Как это отражено в названии стандартов?

Что означают аббревиатуры ГОСТ, ЕСКД, СИБИД?

Какие из этих стандартов могут быть полезны в вашей учебной деятельности?

В текстовых процессорах есть шаблоны для создания документов разного типа.

Шаблон — это отформатированный определённым образом документ-заготовка, который хранится в отдельном файле и используется в качестве основы для создания новых документов определённого типа.

Пользователю достаточно ввести свою информацию в отдельные блоки шаблона, и она автоматически приобретёт заранее заданное оформление. В недалёком будущем каждому из вас для поиска подходящей работы придётся составить и разослать резюме. Подготовить его лучше всего с использованием соответствующего шаблона (рис. 5.2).

▶ [Введите свое имя]

[Введите свой адрес]
Телефон: [Введите свой номер телефона]
Электронная почта: [Введите свой адрес электронной почты]
Веб-узел: [Введите название своего веб-узла]

Цели
[Введите цели]

Образование
[Введите ученую степень] ([Введите дату завершения])
▶ [Введите список поощрений]

Опыт работы
[Введите должность] ([Введите дату начала] – [Введите дату окончания])
[Введите название организации] ([Введите адрес организации])
[Введите служебные обязанности]

Умения и навыки
▶ [Введите список умений и навыков]

Рис. 5.2. Шаблон резюме

В текстовом процессоре Microsoft Word все шаблоны распределены на три группы:

- 1) установленные — шаблоны документов определённых типов (писем, факсов, отчётов и др.), которые инсталлированы на компьютере в составе пакета Microsoft Office;
- 2) Microsoft Office Online — шаблоны документов разнообразных типов (поздравительных открыток, визиток, бюллетеней, сертификатов, грамот, приглашений, заявлений, календарей и др.), которые расположены на веб-сайте Microsoft Office Online;
- 3) шаблоны пользователя — шаблоны, которые созданы пользователем.

При запуске программы Microsoft Word автоматически открывается шаблон **Новый документ** (файл Normal.dotm). При этом по умолчанию устанавливается формат (значения свойств) основных объектов документа — страницы, абзаца, символа, а также задаётся стилевое форматирование заголовков, списков, таблиц и др.

Запустите имеющийся в вашем распоряжении текстовый процессор. Исследуйте формат основных объектов документа в шаблоне **Новый документ**, выяснив для них значения свойств, приведённых ниже:

Страница	Абзац	Символ
<ul style="list-style-type: none"> • Ориентация • Размер листа • Верхнее поле • Нижнее поле • Левое поле • Правое поле 	<ul style="list-style-type: none"> • Междустрочный интервал • Интервал после абзаца • Выравнивание • Отступы 	<ul style="list-style-type: none"> • Шрифт основного текста • Размер • Цвет • Интервал

В процессе работы над документом в программе Microsoft Word часто приходится выполнять задания по некоторому алгоритму, состоящему из определённой последовательности действий. Например, подчеркнуть слова, написанные латинскими буквами, отформатировать какое-то слово во всём документе определённым образом и др.

Макрос — это последовательность команд, сгруппированных в одну макрокоманду, для автоматического выполнения определённого задания.

Основное назначение макроса состоит в том, чтобы освободить пользователя от многократного повторения однообразных действий во время обработки текстового документа, выполнить за него рутинную работу. Макрос создаётся один раз, сохраняется в шаблоне или документе и может многократно выполняться для автоматизации и ускорения обработки текстового документа.

Многостраничные документы (рефераты, брошюры, книги и т. п.) принято делить на структурные части — главы, параграфы, пункты и т. п., создавая таким образом иерархическую структуру документа. Рассмотрим в качестве примера структуру этого учебника. На верхнем (нулевом) уровне иерархии находится название документа («Информатика. 10 класс»); на первом уровне — названия глав, второй уровень составляют названия параграфов, третий — названия пунктов в параграфах, дальше размещается основной текст учебника.

Структура документа — это иерархическая схема размещения составных частей документа.

Использование в текстовом процессоре Microsoft Word специальных стилей с именами Заголовок 1, Заголовок 2 и т. д. даёт возможность автоматизировать создание иерархической структуры документа. В текстовом процессоре Microsoft Word для просмотра структуры документа используется режим просмотра **Структура**. В нём удобно редактировать иерархическую схему документа, изменяя с помощью специальных инструментов уровень текстовых фрагментов и последовательность их размещения.

Современные текстовые процессоры позволяют автоматически создавать оглавления документов, в которых к заголовкам разделов разных уровней применено стилевое форматирование. С помощью специальной команды пользователь указывает, заголовки каких уровней следует включить в оглавление, и абзацы указанных стилей автоматически выбираются из текста документа и помещаются с указанием номеров страниц, с которых они были взяты, в новый раздел «Оглавление».

Оглавление документа — это перечень названий структурных частей документа, упорядоченных в соответствии с его иерархической схемой, с указанием соответствующих номеров страниц.

23.5. Совместная работа над документом



Под совместной (коллективной) работой над документом принято понимать участие нескольких человек в создании одного текстового документа, при котором у каждого из них есть возможность отслеживать все изменения, сделанные в документе другими разработчиками, а также осуществлять возврат к одной из предыдущих версий документа.

На протяжении долгого времени процесс совместной работы над документом был устроен так: кто-то создавал текст на заданную тему, распечатывал его и отдавал другому человеку. Этот человек дописывал «от руки» свои вставки и комментарии и возвращал черновики на доработку. Такие итерации могли происходить неоднократно, в результате работа шла медленно, к тому же отдельные ценные мысли и идеи могли быть утеряны безвозвратно.

Современные инструменты создания текстовой информации предоставляют принципиально иные возможности для совместной работы над документом, поддерживая следующие варианты её организации.

1. Документ, над которым ведётся работа, можно сделать составным, объединяющим несколько других документов. Каждый разработчик создаёт и редактирует свою часть составного документа независимо от других, при этом в процессе работы он может просматривать текущую версию общего документа. Такой вариант организации совместной работы основывается на возможности создания мастер-документа, к которому могут прикрепляться другие документы.
2. Каждая из частей составного документа может редактироваться несколькими людьми. Для этого в современных текстовых процессорах предусмотрена возможность отслеживать и протоколировать сделанные изменения, делать примечания — пометки на полях.

Для отслеживания собственных исправлений, а также чтобы другие соавторы могли вносить изменения в документ, текстовый процессор предлагает использовать маркеры исправлений. Маркеры исправлений помогают увидеть, какие изменения были внесены в документ по сравнению с его последней версией. Для изображения исправлений используется специаль-

ный формат, например подчёркивание. С помощью маркеров исправлений можно сохранить запись о каждом сделанном исправлении и в дальнейшем либо принять его, либо отказаться. Исправление помечается полным именем его автора, датой и временем создания.

При совместной работе над документом важно, чтобы в настройках текстового процессора были указаны корректные данные о пользователе, т. к. именно они останутся в редактируемом документе.

Примечания — это обозначенные инициалами и пронумерованные комментарии, которые записываются и отображаются в специальном окне примечаний и не затрагивают текст документа. Перед тем как вставить примечание, имеет смысл выделить фрагмент текста, который следует прокомментировать. В этом случае при просмотре примечания текст, к которому оно относится, будет подсвечен.

Имеются возможности сравнения двух версий документов или объединения всех исправлений и примечаний в один документ, для просмотра их всех сразу. Сделанные в тексте изменения перед сохранением окончательной версии документа можно принять или отклонить.

Оба рассмотренных варианта организации совместной работы поддерживаются и в OpenOffice Writer, и в Microsoft Word.

3. Можно редактировать документ непосредственно в сети, также отслеживая все изменения, сделанные другими пользователями. Этот вариант поддерживается Google Docs — сетевым приложением, доступным через веб-браузер. Поскольку редактируемый документ сохраняется на сервере, он доступен всем пользователям, между которыми он разделяется. Каждый раз при сохранении документа сохраняется его новая версия и информация о том, кто этот документ редактировал. Команда просмотра изменений открывает список сделанных изменений с указанием, кто и когда их внёс. Различные версии документов можно сравнить между собой.
4. Можно использовать и промежуточный вариант, когда документ редактируется локально на компьютере, а храниться в сети и там же отслеживаются все изменения.

23.6. Оформление реферата как пример автоматизации процесса создания документов

Старшеклассники, студенты, курсанты в процессе своей учебной деятельности готовят рефераты по различным предметам.

Реферат — это самостоятельная исследовательская работа, в которой автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, делает собственные выводы.

Выбрав тему реферата, необходимо определить цель работы, составить план (поставить задачи, определить порядок и сроки выполнения задач), найти и изучить материалы различных информационных источников, собрать и обработать информацию, сделать выводы, оценить полученные результаты.

Содержание реферата должно быть логичным, изложенным ясным языком. Основные положения реферата желательно подкреплять цитатами и ссылками на информационные источники.

Есть определённые требования и к оформлению реферата.

Реферат должен быть выполнен на одной стороне листов белой бумаги формата А4 (210 × 297 мм).

Размеры полей страницы (не менее):

- правое — 30 мм (для замечаний преподавателя);
- верхнее, нижнее, левое по 20 мм.

Отступ первой строки: 8–12 мм, одинаковый по всему тексту.

Интервал междустрочный: полусторонний.

Выравнивание абзаца: по ширине.

Гарнитура шрифта основного текста — Times New Roman или аналогичная.

Кегль (размер): 12–14 пунктов.

Цвет шрифта: чёрный.

Заголовки разделов и подразделов следует печатать на отдельной строке с прописной буквы без точки в конце, не подчёркивая. Если заголовок состоит из нескольких предложений, их разделяют точкой. Выравнивание по центру или по левому краю. Интервал: перед заголовком — 12 пунктов, после — 6 пунктов.

Страницы следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту (титульный лист и оглавление).

ние включают в общую нумерацию). На титульном листе номер не проставляют.

В верхней части титульного листа пишется, в каком образовательном учреждении выполняется работа, далее буквами увеличенного кегля указывается тип («Реферат») и тема работы, ниже в правой половине листа — информация о тех, кто выполнил и кто проверит работу. В центре нижней части титульного листа пишется название населённого пункта и год выполнения работы.

Из курса русского языка вам известно, что цитата — это приведённое полностью или частично высказывание из авторского текста (научной, художественной, публицистической и др. литературы или доклада). Цитаты оформляются как прямая речь или как продолжение предложения.

Правовой статус цитирования определяется Гражданским кодексом РФ, согласно которому цитирование «допускается без согласия автора или иного правообладателя и без выплаты вознаграждения, но с обязательным указанием имени автора, произведение которого используется, и источника заимствования» (статья 1274 части 4 Гражданского кодекса РФ).

Допустим, при работе над рефератом вы взяли (заимствовали) информацию из источника «Андреева, Е. В. Математические основы информатики. Элективный курс: учебное пособие / Е. В. Андреева, Л. Л. Босова, И. Н. Фалина. — М.: БИНОМ. Лаборатория знаний, 2005». Необходимо правильно оформить ссылку на этот источник и внести его в список литературы.

Правила оформления ссылок регулируются ГОСТ Р 7.0.5-2008 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления».

