



Информатика

Экзаменационные билеты - 9 класс

Теория

Информация:

процессы, измерение, кодирование.

Алгоритмизация и программирование.

**КОМПЛЕКТ ДИДАКТИЧЕСКИХ МАТЕРИАЛОВ
ДЛЯ ПОДГОТОВКИ К ГОСУДАРСТВЕННОЙ ИТОГОВОЙ АТТЕСТАЦИИ
ПО ИНФОРМАТИКЕ В 9 КЛАССЕ**

- Разработан на основе положения «О примерных билетах для сдачи экзамена по выбору выпускниками 9 классов общеобразовательных учреждений Российской Федерации, осуществивших переход на новый государственный образовательный стандарт основного общего образования» (Письмо Федеральной службы по надзору в сфере образования и науки от 18 января 2007 г. № 01-14/08-01). Полный список экзаменационных билетов и критериев их оценки приведен в полном соответствии с данным положением в пособии «Рекомендации по выполнению практических заданий».
- Комплект пособий состоит из трех частей:
 - Теория - часть 1: «Информация: процессы, измерение, кодирование. Алгоритмизация и программирование».
 - Теория - часть 2: «Аппаратная часть ПК. Системное и прикладное программное обеспечение».
 - Практика: «Рекомендации по выполнению практических заданий».
- Нумерация параграфов в пособиях не сплошная, а полностью соответствует номерам соответствующих билетов. Например, параграфы 14.1, 14.2 и 14.3 отражают три раздела, соответствующих теоретическому вопросу билета № 14. При этом порядок следования параграфов отражает логическую последовательность изложения материала курса информатики, поэтому данный комплект может быть использован не только при подготовке к экзамену, но и в качестве замены или дополнения к учебнику по информатике. Ответы на экзаменационные вопросы изложены кратко: в основном, это 3 - 4 страницы на 1 теоретический вопрос билета.
- *В данном пособии теоретический материал курса информатики сгруппирован вокруг следующих его разделов:*
 - *Информация, ее измерение и кодирование; информационные ресурсы/процессы/системы - билеты /параграфы/ №№ 1 - 6 и 14;*
 - *Основы алгоритмизации и программирования на языке Pascal - билеты /параграфы/ №№ 7 - 10.*

© Минцис Дмитрий Александрович

учитель информатики, заместитель директора по ИКТ школы № 53

2009 год.

1.1 Информация

Сегодня человек живет в мире, где информация имеет огромное значение. Жизненно важно научиться правильно с ней работать, используя различные технические устройства: от мобильного телефона до компьютера. Именно внедрение компьютерной техники во все сферы деятельности человека послужило толчком к развитию информатики - науки о поиске, получении, хранении, обработке и передаче информации при помощи информационных систем, важнейшим элементом которых является компьютер.

Понятие «информация» является базовым в курсе информатики, поэтому невозможно дать его четкое определение через другие, более простые понятия (аналогично, например, понятию «точка» в геометрии). В некотором приближении **информация** - это данные об объектах окружающего мира, которыми обмениваются между собой люди и живая природа с целью увеличения знания о нем.

Информация в человеческом обществе.

Человек не может существовать иначе как получая через органы чувств и обрабатывая информацию. Сразу после рождения человек начинает познавать мир; процесс развития продолжается в школе, где ученик получает информацию от учителей; в течение всей своей жизни он общается (обменивается информацией) при помощи речи, жестов, письма, различных технических устройств, обрабатывая огромный объем поступающей к нему информации; многие профессии связаны в основном только с информацией (журналисты, программисты, юристы и т.п.). Более того, человек, лишенный общения, чаще всего становится психически неполноценным.

Свойства информации, определяющие ее качество: полезность (практическая ценность), достоверность (истинность); актуальность (своевременность); полнота (всесторонность); понятность.

Для человека информация подразделяется в зависимости от типа воспринимающих её рецепторов на следующие **виды**:

- Визуальная - информация, поступающая через органы зрения (глаза): при чтении книги, сообщений с экрана монитора, просмотре телепередачи и т.д. Через зрение мы получаем до 90% информации об окружающем мире.
- Аудиальная - информация, поступающая через органы слуха (уши): восприятие речи, музыки и т.д. Через слух мы получаем до 9% информации.
- Тактильная - информация, поступающая через органы осязания (кожа), например восприятие «на ощупь» состояния поверхности предмета (глад-

кий или шершавый) или его температуры (холодный или теплый).

- Вкусовая - информация, поступающая через органы вкуса (язык), например восприятие вкуса еды.
- Обонятельная - информация, поступающая через органы обоняния (нос): ощущение запахов, которые распространяют объекты.

Информация в живой природе.

Совершенно очевидна роль информации и для животных: только способность вовремя получить информацию об окружающем мире позволяет им выжить в условиях естественного отбора. Поэтому многие из них обладают более совершенными органами чувств, нежели человек: собаки имеют отличное обоняние; дельфины воспринимают звуки, неслышимые человеком; кошки отлично видят в темноте и т.д.