Правила оформления библиографических сведений в списке использованной литературы должны отвечать требованиям ГОСТ 7.1-2003 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления».

Для оформления ссылки в текстовом процессоре Microsoft Word можно:

- 1) установить курсор после заимствованного текста и выполнить команду **Ссылки** → **Вставить ссылку** → **Добавить новый источник...**;
- 2) заполнить поля диалогового окна **Создать источник** (рис. 5.3);

Рис. 5.3. Окно Создать источник

3) затем щёлкнуть на кнопке **ОК** и после заимствованного фрагмента текста появится ссылка на источник в виде: «(Е. В. Андреева, 2005)».

Для оформления списка литературы в текстовом процессоре Microsoft Word можно:

- 1) выбрать стиль отображения списка литературы (библиографического списка), выполнив команду **Ссылки** → **Стиль:** → **ГОСТ – сортировка по именам 2003**;
- 2) выполнить команду **Ссылки** → **Список литературы** → **Вставить список литературы**.

23.7. Другие возможности автоматизации обработки текстовой информации

Компьютер помогает не только автоматизировать процесс создания текстовых документов, но и решить множество других задач, связанных с обработкой текстовой информации. Вот некоторые из них:

- **поиск текста** в общем массиве по заданным нечётким признакам;
- **рубрицирование текста** — разбиение поступающего потока текстов на тематические подпотоки в соответствии с заранее заданными рубриками;
- **реферирование текста** — подготовка его сокращённой версии;
- **перевод текста** с одного языка на другой;
- **анализ текста** на предмет выявления заимствований.

Область информатики, решающая эти и другие задачи, связанные с обработкой информации на естественном языке, называется компьютерной лингвистикой.

Рассмотрим более подробно задачу поиска текста в общем массиве. Существует несколько подходов к её решению.

Первый подход опирается на поиск фрагмента текста, соответствующего некоторому образцу. Таким способом в большом текстовом массиве можно находить упоминания тех или иных слов, адреса, номера телефонов и другие элементы. Основное достоинство такого подхода — возможность применять его к массиву текста без предварительной обработки (например, сразу при посимвольном получении текста). Применение рассматриваемого способа бывает затруднено, если текст хранится в разных местах.

Второй подход предусматривает предварительную обработку текста с целью получения его преобразованного, сокращённого вида (индекса). Получив запрос, поисковая система выделяет список слов и составляет список документов, в которых они содержатся. При этом рассчитывается релевантность — мера соответствия документа запросу, зависящая от наличия искомых слов, близости их друг к другу и других параметров. Документы с высокой релевантностью помещаются в начало списка, с низкой — в конец.

Одно из интересных применений автоматического анализа текстов — выявление заимствований.

Антиплагиат (antiplagiat.ru) — российский интернет-проект, программно-аппаратный комплекс для проверки текстовых документов на наличие заимствований из страниц сети Интернет и других источников. Проект доступен для всех пользователей.

САМОЕ ГЛАВНОЕ

Информационные технологии (ИТ) — это совокупность методов, производственных процессов, программно-технических и лингвистических средств, объединённых с целью сбора, обработки, хранения, распространения, отображения и использования информации, представленной в цифровой форме.

С позиции информатики, текст — это последовательность знаков некоторого алфавита. Существует множество программных продуктов, предназначенных для работы с текстовой информацией.

При подготовке текстовых документов на компьютере используются три основные группы операций: ввод, редактирование, форматирование.



Операции ввода позволяют сформировать содержимое и первоначальный вид текстового документа и сохранить его в памяти компьютера.

Операции редактирования (правки) позволяют изменить уже существующий электронный документ путём добавления, удаления, перестановки фрагментов, слияния нескольких файлов, разбиения единого документа на несколько более мелких и т. д.

Операции форматирования позволяют точно определить, как будет выглядеть текст на экране монитора или на бумаге после печати на принтере. Операции форматирования могут применяться как к отдельным объектам текстового документа, так и ко всему документу в целом.

Автоматизация процесса создания текстовых документов обеспечивается за счёт возможности работы с фрагментами, проверки правописания, стилевого форматирования, а также использования шаблонов, макросов и средств, обеспечивающих работу со структурными компонентами документа.

Компьютер помогает автоматизировать не только процесс создания текстовых документов, но и решить множество других задач, связанных с обработкой текстовой информации, а именно:

- поиск текста в общем массиве по заданным нечётким признакам;
- рубрицирование текста — разбиение поступающего потока текстов на тематические подпотоки в соответствии с заранее заданными рубриками;
- реферирование текста — подготовка его сокращённой версии;
- перевод текста с одного языка на другой;
- анализ текста на предмет выявления заимствований.



Вопросы и задания

1. Что, по вашему мнению, стимулировало развитие технологий обработки текстовой информации? Как это связано с понятием «безбумажные технологии»? Наступит ли такое время, когда все технологии обработки текста будут безбумажными?
2. На страницах Википедии найдите информацию об истории создания программы Microsoft Word. Составьте хронологическую таблицу.
3. Каковы основные правила ввода текста с клавиатуры?
4. В чём отличие использования в тексте символа «тире» от символа «дефис», «пробела» от «неразрывного пробела»?



5. Найдите информацию о разных способах ввода текста в память компьютера (рукописном, голосовом, сканировании). Какое оборудование и программное обеспечение для этого используют?
6. Ваш одноклассник жалуется на то, что при вводе нового текста старый текст частично исчезает. Что вы ему посоветуете?
7. Перед вами текст на русском языке, содержащий отдельные английские слова:

Растровые изображения можно получить, сканируя рисунки или фотографии, фотографируя объекты цифровым фотоаппаратом, создавая рисунки с использованием графического планшета или разнообразных растровых графических редакторов (Paint, GIMP, Adobe Photoshop).

- Почему правильно написанные слова оказались подчеркнуты так, как если бы в них были ошибки?
8. Почему иногда в тексте, написанном на одном языке, некоторые слова подчёркиваются волнистыми линиями, даже если в них нет ошибок? Как исправить эту ситуацию?
 9. Как в документе осуществить автоматическую замену одного фрагмента текста на другой? Как, используя операцию автоматической замены, быстро объединить все абзацы текста?
 10. Один ученик хотел заменить в тексте слово «мы» на слово «я», но получил текст, в котором появилось много ошибок. Объясните эту ситуацию. Как можно её исправить?
 11. Списки каких типов вам известны? В каких ситуациях следует применять каждый из них?
 12. В списке учеников вашего класса было пропущено несколько фамилий. Каким образом их можно вставить в текст?
 13. Фамилии учеников вашего класса были введены в произвольном порядке. Каким образом можно расположить фамилии в алфавитном порядке?
 14. В таблице выделено несколько строк. Выясните, что произойдёт при нажатии клавиши Delete; клавиши BackSpace.
 15. Графические документы каких видов можно вставлять в текстовый документ?

16. В чём заключается процесс форматирования текста?
17. Автор некоторого документа ввёл заголовок текста, в котором символы отделил пробелами для увеличения расстояния между ними. Правильно ли он сделал? Что вы ему посоветуете?
18. Некоторое слово нужно обязательно разместить в данной строке текста, а оно там не умещается. Что вы предпримете в данной ситуации?
19. Как можно ввести в текст математические выражения, которые содержат верхний и нижний индексы, обычные дроби, буквы греческого алфавита?
20. Подумайте, какие преимущества обеспечивает стилевое форматирование по сравнению с прямым форматированием.
21. Как можно ускорить процесс форматирования текста, в котором часто встречаются фрагменты с одинаковым форматом?
22. Найдите информацию о правилах оформления деловых документов (заявления, справки, докладной записки и др.). Выясните, какие существуют требования к их оформлению.
23. Каковы общие правила стилового оформления документов?
24. Выясните, что понимается под корпоративным (фирменным) стилем оформления документов. Кто и для чего его разрабатывает? Каковы его основные черты? Приведите примеры документов, оформленных в корпоративном стиле.
25. Исследуйте шаблоны документов текстового процессора, имеющегося в вашем распоряжении. Выясните их количество, тематику документов, которые могут быть созданы на их основе.
26. Что такое макрос? Для чего он используется?
27. Что такое структура документа? Приведите примеры структурированных документов.
28. Создайте кластер «Инструменты автоматизации создания текстовых документов» с учётом таких групп операций, как ввод, редактирование и форматирование.
29. Перечислите основные классы задач, связанных с автоматизацией обработки текстовой информации.
30. Охарактеризуйте наиболее распространённые форматы текстовых документов — TXT, RTF, PDF, DOC, DOCX, ODT.



§ 24

Объекты компьютерной графики

24.1. Компьютерная графика и её виды

Компьютерная графика — широкое понятие, обозначающее:

- 1) область деятельности, в которой компьютеры используются как инструменты создания и обработки графических объектов;
- 2) разные виды графических объектов, созданных или обработанных с помощью компьютера.

По способу создания можно выделить следующие классы объектов компьютерной графики (рис. 5.4): двумерные изображения (растровые, векторные, фрактальные), трёхмерные изображения, анимацию.



Рис. 5.4. Объекты компьютерной графики

Растровое графическое изображение состоит из отдельных маленьких прямоугольников — пикселей. Размеры пикселей растрового изображения настолько малы, что при его просмотре в обычном масштабе они неразличимы. Зернистая структура растрового графического изображения становится заметной при увеличении масштаба его просмотра (рис. 5.5).

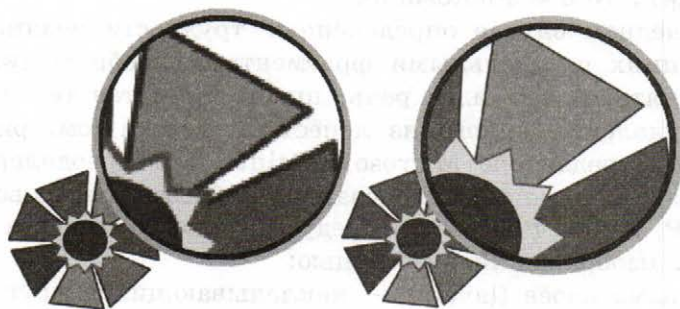


Рис. 5.5. Растровое и векторное изображения в обычном и увеличенном масштабе

Пиксель — наименьший элемент растрового изображения. Он может принимать любой цвет из палитры, содержащей миллионы цветов. Точность цветопередачи — основное достоинство растровых графических изображений. При сохранении растрового изображения в памяти компьютера сохраняется информация о цвете каждого входящего в него пикселя. Качество растрового изображения возрастает с увеличением количества пикселей в изображении и количества цветов в палитре. Но при этом возрастает и информационный объём всего изображения.

Растровая графика — универсальное средство для формирования и обработки любых плоских изображений. С помощью цветов и оттенков отдельных точек на плоском изображении могут быть показаны и пространственные (объёмные) сцены. В полиграфических и электронных изданиях растровые изображения используются в тех случаях, когда нужно качественно и чётко передать в изображении оттенки цветов и плавные переходы от одного цвета к другому.

Серьёзным недостатком растровой графики является существенное падение качества изображения в результате его масштабирования и различных геометрических преобразований. Это связано с тем, что при уменьшении растрового изображения несколько соседних пикселей преобразуются в один, что ведёт к потере чёткости мелких деталей изображения. При увеличении растрового изображения в него добавляются новые пиксели, причём соседние пиксели принимают одинаковый цвет и возникает ступенчатый эффект.

Растровые изображения можно получить, сканируя рисунки или фотографии, фотографируя объекты цифровым фотоаппаратом, создавая рисунки с использованием графического планшета или разнообразных растровых графических редакторов (Microsoft Paint, GIMP, Adobe Photoshop).

В последнем случае определённые трудности возникают при манипуляциях с отдельными фрагментами изображения. Чтобы лучше понять, о чём идёт речь, просто представьте свои действия при удалении одного из лепестков цветка (см. рис. 5.5) в графическом редакторе Microsoft Paint. Для преодоления этого недостатка более мощные программы работы с растровой графикой (GIMP, Adobe Photoshop) предусматривают средства создания составных изображений с помощью:

- 1) механизма слоёв (layers) — накладывающихся друг на друга плоскостей, в каждой из которых используется только часть точек;

- 2) механизма фильтров — преобразующих цвета пикселей с учётом некоторых параметров;
- 3) управления цветовыми каналами;
- 4) управления способом взаимодействия отдельных слоёв.

Векторное изображение — это изображение, построенное из геометрических примитивов (объектов): отрезков прямых, дуг, окружностей, эллипсов, многоугольников и кривых Безье. Примитив не нужно рисовать — выбрав на панели инструментов пиктограмму с его изображением или названием, вы просто задаёте необходимые параметры, по которым компьютер сам выполняет необходимые построения.

Объекты векторного изображения накладываются друг на друга, образуя независимые слои. Каждый слой векторного изображения содержит свой объект. При преобразовании векторного объекта исходное изображение удаляется, а вместо него строится новое — по тем же алгоритмам, но с учётом изменённых данных. Это позволяет без потерь качества масштабировать, поворачивать и трансформировать векторные изображения, оставляя при этом толщину линий неизменной (см. рис. 5.5).

Кривые Безье были разработаны в 60-х годах XX века независимо друг от друга Пьером Безье из автомобилестроительной компании «Рено» и Полем де Кастельжо из компании «Ситроен», где применялись для проектирования кузовов автомобилей. Математический аппарат кривых Безье основан на многочленах Бернштейна, описанных Сергеем Натановичем Бернштейном в 1912 году.

В компьютерной графике в основном применяются кривые Безье второго и третьего порядка (рис. 5.6).

Кривая Безье второго порядка описывается уравнением:

$$B(t) = (1 - t)^2 P_0 + 2t(1 - t)P_1 + t^2 P_2, \\ t \in [0, 1].$$

Здесь:

- P_0 — начало кривой;
- P_1 — опорная точка;
- P_2 — конец кривой;
- прямая P_0P_1 — касательная к кривой в точке P_0 ;
- прямая P_1P_2 — касательная к кривой в точке P_2 .

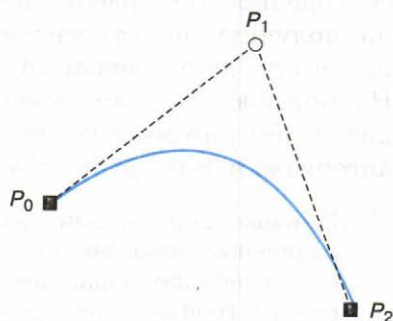


Рис. 5.6. Кривая Безье второго порядка

Вместе с тем не всякое изображение можно представить как совокупность простых геометрических фигур. Векторные графические изображения создают с помощью специальных программ (CorelDRAW, Inkscape) и широко используют в картографии, мультипликации, инженерной графике, при создании логотипов, схем, диаграмм — там, где важны чёткость контуров и возможность увеличения масштаба изображения без потери качества.

Ещё одним видом компьютерной графики является фрактальная графика. Термин «фрактал» (от лат. *fractus* — дроблённый) употребляется для обозначения объектов, обладающих свойством самоподобия, когда целое (в точности или приближённо) имеет ту же форму, что одна или более его частей (рис. 5.7).

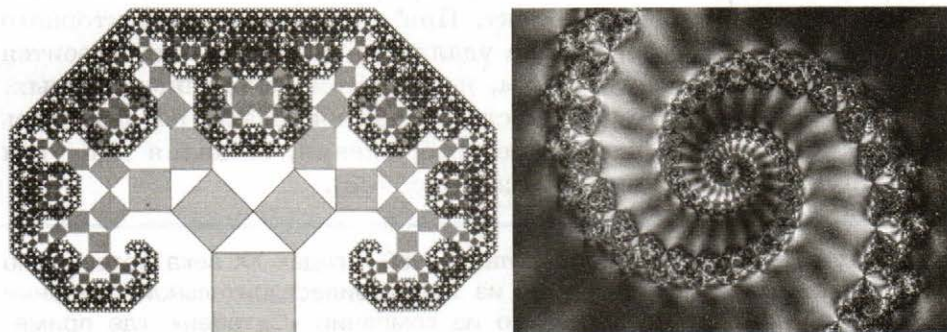


Рис. 5.7. Примеры фрактальных изображений

В основе фрактальной графики лежит очень простая идея: бесконечное по красоте и разнообразию множество фигур можно получить из относительно простых конструкций при помощи всего двух операций — копирования и масштабирования. На компьютере построение фрактального изображения происходит путём автоматической генерации изображений по некоторым алгоритмам или формулам, хранящимся в памяти компьютера.

В наши дни теория фракталов находит широкое применение в различных областях человеческой деятельности. Помимо фрактальной живописи, фракталы используются в теории информации для сжатия графических данных, в физике и географии для правдоподобного представления моделируемых объектов, в радиоэлектронике для качественного приёма сигнала, в экономике для описания кривых колебания курсов валют и т. д.

Много интересного о фракталах вы найдёте на сайте «Элементы большой науки» (elementy.ru/posters/fractals).

В последнее время всё большую популярность приобретает трёхмерная или 3D-графика (от англ. *three dimensions* — три измерения). В ней применяются технологии создания в виртуальном пространстве объёмных моделей, которые максимально приближены к реальным объектам. Трёхмерная графика широко используется в инженерном проектировании, компьютерном моделировании физических объектов и процессов, в мультипликации, кинематографии и компьютерных играх.

Рассмотрим процесс создания трёхмерного изображения с помощью векторной графики.

Сначала в пределах некоторого пространства координат (на сцене) размещаются отдельные объекты, составленные из геометрических объёмных тел.

Далее, на этапе каркасной аппроксимации производится разбивка всех плавных криволинейных поверхностей на треугольники — минимальные плоские фигуры. В дальнейшем поверхности обрабатываются именно как наборы треугольников, заданных координатами своих вершин.

Стадия геометрических построений поверхностей включает заполнение поверхностей, создание перспективы, учёт влияния источников света и т. д.

На заключительном этапе рендеринга происходит раскраска поверхностей, а в более сложных случаях создания профессиональных изображений — ещё и учёт свойств поверхностей при отражении и поглощении света и влияния оптической плотности окружающей среды.

В настоящее время одним из наиболее популярных 3D-редакторов является Blender — свободно распространяемый пакет для создания трёхмерной компьютерной графики. Его можно найти на сайте разработчиков (www.blender.org), скачать и установить на свой компьютер. Освоить Blender помогут видеоуроки (www.youtube.com).

Анимация (от англ. *animation* — одушевление) — это «оживление» изображения. При анимации несколько рисунков (кадров) сменяют друг друга через заданные промежутки времени. Если кадры сменяют друг друга чаще, чем 24 раза в секунду, человеческий глаз воспринимает это как непрерывное движение. В настоящее время широкое распространение получила компьютерная анимация.

www



www

Компьютерная анимация — последовательный показ заранее подготовленных графических файлов, а также компьютерная имитация движения с помощью изменения формы объектов или показа последовательных изображений с фазами движения.

Рассмотрим основные виды компьютерной анимации.

Анимация по ключевым кадрам наиболее близка к традиционной рисованной мультипликации. Прорисовку и расстановку ключевых кадров по временной шкале производит художник, а промежуточные кадры рассчитывает специальная программа.

Запись движения. Движения актёров в специальных костюмах с датчиками записываются камерами и анализируются специальным программным обеспечением. Итоговые данные о перемещении суставов и конечностей актёров применяют к трёхмерным скелетам виртуальных персонажей, чем добиваются высокого уровня достоверности изображения движения последних.

Процедурная анимация автоматически генерируется компьютером в режиме реального времени в соответствии с установленными правилами. Представляет собой симуляцию физического взаимодействия твёрдых тел; имитацию движения систем частиц, жидкостей и газов; расчёт движения персонажа под внешним воздействием и многое другое. Процедурная анимация часто используется в компьютерных играх.

Программируемая анимация. Движения анимируемых объектов программируются, например, на языке JavaScript.

В 1968 г. группой учёных под руководством советского и российского математика Н. Н. Константинова было создано одно из первых компьютерных анимированных изображений. Электронно-вычислительная машина БЭСМ-4 по специальной программе рисовала кадры мультфильма «Кошечка» (www.etudes.ru/ru/etudes/cat).

24.2. Форматы графических файлов

Формат графического файла — это способ представления графических данных на внешнем носителе.

Различают растровые и векторные форматы графических файлов. Среди них, в свою очередь, выделяют собственные (оригинальные) форматы графических приложений и универсальные

графические форматы, которые «понимаются» всеми приложениями, работающими с растровой (векторной) графикой.

Для всех растровых форматов общей проблемой является большой объём данных. Для её решения применяют различные методы сжатия данных.

Под сжатием (упаковкой, компрессией) данных понимается применение алгоритмов преобразования данных, производимое с целью уменьшения их объёма. Обратная процедура называется восстановлением данных (распаковкой, декомпрессией).

Различают две основные группы:

- методы сжатия без потерь, позволяющие восстановить исходные данные без каких-либо изменений;
- методы сжатия с потерями, восстанавливающие данные с отличиями, допустимыми с точки зрения дальнейшего их использования.

BMP (от англ. *Bit Map image* — битовая карта изображения). В этом формате можно хранить только однослойные растры. На каждый пиксель в разных типах файлов может приходиться разное количество бит (глубина цвета): 1, 2, 4, 8, 16, 24, 32, 48 и 64. Изображения в этом универсальном формате в большинстве случаев хранятся в файлах попиксельно, без сжатия, потому размеры таких файлов достаточно большие.

Сколько бит может приходиться на глубину цвета в файлах с расширением bmp, созданных в графическом редакторе Paint?

Имя файла:	Безымянный.bmp
Тип файла:	Монохромный рисунок (*.bmp;*.dib)
	Монохромный рисунок (*.bmp;*.dib)
	16-цветный рисунок (*.bmp;*.dib)
	256-цветный рисунок (*.bmp;*.dib)
	24-разрядный рисунок (*.bmp;*.dib)
	JPEG (*.jpg;*.jpeg;*.jpe;*.jif)

Для глубины цвета 4 и 8 бит формат BMP допускает RLE-кодирование, которое может уменьшить размер файлов.

Алгоритм RLE (от англ. *Run Length Encoding* — кодирование длин серий) — один из простейших методов сжатия изображений. Его основной идеей является поиск цепочек одинаковых элементов и замена их на пары «число повторений — значение», что в определённых случаях существенно уменьшает избыточность данных.