Применимо понятие «информация» и в растительном мире: растения распускаются весной только при определенных погодных условиях: они воспринимают информацию о состоянии окружающей среды (температура воздуха и почвы, интенсивность солнечных лучей), используя ее для запуска соответствующих физико-химических процессов в клетках.

1.2 Информационный процесс. Информационная система

Информационный процесс - совокупность последовательных действий, производимых над информацией (поиск, получение, хранение, обработка и передача) для достижения поставленной цели.

Например, поставлена цель разобраться в некотором вопросе, подготовив по нему реферативную работу. При создании реферата можно выделить следующие этапы информационного процесса: сначала необходимо найти материал для реферата, например с помощью интернета (поиск информации), найденные по заданной теме материалы копируются (получение информации) и размещаются в виде файлов на компьютере пользователя (хранение информации), затем из нескольких файлов с добавлением своих мыслей компонуется итоговый файл реферата (обработка информации), который распечатывается и сдается учителю (передача информации).

Информационная система - совокупность аппаратных и программных средств, а также персонала, используемых для управления информационными процессами. Наиболее распространенная на сегодня информационная система: компьютер и средства телекоммуникаций + программное обеспечение + человек. Ниже представлена классическая структура работы информационной системы:



Например, для управления полетом спутника в его бортовой компьютер вносится программа (ввод информации), работа по которой (обработка информации) приводит к включению тех или иных механизмов; результаты их деятельности передаются в центр управления полетом (вывод информации). Инженеры на основе полученных данных принимают решение и могут внести изменения в программу полета: скорректированная программа (обратная связь) вносится в бортовой компьютер (ввод информации) и т.д.

1.3 Языки: естественные и формальные

В процессе становления *homo sapiens* (человека разумного) для обмена информацией между людьми возникли **естественные языки** – знаковые системы, соотносящие звучание (написание) и понятие об объекте. На сегодняшний день насчитывается ~ 5 – 7 тысяч языков, при этом 2/3 населения планеты говорит на 40 из них. Больше всего людей используют для общения китайский, английский, испанский, хинди, арабский, русский, португальский, французский и немецкий языки.

В основе языка лежит алфавит - набор символов (знаков), которые человек различает по их начертанию. Например, в основе русского языка лежит кириллица, содержащая 33 знака; английский язык использует латиницу из 26 знаков; китайский язык использует алфавит из десятков тысяч знаков (иероглифов) и т.д. Последовательности символов алфавита в соответствии с правилами грамматики образуют основные объекты языка - слова. А из слов в соответствие с правилами синтаксиса языка образуются предложения. В связи с огромными историческими периодами, в течение которых формировались и развивались естественные языки их грамматика и синтаксис формулируются с помощью огромного количества правил, из которых существует большое количество исключений – это создает, кроме прочего, серьезные трудности при изучении иностранных языков.

В процессе развития человеческой цивилизации появилась необходимость в создании **формальных языков** – специально созданных языков для обмена информацией в определенных областях. Примеры формальных языков: системы счисления, ноты, азбука Морзе, языки программирования и др.

Например, любая система счисления является формальным языком, имеющим алфавит (цифры), с помощью которых записываются объекты (числа), над которыми могут производиться арифметические операции по строго определенным правилам. Азбука Морзе в качестве алфавита имеет всего два знака (точка и тире), с помощью разных последовательностей которых кодируются буквы и цифры.

Так как формальные языки создавались искусственно, они, в отличие от естественных языков, построены на сравнительно небольшом количестве строгих правил грамматики и синтаксиса без каких-либо исключений.

14.1 Информационные ресурсы общества

Ресурс – это запас или источник некоторых средств. В индустриальном обществе традиционно выделяют следующие виды ресурсов: материальные (станки, автомобили), природные (леса, поля), трудовые (люди), финансовые (деньги); энергетические (нефть, газ). В информационном обществе, которое приходит на смену индустриальному, особую роль стали играть информационные ресурсы.

Согласно федеральному закону «Об информации, информатизации и защите информации» информационные ресурсы – это отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах). Иными словами, **информационные ресурсы** – это знания, подготовленные людьми для социального использования в обществе и зафиксированные на материальном носителе.

Структура информационных ресурсов России (статистика на 2005 год):

➤ **Библиотечная сеть** насчитывает около 150 тысяч библиотек. При этом более чем в 3 тысячах центральных библиотек развернуты автоматизированные комплексы, поддерживающие электронную картотеку (поиск книг через базы данных), а также доступ в интернет, к национальным и мировым библиотечным сетям. В крупных библиотеках ведется работа по созданию электронных копий печатных документов.

Школьные библиотеки оснащаются компьютерами, на которых устанавливается специализированное программное обеспечение для учета библиотечного фонда и его пользователей. Для учащихся в интернете стали доступны электронные библиотечные ресурсы с произведениями школьной программы, например: <http://www.schlib.ru> и <http://lib.prosv.ru>.