Например, состоящая из 39 символов строка

AAAAAABVCCCCCDEEEEEEAAAAAAAAAAAAAAAAAA

будет закодирована следующим образом: 6A2B7C1D6E17A. Полученная последовательность состоит из 13 символов. Данные оказались сжаты в $39/13 = 3$ раза.

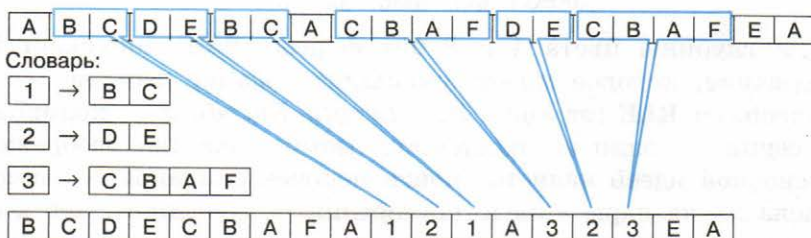
Алгоритм в первую очередь рассчитан на изображения с большими областями одного цвета (деловая графика, схемы, рисунки и т. п.). При его использовании в других ситуациях (например, при сохранении цветных фотографий) вместо уменьшения размера файла может происходить его увеличение.

Возьмём строку, в которой нет большого количества цепочек одинаковых элементов и достаточно часто встречаются символы, идущие без повторов: ABCDVCABCDDDDFFFFFF. После сжатия методом RLE она будет выглядеть так: 1A1B1C1D1B1C1A1B1C3D6F. Исходная строка состоит из 18 символов, а результат сжатия — из 22. Размер данных увеличился в $22/18 \approx 1,22$ раза.

Рассмотрим внимательно исходную строку: среди её первых девяти элементов нет цепочек из идущих подряд одинаковых символов (ABCDVCABC), затем записаны 3 символа «D» и наконец — 6 символов «F». Если для записи длины цепочки идущих подряд одинаковых символов использовать положительные числа, а для записи количества следующих друг за другом различных символов использовать отрицательные числа, то можно записать: -9ABCDVCABC3D6F.

Исходная строка состоит из 18 символов, а сжатая — из 15. Размер данных уменьшился в $18/15 = 1,2$ раза.

Для сжатия сложных изображений зачастую применяется алгоритм словарей или алгоритм LZW — Лемпеля–Зива–Велча, названный так по именам его разработчиков Авраама Лемпеля, Якоба Зива и Терри Велча. В его основе лежит идея замены наиболее часто встречающихся последовательностей в исходном потоке данных ссылками на «образцы», хранящиеся в специально создаваемой таблице (словаре).



Этот способ активно применяется при сжатии самых разных данных, в том числе и графических. Такой способ сжатия применяется в графических форматах GIF, PNG, TIFF. Рассмотрим эти форматы подробнее.

GIF (от англ. *Graphics Interchange Format* — формат обмена графикой) обладает высокой степенью сжатия, дающей возможность уменьшить размеры файлов в несколько раз. Этот формат предназначен для хранения статичных и анимированных изображений, которые содержат до 256 цветов (например, рисованные иллюстрации). GIF-анимация представляет собой последовательность из нескольких статичных кадров, а также информацию о том, сколько времени каждый кадр должен быть показан на экране. В циклических анимациях вслед за последним кадром начинается воспроизведение первого кадра и т. д. Долгое время GIF был одним из наиболее распространённых форматов в Интернете. Стандартное расширение имён файлов данного типа — gif.

PNG (от англ. *Portable Network Graphic* — портативная сетевая графика) — универсальный формат графических файлов, специально спроектированный для замены устаревшего и более простого формата GIF. Формат PNG имеет высокую степень сжатия данных без потерь и предназначен, прежде всего, для применения в сетевых приложениях. Он даёт возможность использовать гораздо больше цветов, чем формат GIF, поэтому может использоваться и для редактирования графики. Стандартное расширение имён файлов данного типа — png.

TIFF (от англ. *Tagged Image File Format* — формат файла размеченного изображения). Большая глубина цвета в файлах этого формата позволяет хранить изображение с высоким качеством. Поэтому формат широко используется при сканировании изображений, отправке факсов, распознавании текста, в полиграфии, широко поддерживается графическими приложениями. Но, как и файлы формата bmp, файлы этого формата имеют большие размеры. Стандартное расширение имён файлов — tif или tiff.

Как известно, в фотоизображениях практически не встречаются точно повторяющиеся последовательности точек. Поэтому рассмотренные выше форматы и используемые в них алгоритмы сжатия малоприспособны для хранения такого рода изображений.

Для решения этой проблемы группой специалистов был разработан специальный формат и способ сжатия, получивший название **JPEG** (от англ. *Joint Photographic Expert Group* — объединённая группа экспертов в области фотографии).

JPEG Алгоритм JPEG подразумевает сжатие с потерей качества, основанное на особенностях человеческого зрения (небольшие изменения цвета слабо распознаются глазом человека — это позволяет удалять часть информации о цвете). Его достоинство в том, что «силу» сжатия можно указать изначально и таким образом найти компромисс между качеством и объёмом изображения. Алгоритм JPEG — один из базовых алгоритмов сжатия изображений. Его широкое распространение позволило резко расширить возможности и сферу применения цифровых методов обработки изображения. Данный формат целесообразно использовать для хранения многоцветных изображений с плавными переходами между цветами, где потеря качества малозаметна. Для файлов этого формата стандартное расширение — `jpg` или `jpeg`.

В векторной графике почти каждый векторный графический редактор использует свой собственный формат хранения изображений.

WMF (от англ. *Windows MetaFile* — метафайл Windows) — универсальный формат для программ, которые работают в ОС Windows. Используется для хранения коллекции графических изображений Microsoft Clip Gallery. Возможные расширения имён файлов — `wmf`, `emf`, `wmz`, `emz`.

CGM (от англ. *Computer Graphic Metafile* — метафайл компьютерной графики) — используется для представления графических объектов, преимущественно, в конструкторских, авиационных и других технических областях. Стандартное расширение имён файлов — `sgm`.

SVG (от англ. *Scalable Vector Graphics* — масштабируемая векторная графика) — универсальный формат, который даёт возможность с высоким качеством хранить в файле текст, графическое изображение и анимацию. Широкое применение получил в инженерной графике и при разработке веб-сайтов. Стандартное расширение имён файлов — `svg`.

CDR (от англ. *CorelDraw files* — файлы CorelDraw) — собственный формат файлов векторного графического редактора CorelDraw. Изображение в файле может иметь несколько страниц, даёт возможность хранить не только векторную графику, но и текст, и растровые изображения. Файлы данного формата могут иметь расширение имени `cdr` или `cdt`.

AI (от англ. *Adobe Illustrator files* — файлы Adobe Illustrator) — собственный формат файлов редактора векторной графики Adobe Illustrator.

Совместимость форматов векторной графики очень низкая. Сложность преобразования данных из одного векторного формата в другой заключается в использовании разными программами разных алгоритмов построения графических примитивов.

Среди всего разнообразия форматов нет того идеального, который бы удовлетворял всем возможным требованиям пользователей. Поэтому графические редакторы предоставляют пользователю возможность самостоятельно выбирать формат файла, в котором будет сохранено изображение, в зависимости от целей работы с ним и последующего использования. Вот только некоторые рекомендации, которые могут вам пригодиться:

- записывать изображение в формате, использующем сжатие с потерями, следует только после окончания редактирования, т. к. многие манипуляции с изображением могут умножить дефекты сжатия;
- если вы собираетесь работать с графическим изображением только в одном графическом редакторе, целесообразно выбрать тот формат, который редактор предлагает по умолчанию. Если же последующая обработка данных предполагается в разных программах, стоит использовать один из универсальных форматов;
- для файлов, которые передаются по сети Интернет, очень важен небольшой размер, поскольку от этого зависит время передачи данных. Основной недостаток цифровых растровых изображений — их большой размер. Поэтому растровые фотографии и рисунки сохраняются в сжатом виде в различных графических форматах. Выбор формата зависит от типа изображения и способа его использования. Для размещения полноцветных фотографий в Интернете оптимальным является формат jpeg, а для чертежей, символьных и текстовых структур больше подходят форматы png или gif.

24.3. Понятие разрешения

При работе с объектами компьютерной графики приходится иметь дело с разрешением экрана, разрешением изображения и разрешением печатающего устройства.

Разрешение — величина, определяющая количество точек (элементов растрового изображения) на единицу площади (или единицу длины).



Разрешением экрана монитора обычно называют размеры получаемого на экране изображения в пикселях: 1024×768 , 1280×1024 , 1920×1080 . Разрешение экрана монитора — это свойство компьютерной системы (зависит от монитора и видеокарты) и операционной системы (зависит от настроек ОС).

Разрешение экрана монитора в **ppi** (произносится как «пи-пи-ай», сокращение от англ. *pixels per inch* — пикселей на дюйм; 1 дюйм $\approx 2,54$ см) можно рассчитать как отношение размера диагонали экрана в пикселях к размеру диагонали в дюймах:

$$ppi = \frac{d_p}{d_i}, \text{ где:}$$

- d_p — размер диагонали экрана в пикселях;
- d_i — размер диагонали экрана в дюймах.

Рассчитать размер диагонали экрана в пикселях можно по теореме Пифагора:

$$d_p = \sqrt{w_p^2 + h_p^2}, \text{ где:}$$

- w_p — ширина разрешения экрана в пикселях;
- h_p — высота разрешения экрана в пикселях.

Например, для 21,5-дюймового экрана с разрешением 1920×1080 (здесь $w_p = 1920$, $h_p = 1080$ и $d_i = 21,5$), получим ≈ 102 ppi.

Для монитора с диагональю 17 дюймов при разрешении 1280×1024 получаем разрешение в ppi, равное 96; при разрешении 1024×768 — 75 ppi.

Разрешение экрана определяет размер изображения, которое может поместиться на экране целиком.

Когда вы создаёте изображение в графическом редакторе, как правило, в начале работы устанавливаете его размеры. Вы можете посмотреть размеры в пикселях любого растрового изображения, вызвав его контекстное меню (рис. 5.8).

В графическом редакторе также есть возможность посмотреть размеры обрабатываемого в нём изображения. Например, в графическом редакторе Gimp в окне **Смена размера изображения** можно увидеть размер изображения в пикселях, миллиметрах, дюймах или других единицах по выбору пользователя (рис. 5.9).