➤ **Архивный фонд** включает в себя 15 федеральных архивов, в которых хранится более 600 млн. документов, ежегодно он пополняется почти на 2 млн. единиц. При этом компьютерные технологии существенно меняют традиционное представление о «пыльных бумажных архивах» - им на смену приходят электронные накопители информации под управлением суперкомпьютеров.

➤ **Информационные ресурсы социальной сферы.**

В здравоохранении информация аккумулируется в государственной центральной научной медицинской библиотеке, информационно-аналитическом центре Минздравсоцразвития, информационных медицинских центрах в каждом субъекте федерации.

В образовании основой системы информационных ресурсов традиционно являются библиотеки более чем 500 вузов страны с фондом хранения более 300 млн. единиц; развивается федеральная компьютерная университетская сеть RUNNet; открываются региональные центры новых информационных технологий в образовании и т.д.

Развиваются внутриведомственные информационные ресурсы и в других областях социальной сферы: пенсионного обеспечения, миграционной службы, физической культуры и спорта и т.д.

➤ **Информационные ресурсы отраслей материального производства:** электронные массивы информации имеют более 60% предприятий гражданских отраслей промышленности, 70% предприятий оборонного комплекса, около 50% предприятий агропромышленного комплекса.

➤ **Государственная система научно-технической информации** занимается обработкой и хранением научно-технической документации: сюда, например, поступает информация о всех патентах на открытия и изобретения.

Кроме вышперечисленных к информационным ресурсам относятся государственная система правовой информации, государственная система статистики, информационные ресурсы органов государственной власти и местного самоуправления, информация о природных ресурсах и другие.

14.2 Основы информационной безопасности

Сегодня как никогда становится актуальной крылатая фраза «Кто владеет информацией – тот правит миром». Значимость информационных ресурсов постоянно растет: их совокупная стоимость на рынке становится сопоставима со стоимостью традиционных ресурсов, а рейтинг самых богатых людей мира последние 15 лет с небольшими перерывами возглавляет основатель корпорации Microsoft (производителя информации - программного обеспечения) Билл Гейтс. По некоторым данным деньги лишь 10% времени существуют в своем физическом виде, в остальное время они находятся в виде информации (на карте, на счету в банке и т.п.). А если учесть, что существует информация, содержащая государственную, военную, коммерческую или врачебную тайну, то становится очевидным необходимость защиты ее от несанкционированного доступа.

Поэтому **организации**, для которых информация является важнейшим ресурсом, обращают серьезное внимание на ее защиту:

- Безопасность зданий, где хранится важная информация: повышенная устойчивость к стихийным бедствиям и кражам, пропускной режим и т.п.
- Контроль доступа к информации: пароли, идентификация личности по отпечаткам пальцев, голосу, сетчатке глаз.
- Разграничение доступа к информации: каждому пользователю в зависимости от полномочий и служебной необходимости предоставляются права доступа только к определенной части информации.
- Дублирование информации и каналов ее передачи: создание копии информации на случай повреждения оригинала, наличие резервного канала связи для обмена информацией.
- Использование криптографии – науки о шифрах: наиболее важная информация кодируется таким образом, что не имея на руках ключа для дешифровки, воспользоваться ею невозможно.

Защите информации большое внимание уделяют и **производители программного обеспечения**. Для защиты своих программ от нелегального копирования и использования они применяют:

- Серийный номер, который необходимо ввести при установке программы: поставляется вместе с лицензионным установочным диском.
- Размещение на установочном диске закодированного программного ключа, который теряется при копировании, и без которого программа не может быть установлена.

- Аппаратный ключ, который чаще всего присоединяется к LPT-порту компьютера. Защищаемая программа в начале своей работы обращается к ключу, запрашивая секретный код; если же аппаратный ключ не обнаруживается, программа прекращает свое выполнение.

Каждый **пользователь** компьютера также должен уметь защищать информацию на своем компьютере:

- Система паролей: кроме пароля на свой профиль в операционной системе, это может быть административный пароль, без введения которого не начнется процесс загрузки операционной системы (устанавливается в программе BIOS Setup, на вход в которую также может быть установлен пароль).
- Антивирусная программа, которая должна своевременно обновляться.
- Брандмауэр (firewall), блокирующий нежелательную информацию, поступающую из сети: это, например, дает возможность предотвратить теоретически возможный доступ к компьютеру для любого пользователя интернета.
- Установка прав доступа (полный, только чтение, по паролю) на диск (папку, файл), причем права могут быть различными для разных пользователей.
- Установка биометрической системы идентификации пользователя, например загрузка компьютера по отпечатку пальца.

14.3 Правовые и этические нормы работы с информацией

Развитие рыночных отношений в информационной деятельности поставило вопрос о защите информации как объекта интеллектуальной собственности и имущественных прав на нее. В России принят ряд законов, таких как «Об информации, информатизации и защите информации», «Об авторском праве и смежных правах», «О правовой охране программ для ЭВМ и баз данных», «О правовой охране топологий интегральных схем». В них предлагается рассматривать информационные ресурсы в двух аспектах: как материальный продукт, который можно покупать и продавать; и как интеллектуальный продукт, на который распространяется право интеллектуальной собственности и авторское право.