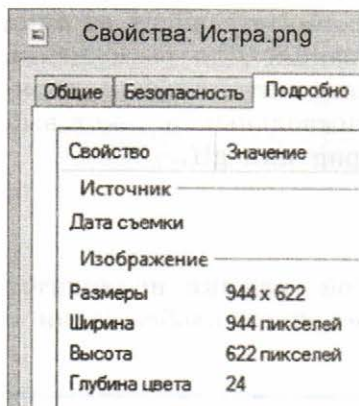


Рис. 5.8. Фрагмент окна **Свойства** графического изображения

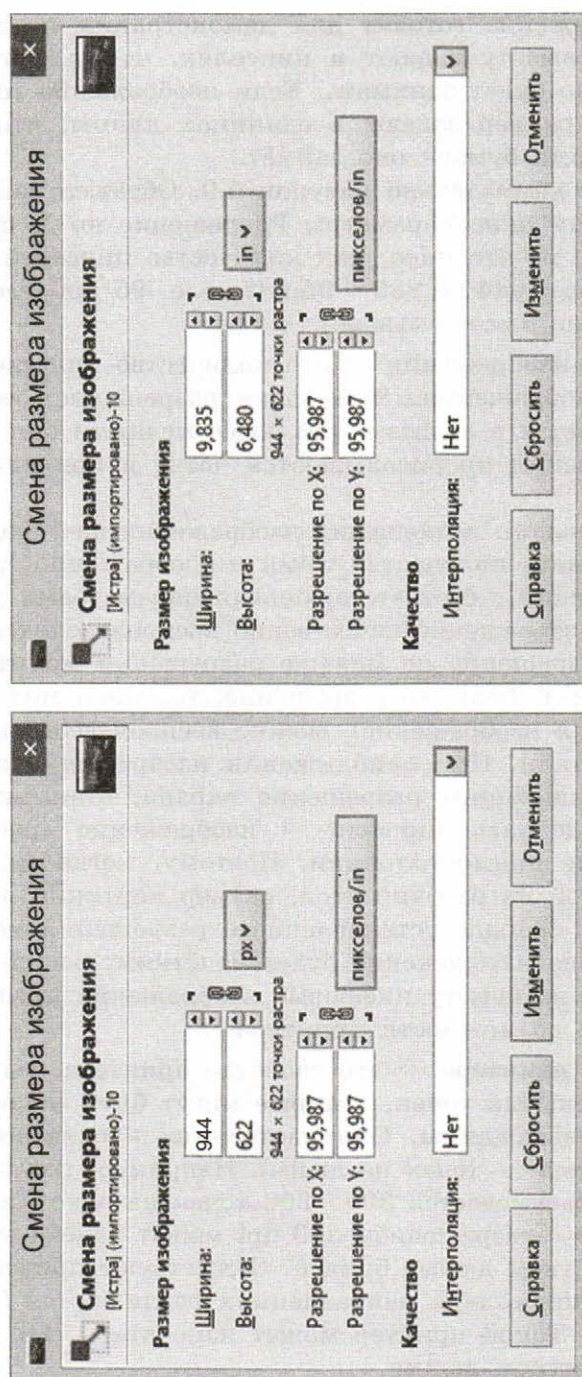


Рис. 5.9. Окно Смена размера изображения редактора Gimp

Если изображение готовят для демонстрации на экране, то его ширину и высоту задают в пикселях, чтобы знать, какую часть экрана оно будет занимать. Если изображение готовят для печати, то его размер задают в единицах длины, чтобы знать, какую часть листа бумаги оно займёт.

Рассмотрите внимательно рисунок 5.9. Обратите внимание на указанные в этом окне параметры **Разрешение по X** и **Разрешение по Y** — это не что иное, как количество пикселей на дюйм. В нашем примере: $944 : 9,835 = 95,987$, т. е. 96 ppi (вести речь о части пикселя не имеет смысла!).

Разрешение изображения — это количество пикселей на единицу длины изображения. Чем выше разрешение, тем больше пикселей уместится в дюйме и тем более мелкими они являются: детали изображения прорисовываются чётче и оригинал отображается точнее.

Для оптимального размещения изображения на экране необходимо согласовывать количество точек в изображении, пропорции сторон изображения с соответствующими параметрами устройства отображения. Изображение качественно воспроизводится на экране, если его разрешение не меньше разрешения монитора. Если изображение имеет большее разрешение, то, увеличивая его масштаб (приближая изображение), можно воспроизвести невидимые до этого его детали. При приближении изображения с разрешением, не превышающим разрешение экрана, становятся отчётливо видны отдельные пиксели — изображение «распадается» на разноцветные прямоугольники. Поэтому, когда мы подбираем красивые обои на рабочий стол, размер картинки в пикселях не должен быть меньше установленного разрешения монитора, в противном случае изображение будет нечётким, расплывчатым.

При выводе на бумагу цифровые изображения преобразуются под физические возможности принтера.

Разрешение принтера — это свойство принтера, выражающее количество отдельных точек, которые могут быть напечатаны на участке единичной длины. Оно измеряется в единицах dpi (от англ. *dots per inch* — точек на дюйм). Например, под разрешением 300 dpi подразумевается 300×300 точек на одном квадратном дюйме. Принтер с разрешением 300 dpi может напечатать 90 тыс. точек на квадратном дюйме бумаги. Существуют принтеры, у которых разрешения в двух направлениях различаются (например, 600×1200 dpi). Такой принтер может напечатать 720 тыс. точек на одном квадратном дюйме.

Разрешение принтера определяет размер изображения при заданном качестве или, наоборот, качество изображения при заданном размере.

Большинство форматов графических файлов позволяют хранить данные о желаемом масштабе при выводе на печать, т. е. о желаемом разрешении в dpi. Это исключительно справочная величина. Считается, что для печати изображения, которое будет рассматриваться с расстояния порядка 40–45 сантиметров, достаточно разрешения 300 dpi. Исходя из этого, можно рассчитать размер качественного отпечатка имеющегося цифрового изображения или выяснить, какого размера (в пикселях) должно быть изображение, чтобы можно было получить отпечаток требуемого размера.

Так, чтобы напечатать на бумаге изображение размером 10 × 15 см (4 × 6 дюймов) с разрешением в 300 dpi, размер исходного изображения должен быть не менее 1200 × 1800 пикселей.

Для печати изображения в чёрно-белой газете достаточно разрешения в 50–70 dpi, для полноцветной печати в книге требуется разрешение примерно 120–150 dpi, а для рекламного щита на улице — всего 6–10 dpi. Здесь всё определяют особенности восприятия изображения наблюдателем — чем больше картинка, тем с большего расстояния человек может её комфортно воспринимать. Но с большого расстояния не видны мелкие детали!

24.4. Цифровые фотографии

В наше время повсеместное распространение получила цифровая фотография. К её неоспоримым преимуществам можно отнести:

- простоту процесса съёмки и оперативность получения конечного результата;
- возможность проверки на месте (предпросмотр) качества результата;
- возможность съёмки очень большого количества кадров;
- простота создания панорам и спецэффектов;
- возможность обработки сделанных фотографий средствами компьютерной графики;
- простота встраивания фотографий во всевозможные цифровые материалы;
- возможность передачи фотографий по телекоммуникационным сетям или размещения в Интернете.

Цифровая фотография — растровое изображение, состоящее из множества цветных точек (пикселей). Её размер определяется

количеством пикселей в строке и количеством таких строк. Общее количество пикселей в фотографии рассчитывают как произведение количества строк на количество пикселей в строке и указывают в мегапикселях (миллионах пикселей).

Размер цифровой фотографии определяет качество (детализацию) изображения и размер соответствующего файла. Чем больше размер фотографии, тем выше её детализация и больше размер файла, и наоборот.

Если мы планируем разместить фотографию в Интернете, то её размер желательно уменьшить. В некоторых случаях размер фотографии требуется увеличить. При печати цифровых фотографий следует иметь в виду, что качество отпечатков желаемых размеров зависит от количества мегапикселей (табл. 5.3).

Таблица 5.3

Соотношение желательных размеров отпечатков и размеров цифровой фотографии

Желательный размер отпечатков (см)	Приемлемые размеры цифровой фотографии (пиксели)	Предпочтительные размеры цифровой фотографии (пиксели)
9 × 12	768 × 1024	1200 × 1600
10 × 15	768 × 1024	1200 × 1712
20 × 30	1200 × 1600	2272 × 3048

Если печатать маленькие фотографии в большом формате, то изображение получится нерезким, а на контрастных границах объектов будет заметна ступенчатость.

Познакомимся с простейшими приёмами обработки цифровых фотографий. Все приёмы, о которых будет идти речь, лучше всего сразу же испытать на практике. Для этого необходимо установить на компьютер свободно распространяемый растровый графический редактор GIMP («Гимп») и запастись хотя бы одной цифровой фотографией.

Запустите GIMP и откройте в нём какую-нибудь из имеющихся фотографий (**Файл** → **Открыть...**). Выполнив команду **Изображение** → **Размер изображения**, откройте диалоговое окно **Смена размера изображения** (см. рис. 5.9) и уточните размеры своей цифровой фотографии в сантиметрах.

Измените исходный размер изображения, указав желаемый размер, например, в поле ввода **Ширина**. Обратите внимание — пересчёт высоты выполнится автоматически.



Для выполнения операции изменения размера изображения графические редакторы применяют разнообразные алгоритмы определения цветов пикселей уменьшенного/увеличенного изображения.

Уменьшение размеров изображения характеризуется тем, что часть пикселей исходного изображения удаляется. Это обеспечивает уменьшение размера файла, но приводит к снижению качества исходного изображения. Почему в таком случае мы допускаем снижение качества изображения?

Увеличение размеров изображения выполняется добавлением к нему пикселей, однако качество изображения при этом не повышается. Как вы можете это объяснить?

Кадрирование — операция, позволяющая «вырезать» из исходного изображения его прямоугольную часть, «отсекая» всё, что находится снаружи выбранного прямоугольника.

Вызовите инструмент кадрирования (**Инструменты** → **Преобразование** → **Кадрирование**). Внутри изображения выделите нужную вам прямоугольную часть изображения, при этом всё, что не попало в область выделения, будет отображено с затемнением. Завершите операцию кадрирования, нажав клавишу **Enter**.

Коррекция — изменение характеристик изображения, позволяющее добиться нужного качества.

Обратите внимание на чёткость границ между элементами изображения. Иначе эту характеристику называют резкостью фотографического изображения. В арсенале GIMP есть несколько инструментов увеличения резкости (**Фильтры** → **Улучшение** ►). Выберите самый простой из них — **Повышение резкости**. В открывшемся диалоговом окне (рис. 5.10) можно устанавливать разные значения резкости и наблюдать за изменениями, происходящими с фотографией.

Зачастую сделанные нами фотографии оказываются блёклыми или недостаточно контрастными.



Рис. 5.10. Окно **Повышение резкости**

Контраст — степень тонового различия между областями изображения. **Яркость** — характеристика цвета, определяющая его интенсивность в изображении.

В большинстве современных графических редакторов у пользователя есть возможность посмотреть распределение количества пикселей изображения по их яркости.

Откройте диалоговое окно **Уровни** (Цвет → Уровни...) (рис. 5.11).

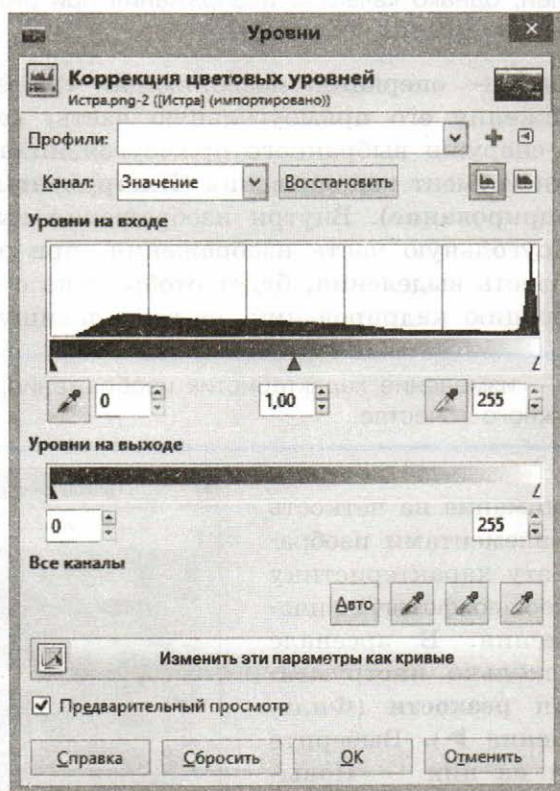


Рис. 5.11. Окно **Уровни**

Гистограмма в центральной части окна показывает распределение количества пикселей изображения по их яркости. По горизонтальной оси гистограммы откладываются значения яркости пикселя от нуля до 255. Высота вертикальных отрезков (вертикальная ось) определяется количеством пикселей, имеющих одинаковую яркость. Проследите за изменениями, происходящими с

фотографией, при перемещении чёрного, серого и белого движков (треугольников) под гистограммой.

Вы можете корректировать всё изображение, а можете работать с одним из трёх цветовых каналов, выбрав соответствующее значение из выпадающего списка **Канал**, размещённого над гистограммой. Параметры, изменяемые в процессе коррекции цветовых уровней, можно представить как кривые (рис. 5.12) — соответствующая кнопка размещена в нижней части окна **Уровни**.

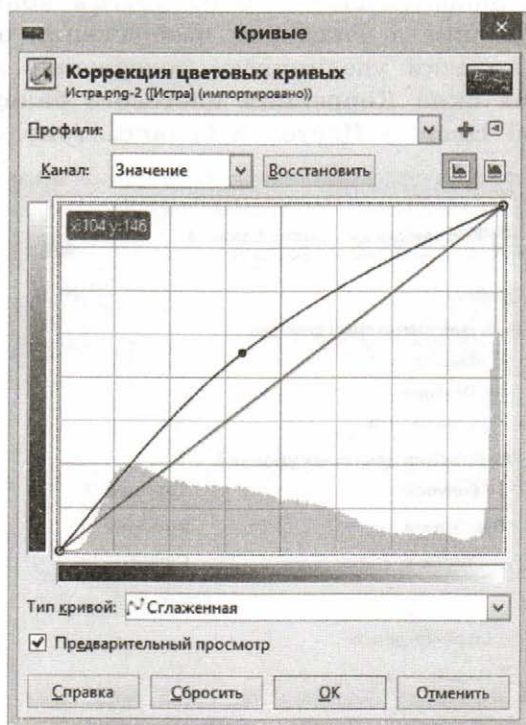


Рис. 5.12. Окно **Кривые**

С помощью кривых вы можете полностью контролировать яркость изображения, выгибая кривую в необходимых местах. Обратите внимание:

- если «тянуть» кривую влево и вверх, то результат будет ярче оригинала; если «тянуть» кривую вправо и вниз — темнее оригинала;
- если кривую построить «крутой», то контрастность изображения повысится, если пологой — контрастность снизится.

Более сложную обработку изображения можно произвести с помощью **коррекции цветов**. Специфика работы с цветом заключается в том, что изменение одного из них обязательно отражается на других. Поэтому в основе любой цветовой коррекции лежит не настройка отдельных цветов, а настройка цветового баланса. Она реализуется с помощью цветовой модели HSB, о которой рассказывалось в § 15. Цветовой круг, о котором там шла речь, наглядно представляет взаимосвязь цветов: каждый цвет находится напротив дополняющего его на другом конце диагонали. Уменьшая содержание в некотором изображении одного из цветов, мы автоматически увеличиваем содержание дополняющего его цвета. Вызов окна **Коррекция цветового баланса** осуществляется командой **Цвет → Цветовой баланс...** (рис. 5.13).

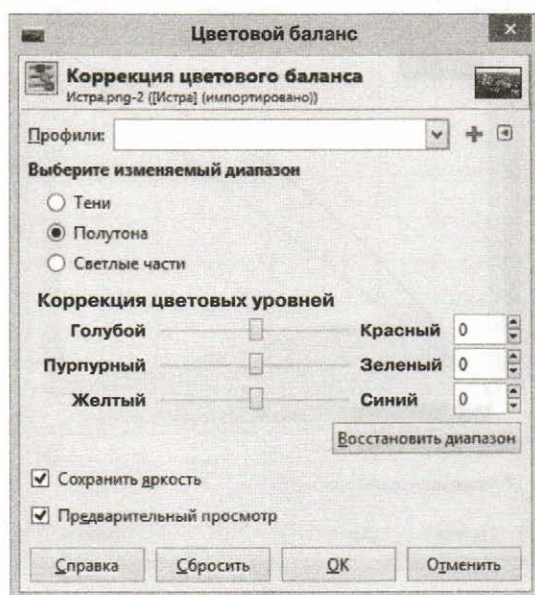


Рис. 5.13. Окно Цветовой баланс

Самостоятельно исследуйте многочисленные художественные фильтры, имеющиеся в графическом редакторе GIMP.

Мы рассмотрели только малую часть возможностей обработки цифровых фотографий современными графическими редакторами. Желающие могут продолжить дальнейшее их изучение самостоятельно, используя для этого многочисленные ресурсы, имеющиеся в сети Интернет.

Высокий уровень автоматизации основных операций по обработке цифровых фотографий позволяет сегодня каждому пользователю создавать эффектные работы, даже не имея специальной подготовки или опыта работы с соответствующим программным обеспечением. Это можно сделать с помощью онлайн-фоторедакторов. Один из них — Pho.to (**pho.to/ru**). Аналогичные программные продукты можно устанавливать на планшетные компьютеры и смартфоны.

САМОЕ ГЛАВНОЕ

Компьютерная графика — широкое понятие, обозначающее: 1) область деятельности, в которой компьютеры используются как инструменты создания и обработки графических объектов; 2) разные виды графических объектов, созданных или обработанных с помощью компьютера.

По способу создания можно выделить следующие классы объектов компьютерной графики: двумерные изображения (растровые, векторные, фрактальные), трёхмерные изображения, анимацию.

Графический формат — это способ записи графической информации. Графические форматы делятся на векторные и растровые. Большинство графических форматов реализуют сжатие данных (одни — с потерями, другие — без).

Физический размер изображения определяет размер рисунка по вертикали (высота) и горизонтали (ширина) и может измеряться как в пикселях, так и в единицах длины (миллиметрах, сантиметрах, дюймах). Он задаётся при создании изображения и хранится вместе с файлом. Если изображение готовят для демонстрации на экране, то его ширину и высоту задают в пикселях, чтобы знать, какую часть экрана оно занимает. Если изображение готовят для печати, то его размер задают в единицах длины, чтобы знать, какую часть листа бумаги оно займёт.

При работе с объектами компьютерной графики приходится иметь дело с разрешением экрана, разрешением изображения и разрешением печатающего устройства. Разрешением экрана монитора обычно называют размеры получаемого на экране изображения в пикселях: 1024×768 , 1280×1024 , 1920×1080 . Разрешение изображения — это количество пикселей на единицу длины изображения. Разрешение принтера — это свойство принтера, выражающее количество отдельных точек, которые могут быть напечатаны на участке единичной длины. Для печати изображе-

ния, которое будет рассматриваться с расстояния порядка 40–45 сантиметров, достаточно разрешения 300 dpi.

Неоспоримыми достоинствами цифровой фотографии являются: простота процесса съёмки и оперативность получения конечного результата; возможность проверки на месте (предпросмотр) качества результата; возможность съёмки очень большого количества кадров; простота создания панорам и спецэффектов; возможность обработки сделанных фотографий средствами компьютерной графики; простота встраивания фотографий во всевозможные цифровые материалы; возможность передачи фотографий по телекоммуникационным сетям или размещения в Интернете.

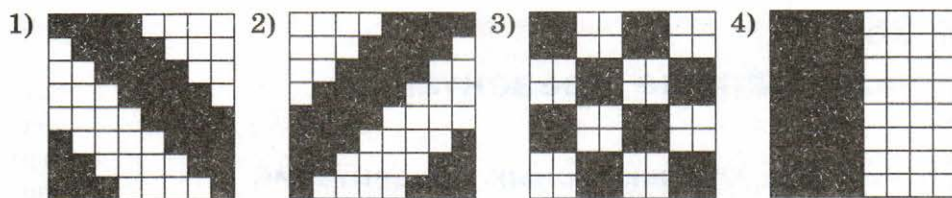


Вопросы и задания

1. Назовите основные виды графических изображений, выделяемые по способу их создания.
2. Назовите основные параметры, определяющие качество растровых изображений.
3. Проведите сравнительный анализ растровой и векторной графики с точки зрения технологии создания изображений, возможности их редактирования (включая масштабирование), объёма занимаемой памяти, применения и других характеристик.
4. Сколько памяти требуется для хранения изображения размером 1280×1024 пикселей при использовании палитры из 16 777 216 цветовых оттенков?
5. Какие преимущества при формировании изображения обеспечивает механизм слоёв?
6. Подготовьте небольшое сообщение об истории разработки кривых Безье.
7. В зависимости от области применения выделяют научную, деловую, конструкторскую, иллюстративную, художественную и рекламную графику. Подберите печатные или электронные образцы изображений каждого вида.
8. Что такое формат графического файла? Предложите классификацию известных вам форматов графических файлов.
9. В связи с чем в графических форматах применяются алгоритмы сжатия?
10. Какая из следующих картинок будет иметь самый короткий код при сжатии алгоритмом RLE¹⁾?

1) По материалам международного конкурса по информатике «Бобёр».





11. Сообщение, сжатое с помощью алгоритма RLE, имеет вид:

```
8O4X11O6X5O1X1O1X2O2X2O2X5O1X1O1X1O2X1O2X1O2X
4O1X1O1X1O2X1O2X1O2X5O5X1O1X2O2X5O5X1O4X6O6X1O3X
7O1OX2O16X
```

Восстановите первоначальный вид сообщения, если известно, что оно состоит из 10 строк, в каждой из которых содержится по 16 символов.

12. Рассчитайте разрешение 10,1-дюймового экрана нетбука в ppi, если его разрешение в пикселях 1024×600 .
13. Укажите физические размеры цифрового изображения в пикселях, достаточные для того, чтобы обеспечить высокое качество его отпечатка размером 15×20 см.
14. Одним из серьёзных недостатков цифровой фотографии считаются принципиальные трудности доказательства её аутентичности. Прокомментируйте это утверждение.
15. Сравните цифровую и плёночную фотографии с точки зрения технологии создания изображений, их качества, возможности обработки и распространения и т. д.
16. Выразите предпочтительные размеры цифровых фотографий, указанные в таблице 5.3 в мегапикселях. Сравните их с возможностями фотокамеры, имеющейся в вашем распоряжении или кого-то из членов вашей семьи. Какой вывод о возможностях своей фотокамеры вы можете сделать?
17. Какую информацию можно получить с помощью гистограммы распределения количества пикселей изображения по их яркости?
18. Исследуя меню графического редактора GIMP, найдите возможность исправления на фотографиях эффекта «красных глаз».
19. Почему при цветовой коррекции нельзя настроить один определённый цвет?