Данные законы утверждает следующее: **авторские права** собственности распространяются на исходный текст (на языке программирования или в машинном коде) всех видов компьютерных программ; но не распространяются на идеи и принципы (алгоритм, интерфейс), лежащие в основе программы. Авторское право на программу возникает автоматически при ее создании, какая-либо регистрация не требуется. Для оповещения о своих

правах разработчик программы может использовать знак охраны авторского права, состоящий из: символа ©, наименования правообладателя и года (годов) выпуска различных версий данной программы. Например, знак охраны авторских прав на Windows Vista выглядит следующим образом: © Корпорация Майкрософт, 2007.

Программы по их правовому статусу можно разделить на три группы:

➤ **Лицензионные программы.** Фирма-производитель программы (автор) создает программу и продает права на ее тиражирование фирме-дистрибьютору (продавцу). Частный пользователь покупает лицензионный диск у продавца, но при установке программы на свой ПК обязан согласиться с лицензионным соглашением, в котором, в частности, запрещается установка данной программы на другой компьютер.

Довольно часто производители предоставляют скидки при покупке лицензий на использование своей программы на большом количестве компьютеров или в учебных заведениях. Так, например, на льготных условиях государством в рамках национального приоритетного проекта «Образование» была осуществлена закупка большого количества дистрибутивов для установки в российских образовательных учреждениях: Windows 2000, XP, Vista; Microsoft Office 2003, 2007; Adobe Creative Suite 2, 3, 4 и другие.

В соответствии с лицензионным соглашением разработчики программы гарантируют ее нормальное функционирование в определенной операционной системе и несут за это ответственность.

➤ **Условно бесплатные программы (shareware).** Многие фирмы в целях продвижения на рынок своих программных продуктов предлагают пользователям бесплатную версию программы, но либо с ограниченными возможностями, либо с ограничением срока действия. Ознакомившись с данной программой, пользователь может принять решение о покупке ее полной (лицензионной) версии или удалить ее со своего компьютера.

➤ **Свободно распространяемые программы (freeware).** Таковыми являются большое количество программ, создаваемых программистами-энтузиастами, но также и фирмами: драйвера устройств, бета-версии программ для их тестирования и т.д.

Законодательно запрещено:

- Нелегальное копирование и продажа лицензионного программного обеспечения (в этом случае его называют «пиратским»). Запрещено и использование пиратского программного обеспечения, то есть по закону виноваты и продавец и покупатель пиратского диска.

- Пропаганда насилия, наркотиков и порнографии в интернете. Здесь виновными также могут быть признаны не только авторы запрещенных сайтов, но и пользователи, регулярно их посещающие.

Кроме юридических норм, культурный человек должен соблюдать и **этические нормы** - нормы поведения человека, принятые в обществе, но не закрепленные законодательно. Например, не следует заниматься спамом - рассылать по электронной почте в огромном количестве незнакомым людям рекламные сообщения (эту норму переводят из этической в правовую), хамить или отклоняться от темы обсуждения в конференциях и т.д.

2.1 Измерение информации

В определении понятия «количество информации» существует два основных подхода: содержательный и алфавитный.

Содержательный (вероятностный) подход к измерению информации

Мы говорим: «человек получает информацию через органы чувств, хранит информацию в памяти, обрабатывает ее». Таким образом, информация в процессе познания окружающего мира выступает как новое знание. Никакой информации не несет сообщение, содержание которого непонятно или уже известно. Если же некоторое сообщение приводит к увеличению наших знаний, то оно содержит информацию.

Например, $2 \times 2 = 4$ для взрослого человека не является информацией, а для ученика 1-го класса - вполне может быть информацией, так как увеличивает его знание об окружающем мире. Таким образом, информация является характеристикой не сообщения, а соотношения между сообщением и человеком: количество одной и той же информации может быть различно для разных людей.

Основная единица измерения количества информации - **бит (содержательный подход)** - количество информации, которое содержит сообщение, уменьшающее неопределенность знаний в два раза. Например, сообщение о том, что при бросании монеты выпал «орел» означает, что из двух равновероятных событий («орел» и «решка») случилось одно и наша неопределенность знаний уменьшилась в 2 раза. Следовательно, такое сообщение несет 1 бит информации.

В общем случае, сообщение о том, что одно из N равновероятных событий произошло несет $I = \log_2 N$ бит информации (формула Хартли). Отсюда следует, что $N = 2^I$, то есть при помощи I бит может быть закодировано 2^I

равновероятных событий. Примеры:

➤ Сообщение «при двух бросках монеты выпали решка + решка» указывает на реализацию некоторого события из 4 равновозможных (орел + орел, орел + решка, решка + орел, решка + решка) $\Rightarrow I = \log_2 4 = 2$ бита.

➤ Сообщение «Ученик занял на олимпиаде по информатике 3-е место из 12 участников» указывает на реализацию некоторого события из 12 возможных $\Rightarrow I = \log_2 12 \approx 3,6$ бит. Однако бит является элементарной неделимой единицей измерения информации: количество информации может быть выражено только в целом количестве бит. Поэтому ответом в данной задаче будет $I = 4$ бит – это минимальное целое число бит, с помощью которого может быть закодировано 12 событий. Таким образом, для определения количества информации более актуальна формула $2^{I_{\min}} \geq N$.

➤ В соревновании участвуют 20 спортсменов. Каков информационный объем сообщения о промежуточном финише 10-и из них.

Информационный объем финиша 1 спортсмена из 20 возможных: $2^{I_{\min}} \geq 20 \Rightarrow I = 5$ бит. А для 10 спортсменов: $5 \text{ бит} \cdot 10 = 50$ бит.

Алфавитный подход к измерению информации

Алфавитный подход к информации хорошо характеризует следующее высказывание: «Величайшее литературное произведение - не что иное, как в беспорядке разбросанный алфавит» (Жан Кокто), то есть содержание произведения нас не интересует, мы рассматриваем его как набор символов.

Действительно, техническим устройствам нет дела до содержательной стороны сообщений. Любая информация в компьютере представляется при помощи двоичной системы счисления, алфавит которой состоит из двух знаков (цифр): 0 и 1.

Бит (алфавитный подход) - одна цифра при двоичном кодировании: 0 или 1 (binary digit - двоичная цифра). Для расчета объема памяти (в битах), который занимает та или иная информация в компьютере, необходимо подсчитать количество 0 и 1, с помощью которых она кодируется.

Таким образом при алфавитном подходе измеряется не содержание информации с точки зрения ее новизны и полезности, а размер несущего информацию сообщения. Поэтому, например с точки зрения алфавитного подхода информационный объем слов «фывапролд» и «компьютер» совершенно одинаков, а слов «ученик» и «учащийся» - различен (содержательный подход дает диаметрально противоположный результат).

Бит - очень маленькая единица измерения информации, поэтому чаще оперируют понятиями байт и производными от него единицами.

2.2 Единицы измерения информации

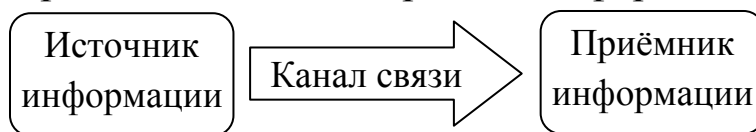
- Байт = 8 бит.
- Килобайт 1 Кбайт = 2^{10} байт = 1 024 байт $\approx 10^3$ (тысяча) байт.
- Мегабайт 1 Мбайт = 2^{20} байт = 1 048 576 байт $\approx 10^6$ (миллион) байт.
- Гигабайт 1 Гбайт = 2^{30} байт = 1 073 741 824 байт $\approx 10^9$ (миллиард) байт.
- Терабайт 1 Тбайт = 2^{40} байт = 1 099 511 627 776 байт $\approx 10^{12}$ (триллион) байт.
- Петабайт 1 Пбайт = 2^{50} байт = 1 125 899 906 842 624 байт $\approx 10^{15}$ (квадриллион) байт.

Примеры:

- Носители информации: FD - 1,44 Мбайт; CD \sim 700 Мбайт; DVD \sim 4,7 Гбайт; Flash \sim 8 Гбайт; HDD \sim 500 Гбайт.
- Страница текста занимает \sim 10 Кбайт; рисунок формата А4 \sim 100 Кбайт, минута звука \sim 1 Мбайт, минута видео \sim 10 Мбайт.
- С помощью 1 бита можно закодировать 2 (2^1) символа: 0 и 1. С помощью 2 бит могут быть закодированы 4 (2^2) символа: 00, 01, 10, 11. С помощью 3 бит - 8 (2^3) символов: 000, 001, 010, 011, 100, 101, 110, 111. В общем случае с помощью n бит кодируется 2^n символов.
- Содержательный и алфавитный подходы приводят к совершенно разным результатам при подсчетах количества информации для одного и того же сообщения. Например, для приведенного выше сообщения «при двух бросках монеты выпали решка + решка»:
 - содержательный подход дает 2 бита информации (см. выше) – именно настолько увеличилось наше знание;
 - алфавитный подход: 44 символа (включая пробелы), для хранения каждого используется, например, 8 бит. Итого $44 \cdot 8 = 352$ бита – именно столько данное сообщение займет в памяти компьютера.
- Сколько рисунков, размером 100 Кбайт, поместится на FD и на CD?
Объем 1 рисунка: $V_1 = 100 \text{ Кбайт} = 100 \cdot 2^{10}$ байт.
Объем гибкого магнитного диска FD: $V_{FD} = 1,44 \text{ Мбайт} = 1,44 \cdot 2^{20}$ байт.
Количество рисунков на дискете: $N_{FD} = V_{FD} / V_1 = 1,44 \cdot 2^{20} / 100 \cdot 2^{10} = 1,44 \cdot 1024 / 100 \approx 14,7 \Rightarrow$ ответ - 14.
Объем оптического диска CD: $V_{CD} = 700 \text{ Мбайт} = 700 \cdot 2^{20}$ байт.
Количество рисунков на CD: $N_{CD} = V_{CD} / V_1 = 700 \cdot 2^{20} / 100 \cdot 2^{10} = 7 \cdot 1024 = 7168$.