§ 25

Компьютерные презентации

25.1. Виды компьютерных презентаций

В профессиональной деятельности многих специалистов часто возникает потребность в представлении чего-то (идей, проектов, продукции и т. д.) широкой аудитории.

Презентация (от англ. *presentation* — представление) — это публичный способ представления информации, наглядный и эффектный. **Компьютерная презентация** — это электронный мультимедийный документ, который создают и используют для подачи информации широкой аудитории в наглядном и лаконичном виде.

Демонстрируют компьютерные презентации на персональных компьютерах, демонстрационных экранах больших размеров, с помощью мультимедийных проекторов и т. д.

Различают слайдовые и потоковые компьютерные презентации. Презентация первого типа аналогична набору фотографий, второго — кинофильму.

Слайдовая презентация разрабатывается как последовательность слайдов — отдельных экранных страниц, каждая из которых может содержать текстовые, графические, видео- и аудио-объекты, а также гиперссылки. Основные свойства отдельных объектов представлены на схеме (рис. 5.14).

Наиболее распространёнными программными средствами для создания слайдовых презентаций являются Microsoft PowerPoint и OpenOffice Impress.

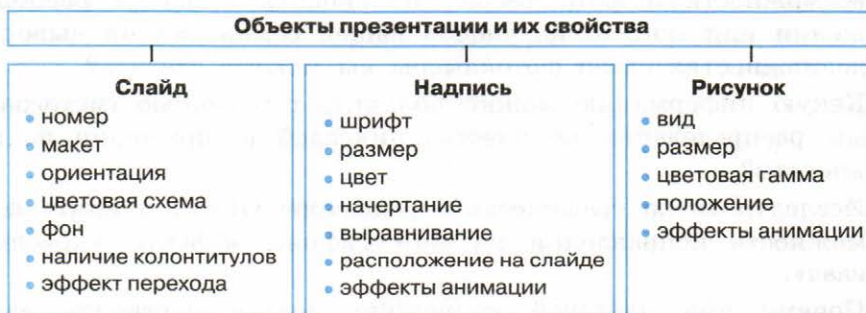


Рис. 5.14. Свойства основных объектов презентаций

Потоковые презентации предназначены для непрерывного воспроизведения последовательности (потока) объектов с предварительно определённым временем показа каждого из них. Пример программы для создания таких презентаций — Windows Movie Maker (Киностудия Windows).

В последнее время большую популярность приобрёл веб-сервис **Prezi (prezi.com)**, позволяющий создавать интерактивные мультимедийные презентации с нелинейной структурой. Его основные преимущества: бесплатный доступ; возможность создания презентации на любом компьютере, подключённом к Интернету; возможность совместного доступа пользователей к презентациям.

Работа веб-сервиса Prezi.com основана на технологии масштабирования (приближения и удаления объектов). В отличие от «классической» презентации, выполненной в Microsoft PowerPoint или OpenOffice Impress, где презентация разбита на слайды, в Prezi используется не переход от слайда к слайду, а увеличение отдельных частей единственного слайда¹⁾.

25.2. Создание презентации

Создание презентации — это творческий процесс, во многом зависящий от индивидуальности её разработчика. Тем не менее он базируется на ряде принципов, соблюдение которых даёт возможность создавать качественные продукты.

Создавая презентацию, следует придерживаться следующих этапов:

- 1) планирование (разработка структуры) презентации;
- 2) создание и редактирование слайдов, монтаж презентации;
- 3) репетиция выступления с разработанной презентацией, устранение выявленных недочётов.

Прежде всего определяют цель презентации. В зависимости от этого планируется содержание презентации, в том числе примерное количество слайдов и объектов, которые будут на них размещены. Определяются типы слайдов, которые будут использованы в презентации, структура презентации, а также система навигации — способ перехода от одного слайда к другому.

Зная, какие именно информационные объекты будут размещены на слайде, вы можете выбрать соответствующий макет слайда, на котором уже определены места для размещения требуемых объектов. В любом случае право редактирования слайдов, т. е.

¹⁾ По материалам сайта oprezi.ru.

изменения содержания, формы и места расположения информационных объектов, остаётся за разработчиком.

❓ Выясните, какие основные типы слайдов (макеты) предусмотрены в имеющемся в вашем распоряжении редакторе презентаций.

Создать презентацию пользователь может:

- на основе шаблона;
- на основе существующей презентации;
- полностью самостоятельно.

После того как определена цель создания презентации, можно попробовать подобрать шаблон, рекомендуемый профессионалами для достижения подобной цели. Шаблон — это специальная заготовка из нескольких слайдов, в которых предусмотрены места для ввода определённых информационных объектов. Последовательность слайдов в шаблоне выстроена так, чтобы помочь вам наиболее эффективно достичь поставленной цели.

❓ Выясните, какие шаблоны доступны в редакторе презентаций, имеющемся в вашем распоряжении. Попробуйте воспользоваться одним из имеющихся шаблонов.

Скорее всего, вы не найдёте нужного вам шаблона или вас не устроит логика представления материала в найденном шаблоне. Поэтому чаще всего пользователи выбирают новую презентацию или пустой шаблон, т. е. разрабатывают презентацию самостоятельно. В этом случае необходимо самостоятельно разработать композицию каждого слайда.

Композиция (от лат. *compositio* — составление, связывание, сложение, соединение) — составление целого из частей.

При самостоятельной разработке композиции слайда нужно соблюдать требования целостности и выразительности презентации, учитывать закономерности восприятия информации человеком.

При просмотре презентации у зрителя должно создаваться ощущение единого целого (свойство целостности композиции). Все элементы презентации должны быть чем-то связаны — единым стилем оформления, единым подходом к реализации анимации, стандартным для данной презентации размещением повторяющихся элементов и т. п.

Важно, чтобы основная идея презентации или отдельного слайда подавалась предельно точно (свойство выразительности). Высокий уровень выразительности достигается умелым использованием контрастных цветов, изменением формы объекта, разме-

щением наиболее важных объектов в центре слайда или вставкой их в простую геометрическую фигуру — круг, овал, квадрат, треугольник.

Большое значение для восприятия информации имеет взаимное размещение отдельных объектов на слайде. Симметричное размещение объектов подчёркивает статичность, надёжность и гармоничность композиции, а асимметрия — способность к быстрому развитию и готовность к изменениям, мобильность композиции. Следует учитывать, что наличие полей в слайдах даёт ощущение пространства, а их отсутствие — уменьшает комфортность, создаёт чувство замкнутости.

В процессе создания презентации размещать элементы управления следует таким образом, чтобы они чётко различались между собой, а размеры элементов обеспечивали их быстрый выбор и т. п.

Для создания эффективной и гармоничной презентации следует правильно выбрать её основной цвет — цвет фона большинства слайдов. Во время выбора основного цвета рекомендуется учитывать так называемую психологическую характеристику цвета, выражающую его влияние на психическое состояние человека. Это влияние может различаться в зависимости от возраста, социального статуса и настроения человека. Также следует учитывать, что воздействие цвета на психику человека ослабевает при уменьшении его интенсивности и яркости.

Приведём психологическую характеристику нескольких цветов:

- красный цвет — энергичный, агрессивный, возбуждающий, на определённое время активизирует все функции организма, поднимает настроение;
- оранжевый цвет — цвет радости, удовольствия, стремления к достижениям; оказывает благотворное действие на психику человека, снимает напряжение;
- жёлтый цвет — уменьшает утомляемость, стимулирует органы зрения и нервную систему, способствует умственной деятельности и решению проблем;
- зелёный цвет — физиологически наиболее благоприятный для человека, уменьшает напряжение и успокаивает нервную систему, на длительное время увеличивает работоспособность;
- фиолетовый цвет — может вызывать неуравновешенность, ощущение незащищённости.

Для получения более полной информации рекомендуем вам познакомиться с колористикой (от лат. *color* — цвет) — наукой о цвете, его свойствах, особенностях восприятия цвета людьми разных возрастных и социальных категорий.



Кроме основного цвета можно подобрать несколько вспомогательных цветов, которые в совокупности составят цветовую гамму презентации. При этом можно использовать цветовой круг, один из вариантов которого приведён на рисунке 5.15.

Цветовой круг состоит из 12 секторов. Чем ближе друг к другу находятся секторы, тем более гармоничным является сочетание их цветов.

В презентации можно использовать следующие сочетания цветов:

- монохроматические (оттеночные) цвета — цвета, которые размещены в одном секторе на цветовом круге; фактически это один и тот же цвет, но с разной насыщенностью;
- дополнительные цвета — два цвета, которые на цветовом круге размещены друг напротив друга, например синий и оранжевый, фиолетовый и жёлтый и т. п.;

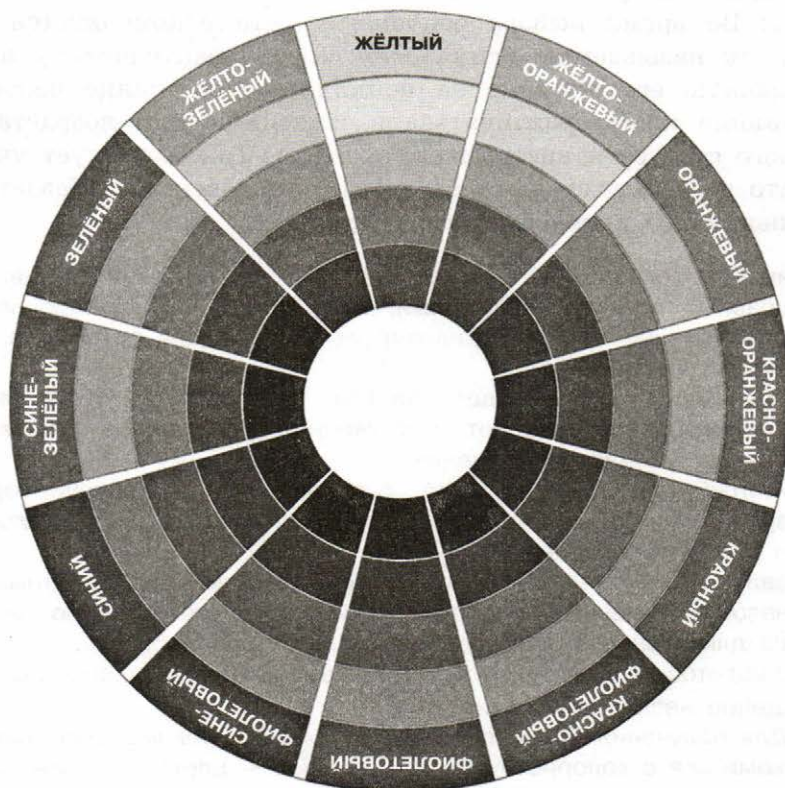


Рис. 5.15. Цветовой круг

- контрастные цвета — два цвета, между которыми на цветовом круге находится три промежуточных цвета, например синий и красный, фиолетовый и оранжевый и т. п.;
- тёплые цвета — цвета, которые размещены в правых секторах цветового круга, от красного к жёлто-зелёному;
- холодные цвета — цвета, которые размещены в левых секторах цветового круга, от красно-фиолетового к зелёному.