5.1 Передача информации

Человек как социальное существо не может существовать без обмена информацией с другими людьми для решения многих личностных и социальных проблем. Простейшая схема передачи информации при этом:



Источник информации – человек или техническое устройство, которое передает информацию.

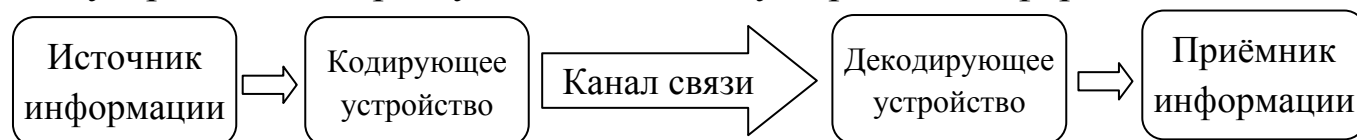
Приемник информации – человек или техническое устройство, которое получает информацию.

Канал связи – это физическая среда распространения сигналов для передачи информации от источника к приемнику.

Продemonстрируем простейшую схему передачи информации на примере общения двух людей. Информация от одного человека (источника информации) передается другому человеку (приемнику информации) при помощи каналов связи, в качестве которых могут быть использованы механические волны (звук: голосовые связки источника – слуховые анализаторы приемника) или электромагнитные волны (свет: мимика и жестикуляция источника – зрение приемника).

С развитием цивилизации стали появляться технические средства, существенно увеличивающие скорость передачи информации: от сигнальных башен с кострами, предупреждающих о приближении врага до мобильных телефонов, факсов и персональных компьютеров, позволяющих за считанные секунды доставить информацию в любую точку земного шара.

Современный канал связи, решающий задачи быстрой передачи информации на большие расстояния, построен на основе сложнейших технических устройств, которые усложняют схему передачи информации:



Например, человек (источник) пишет электронное письмо другому человеку (приемнику). Канал связи в этом случае выглядит следующим образом: - **Кодирующее устройство** - устройство, предназначенное для преобразования исходного сообщения источника информации к виду, удобному для передачи. В данном случае это компьютер источника: клавиатура – систем-

ный блок – сетевая карта (модем): сообщение источника кодируется электрическими импульсами в двоичной системе счисления.

- **Канал связи:** телефонные, оптоволоконные, радио и спутниковые линии связи, по которым передается электрический ток или электромагнитная волна, несущие информацию об исходном сообщении.

- **Декодирующее устройство** - устройство для преобразования кодированного сообщения в исходное. В данном случае это компьютер приемника: сетевая карта (модем) – системный блок – монитор: сообщение представляется в виде светового излучения.

В случае общения человек – человек напрямую также можно вести речь о кодирующих (голосовые связки) и декодирующих (слух, зрение) устройствах, сложность устройства которых намного превышает все созданные на сегодняшний день технические новинки.

Характеристики канала связи

1. **Помехоустойчивость** (надежность). В процессе кодирования / декодирования, а также передачи информации через канал связи она может теряться и искажаться. Например, мы иногда плохо слышим человека по телефону, принимаем с искажениями радио или телевизионный сигнал: часто это связано с особенностями распространения радиосигнала в окружающем пространстве, наложениями на него внешних электромагнитных воздействий (при грозах, например). Поэтому радиосигнал (электромагнитные волны) используют в качестве канала связи лишь в случае невозможности организовать проводные линии связи, которые, как правило, практически лишены подобных проблем.

2. **Пропускная способность** – скорость передачи информации по каналу связи. Она измеряется в битах в секунду и кратных единицах:

- Килобит в секунду: $1 \text{ Кбит/с} = 2^{10} = 1024 \approx 10^3$ (тысяча) бит/с.

- Мегабит в секунду: $1 \text{ Мбит/с} = 2^{20} = 1\,048\,576 \approx 10^6$ (миллион) бит/с.

- Гигабит в секунду: $1 \text{ Гбит/с} = 2^{30} = 1\,073\,741\,824 \approx 10^9$ (миллиард) бит/с.

Примеры:

➤ Обычный модем $\sim 56 \text{ Кбит/с} = 56 / 8 \text{ Кбайт/с} = 7 \text{ Кбайт/с}$

- ADSL-модем $\sim 2 \text{ Мбит/с} = 2 \cdot 2^{20} \text{ бит/с} = 2^{21}/2^3 \text{ байт/с} = 2^{18} \text{ байт/с} = 2^8 \cdot 2^{10} \text{ байт/с} = 256 \text{ Кбайт/с}$

- Выделенная линия $\sim 10 \text{ Мбит/с} = 10 / 8 \text{ Мбайт/с} = 1,25 \text{ Мбайт/с}$

- Локальная сеть $\sim 100 \text{ Мбит/с} = 100 / 8 \text{ Мбайт/с} = 12,5 \text{ Мбайт/с}$

- За какое время будет скачан файл размером 500 Мбайт при пропускной способности канала связи 5 Мбит/с?

$$V = 500 \text{ Мбайт} = 500 \cdot 2^{20} \text{ байт} = 500 \cdot 2^{23} \text{ бит.}$$

$$v = 5 \text{ Мбит/с} = 5 \cdot 2^{20} \text{ бит/с.}$$

$$t = V / v = 500 \cdot 2^{23} / 5 \cdot 2^{20} = 100 \cdot 2^3 = 800 \text{ с} \approx 13,3 \text{ мин.}$$

- Какова пропускная способность канала связи, если файл объемом 1,4 Гбайт был скачан за 2 минуты?

$$V = 1,4 \text{ Гбайт} = 1,4 \cdot 2^{30} \text{ байт} = 1,4 \cdot 2^{33} \text{ бит.}$$

$$t = 2 \text{ мин.} = 120 \text{ с.}$$

$$v = V/t = 1,4 \cdot 2^{33} / 120 = 1,4 \cdot 2^{13} \cdot 2^{20} / 120 = 1,4 \cdot 8192 / 120 \text{ Мбит/с} \approx 95,6 \text{ Мбит/с.}$$

3.1 Двоичная система счисления

Любая информация в компьютере (числа, текст, графика, аудио, видео) представляется при помощи двоичной системы счисления, то есть последовательностью из 0 и 1. Выбор такого способа кодирования связан с технической простотой его реализации: 0 - нет электрического сигнала, 1 - есть электрический сигнал (электронный ключ /транзистор/ замкнут - разомкнут, участок CD отражает - не отражает и т.п.).

Система счисления (с. сч.) - это знаковая система, в которой числа записываются по определенным правилам при помощи цифр. Цифра - это символ, имеющий определенное количественное значение.

Системы счисления делятся на позиционные и непозиционные.

- **Непозиционная с. сч.** - с. сч., в которой количественное значение цифры не зависит от ее расположения в числе. Пример - римская с.сч.: вне зависимости от расположения в числе цифра I имеет значение 1, V - 5, X - 10, L - 50, C - 100, D - 500, M - 1000.

- **Позиционная с.сч.** - с. сч., в которой количественное значение цифры зависит от ее расположения (позиции) в числе. Пример – широко используемая уже много веков арабская (десятичная) с.сч. Так в числе 125 цифра 5 имеет значение 5, а в числе 521 та же цифра 5 имеет значение 500.

Ниже рассмотрены основные понятия позиционных систем счисления на примере сравнения 10-ной и 2-ной систем счисления.

- **Основание с.сч. (P)** - количество цифр, используемых для записи числа. Значения цифр лежат в пределах от 0 до P-1.

10-ная с. сч.: P = 10; алфавит цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

2-ная с. сч.: P = 2; алфавит цифр: 0, 1.

Название с.сч. производится по ее основанию: 2-ная, 3-ная, ..., 10-ная, ...

➤ **Разряд** цифры характеризует ее позицию в числе, причем отсчет ведется справа налево, разряд единиц = 0.

Пример: 10-ная с. сч.: $4\ 3\ 8\ 9 = 4 \cdot 10^3 + 3 \cdot 10^2 + 8 \cdot 10^1 + 9 \cdot 10^0$
3 2 1 0 - разряды

➤ **Запись чисел:** при накоплении младшего разряда до максимальной цифры он обнуляется, а старший увеличивается на 1.

10-ная	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2-ная	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

Перевод целых чисел из 2-ной в 10-ную систему счисления

1. Проставить номера разрядов цифр 2-го числа: справа налево, начиная с 0-го разряда.

2. Просуммировать произведения цифр числа на 2 в степени номера разряда.

Пример. $1\ 0\ 1\ 0\ 1_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16 + 4 + 1 = 21_{10}$
4 3 2 1 0

Перевод целых чисел из 10-ной в 2-ную систему счисления

10-ное число делится на 2, при этом остатки от деления, записываемые справа налево, образуют искомое 2-ное число. Деление производится до получения в целой части единицы, которая записывается первой цифрой искомого 2-го числа.

Примеры.