Важной составляющей цветовой гаммы презентации является цвет символов текста. Для того чтобы текст лучше читался, он должен быть как можно более контрастным по отношению к фону. Так, с белым фоном лучше всего сочетаются чёрный, синий и красный цвета.

На эффективность восприятия текста в презентациях кроме цвета большое влияние оказывает вид шрифта, его размер, начертание, а также количество текста на слайде презентации. При разработке презентаций следует иметь в виду, что:

- шрифты без засечек (например, Arial) воспринимаются лучше, чем с засечками (например, Times New Roman);
- размер символов должен быть достаточным для их распознавания из самой отдалённой части аудитории, в которой проходит демонстрация;
- чем больше вариантов начертания фрагментов текста использовано на одном слайде, тем хуже воспринимается текст;
- чем меньше текста на слайдах, тем лучше воспринимается презентация.

Пользователь во время создания новой презентации на основе шаблона или существующей презентации выбирает и определённый стиль её оформления. Как вам уже известно, под стилем мы понимаем совокупность значений свойств разных объектов. Соответственно стиль оформления слайда задаёт: формат символов (шрифт, размер символов, начертание, эффекты, цвет и т. п.), формат фона (цвет, наличие, размещение и вид графических объектов), дополнительные цвета, формат графических и других объектов.

Стиль оформления презентации пользователь может разрабатывать самостоятельно. Но во избежание возможных трудностей и ошибок лучше воспользоваться имеющимися заготовками — так называемыми темами презентаций, разработанными профессиональными дизайнерами с соблюдением всех необходимых требований композиции и колористики.

Для использования определённой темы оформления презентации в Microsoft PowerPoint нужно:

- создать новую или открыть существующую презентацию;
- выбрать на ленте вкладку **Дизайн**;
- в группе **Темы** выбрать подходящую тему оформления.

Выборанную тему лучше применить ко всем слайдам презентации. Чтобы применить выбранную тему к отдельным слайдам презентации, их следует выделить, вызвать контекстное меню эскиза темы и выбрать в нём соответствующий вариант.

Выбрав определённую тему, пользователь в дальнейшем может изменить цвета, шрифты, эффекты и стили фона.

Аналогично текстовому процессору Microsoft Word, в Microsoft PowerPoint пользователь может разработать собственные темы оформления слайдов презентации, определив для них фон, наборы цветов и шрифты, и сохранить их для последующего использования. Соответствующие инструменты находятся на вкладке **Вид** в группе **Режимы образцов**.

Выбор цветовой гаммы, размера и расположение объектов на слайде, а также ряд других вопросов, возникающих в процессе создания презентации, тесно связаны с эргономикой.

Эргономика (от др.-греч. ἔργον — работа и νόμος — закон) — наука об эффективности и безопасности использования человеком устройств, средств, инструментов на основе учёта особенностей построения и функционирования человеческого организма.

Эргономика широко используется при проектировании интерфейсов пользователя, помогает создавать комфортные условия в работе с соответствующим программным обеспечением, способствующие сохранению здоровья и снижению усталости, повышению эффективности труда.

Как правило, важнейшая цель презентации — привлечение внимания аудитории к излагаемому материалу. Очень часто для этого используются всевозможные анимационные эффекты.

Компьютерная анимация — компьютерная имитация движения с помощью изменения формы объектов или показа последовательных изображений с фазами движения.

В компьютерных презентациях есть возможность добавления разнообразных анимационных эффектов к размещённым на слайдах графическим и текстовым объектам.

Кроме анимации объектов в презентациях часто используют анимацию при смене слайдов. Это позволяет акцентировать внимание аудитории на переходе к изложению следующей мысли или идеи.

В целом, разумное (умеренное) использование анимационных эффектов обеспечивает лучшую наглядность и динамичность показа и в результате — большую эффективность презентации.

Редакторы презентаций имеют достаточно широкий набор средств для настройки показа презентации. Пользователь может установить один из трёх режимов показа презентации:

- 1) управляемый докладчиком (полный экран) — для случая, когда докладчик демонстрирует презентацию аудитории;
- 2) управляемый пользователем (окно) — для случая, когда презентация создана для самостоятельного просмотра пользователем на мониторе компьютера; в этом случае пользователь имеет возможность изменять заданную автором длительность просмотра каждого из слайдов;
- 3) автоматический (полный экран) — для случая, когда не предусматривается вмешательство зрителя в процесс демонстрации презентации, например при демонстрации презентации на выставочном стенде, в торговом зале, на вокзале и т. п.

В заключение перечислим основные возможности редакторов презентаций:

- включение в слайды презентации текстов, графических изображений, видео- и аудиообъектов;
- редактирование и форматирование объектов презентации;
- использование шаблонов и стилей оформления слайдов;
- применение эффектов анимации к объектам презентации;
- настройка последовательности и длительности отображения объектов презентации;
- настройка режимов демонстрации слайдов на экране монитора или с использованием мультимедийного проектора;
- демонстрация созданной презентации;
- подготовка слайдов презентации к печати на принтере;
- сохранение презентации в файлах разных форматов для воспроизведения с использованием разных программных продуктов;
- включение в файл презентации средств воспроизведения презентации для демонстрации её на компьютере, на котором не установлена ни одна система обработки презентаций.

САМОЕ ГЛАВНОЕ

Компьютерная презентация — это электронный мультимедийный документ, который создают и используют для подачи информации широкой аудитории в наглядном и лаконичном виде.

Различают слайдовые и потоковые компьютерные презентации.

Слайдовая презентация разрабатывается как последовательность слайдов — отдельных экранных страниц, каждая из которых может содержать текстовые, графические, видео- и аудио-объекты, а также гиперссылки. Наиболее распространёнными программными средствами для создания слайдовых презентаций являются Microsoft PowerPoint и OpenOffice Impress.

Потоковые презентации предназначены для непрерывного воспроизведения последовательности (потока) объектов с предварительно определённым временем показа каждого из них. Пример программы для создания таких презентаций — Windows Movie Maker (Киностудия Windows).

Следует придерживаться следующих этапов создания презентации:

- 1) планирование (разработка структуры) презентации;
- 2) создание и редактирование слайдов, монтаж презентации;
- 3) репетиция выступления с разработанной презентацией, устранение выявленных недочётов.

Процесс создания презентации будет значительно проще при использовании разработанных профессиональными дизайнерами тем (стилей) презентаций. При разработке собственного стиля презентации следует учитывать закономерности восприятия информации человеком.

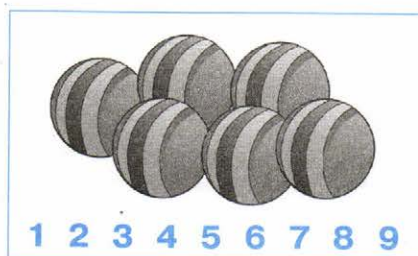
Разумное (умеренное) использование анимационных эффектов обеспечивает лучшую наглядность и динамичность показа и в результате — большую эффективность презентации.



Вопросы и задания

1. Для каких целей создаются презентации?
2. Назовите виды компьютерных презентаций и инструменты для их создания.
3. Каковы основные этапы разработки компьютерной презентации? Вспомните основные этапы разработки программного обеспечения. Соотнесите их с разработкой компьютерной презентации.

4. Исследуйте и попытайтесь классифицировать анимационные эффекты, которые можно использовать в имеющемся в вашем распоряжении редакторе презентаций. Всегда ли использование анимационных эффектов способствует большей эффективности презентации?
5. Подготовьте презентацию об учёном, внёшем вклад в развитие компьютерных наук. Презентация должна содержать не более пяти наиболее впечатляющих фактов из жизни этого человека. Попробуйте создать презентацию с помощью веб-сервиса Prezi.com (материалы для его самостоятельного освоения можно найти на сайте oprezi.ru/o-prezi.html). Будьте готовы выступить с презентацией перед своими одноклассниками.
6. С помощью редактора презентаций создайте электронное учебное пособие по математике для учеников первого класса. Презентация должна содержать не менее девяти слайдов. На каждом слайде должно быть изображено некоторое количество одинаковых предметов и цифры от 1 до 9. При щелчке мышью по числу, соответствующему количеству изображённых предметов, должна зазвучать короткая пощриательная мелодия, а все прочие числа должны исчезнуть. В случае ошибочного ответа выбранное число должно исчезнуть. Вот возможный вид слайда:



7. Слайдовую презентацию, созданную в редакторе презентаций, можно сохранить в нескольких форматах. Исследуйте возможности сохранения презентации в форматах PDF-документа и видеofilmа. Какой формат позволяет превратить слайдовую презентацию в потоковую?
8. С помощью редактора презентаций Петя решил создать слайд-шоу со звуковым сопровождением. В слайд-шоу последовательно воспроизводится 10 слайдов с неповторяющимися изображениями, размером 1024×512 точек, закоди-

рованных с использованием цветовой палитры, содержащей 65 536 цвета. Каждый слайд проигрывается 4 секунды. Переключение слайдов выполняется мгновенно. На протяжении всего слайд-шоу проигрывается моноаудиофайл, кодированный с частотой дискретизации 32 000 Гц при глубине звука 16 бит. Известно, что сжатие изображений и звука не производилось, а вся служебная информация об организации слайд-шоу занимает 10 Кбайт. Сможет ли Петя сохранить своё слайд-шоу на флешке ёмкостью 2 Гбайта, если известно, что она уже заполнена на 90%? Укажите размер слайд-шоу в килобайтах.

9. Кто такой Роберт Гаскинс и каков его вклад в создание программы для подготовки презентационных материалов с использованием компьютера?

Дополнительные материалы к главе смотрите в авторской мастерской.

Введение	3
Глава 1. ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ	
§ 1. Информация. Информационная грамотность и информационная культура	5
§ 2. Подходы к измерению информации	16
§ 3. Информационные связи в системах различной природы	30
§ 4. Обработка информации	35
§ 5. Передача и хранение информации.....	50
Глава 2. КОМПЬЮТЕР И ЕГО ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	
§ 6. История развития вычислительной техники	62
§ 7. Основопологающие принципы устройства ЭВМ.....	72
§ 8. Программное обеспечение компьютера	82
§ 9. Файловая система компьютера.....	90
Глава 3. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ	
§ 10. Представление чисел в позиционных системах счисления	99
§ 11. Перевод чисел из одной позиционной системы счисления в другую	110
§ 12. Арифметические операции в позиционных системах счисления	120
§ 13. Представление чисел в компьютере.....	129
§ 14. Кодирование текстовой информации	138
§ 15. Кодирование графической информации	145
§ 16. Кодирование звуковой информации.....	159

Глава 4. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ И АЛГЕБРЫ ЛОГИКИ.....	166
§ 17. Некоторые сведения из теории множеств	166
§ 18. Алгебра логики.....	174
§ 19. Таблицы истинности.....	189
§ 20. Преобразование логических выражений.....	197
§ 21. Элементы схемотехники. Логические схемы.....	209
§ 22. Логические задачи и способы их решения	219
Глава 5. СОВРЕМЕННЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ И ОБРАБОТКИ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ.....	232
§ 23. Текстовые документы.....	233
§ 24. Объекты компьютерной графики.....	253
§ 25. Компьютерные презентации	276

Цветовой круг



Фрактальные изображения