$36_{10} = ?_2$

$$\begin{array}{r|l} 36 & 2 \\ \hline 18 & 100100 \\ 9 & \\ 4 & \\ 2 & \\ 1 & \end{array}$$

$36_{10} = 100100_2$

Проверка: $1\ 0\ 0\ 1\ 0\ 0_2 = 2^5 + 2^2 = 32 + 4 = 36_{10}$
5 4 3 2 1 0

$25_{10} = ?_2$

$$\begin{array}{r|l} 25 & 2 \\ \hline 12 & 11001 \\ 6 & \\ 3 & \\ 1 & \end{array}$$

$25_{10} = 11001_2$

Проверка: $1\ 1\ 0\ 0\ 1_2 = 2^4 + 2^3 + 2^0 = 16 + 8 + 1 = 25_{10}$
4 3 2 1 0

3.2 Двоичное кодирование текстовой информации

Каждый символ текста, как правило, кодируется при помощи 8 или 16 бит в зависимости от используемой кодировки.

Например, при наборе текста в Блокноте каждый символ текста кодируется при помощи 8 бит (1 байт). При нажатии на клавиатуре символьной клавиши вырабатывается соответствующий ей электрический сигнал, который преобразуется контроллером клавиатуры в двоичный код: от 00000000 (0) до 11111111 (255). Существует $2^8 = 256$ неповторяющихся комбинаций из 8 двоичных цифр, что позволяет кодировать 256 символов. Каждый символ имеет свой код в соответствии с кодовой таблицей, причем половина таблицы (коды 0 - 127) является интернациональной; вторая половина (коды 128 - 255) содержит национальный алфавит.

Интернациональная часть кодовой таблицы (соответствует ASCII - American Standard Code for Information Interchange) - от 0 до 127:

- 0 - 32: коды операций, например 8 - Backspace, 13 - Enter, 32 - пробел и т.п.
- 48 - 57: цифры (0 .. 9);
- 65 - 90: латинские прописные буквы (A .. Z);
- 97 - 122: латинские строчные буквы (a .. z).

Национальная часть кодовой таблицы (от 128 до 255) кодирует символы национального алфавита. Существует несколько кодовых таблиц для русского алфавита (кириллицы): КОИ8, CP1251, CP866, Mac, ISO; причем тексты, созданные в одной кодировке, не могут быть правильно отображены в другой без специальных программ-конверторов, которые встраиваются в приложения для работы с текстовыми документами. Например, для кодировки CP1251: 192 - 223 - коды русских заглавных букв (А .. Я, кроме Ё - 168); 224 - 255 - коды русских строчных букв (а .. я, кроме ё - 184).

Обратите внимание, что:

- Несмотря на одинаковое отображение на мониторе, русские и английские буквы представляются в компьютере по-разному. Например английское заглавное А имеет код 65 (01000001), а русское А – код 192 (11000000).
- Числа могут кодироваться непосредственно как числа (по правилам перевода в 2-ную систему), и как набор символов-цифр (по кодовой таблице). Например, 25 как число может быть представлено как 00011001, а как текст, состоящий из двух цифр 2 и 5: 00110010 00110101.

координатами. Пиксели, которые чаще всего имеет форму квадрата, выстраиваются в табличном виде и благодаря различному цвету и яркости образуют изображение. Это хорошо заметно при большом увеличении изображения, однако при их реальном размере из-за крайней малости пикселей создается впечатление плавности смены цветов в картинке.

Основные характеристики растрового изображения:

1. **Экранный размер** - количество пикселей, составляющих изображение по ширине и высоте. Например 350×220 - изображение размером 350 пикселей по горизонтали и 220 пикселей по вертикали. При этом надо иметь в виду, что экранный размер монитора, как правило 1280×1024 (для 17" и 19") или 1024×768 (для 15").

2. **Глубина цвета** - количество бит, используемое для кодирования цвета одного пикселя. Глубина цвета (n) определяет **цветовую палитру** - количество отображаемых цветовых оттенков, которая $= 2^n$: 8-битная глубина цвета позволяет кодировать $2^8 = 256$ оттенков; 16-битная (High Color) – $2^{16} > 65$ тыс. оттенков, 24-битная (True Color) – $2^{24} > 16$ млн. оттенков, 32-битная – $2^{32} > 4$ млрд. оттенков.

Цветовая модель - способ числового представления цвета. В каждой цветовой модели любой цвет получают как результат смешивания базовых цветов в различных пропорциях. Наиболее распространенной цветовой моделью является модель RGB (базовые цвета - красный /Red/, зеленый /Green/ и синий /Blue/), которая используется, например, в мониторах и телевизорах. Любой цвет в модели RGB может быть получен смешиванием трех базовых цветов и записан в следующем формате: (R, G, B), где R, G и B - интенсивности соответствующих базовых цветов. При глубине цвета = 24 бит каждому из базовых цветов выделяется по 8 бит, что соответствует $2^8=256$ -и уровням интенсивности данного цвета (от 0 до 255).

Примеры: (255, 0, 0) - красный; (0, 255, 0) - зеленый; (0, 0, 255) - синий; (255, 100, 50) - оранжевый оттенок; (56, 228, 218) - голубой оттенок и т.п. Очевидно, что черный цвет - это отсутствие света, т.е. его код (0,0,0), а белый свет - это смешение всех базовых цветов в максимальной интенсивности, следовательно его код (255, 255, 255).

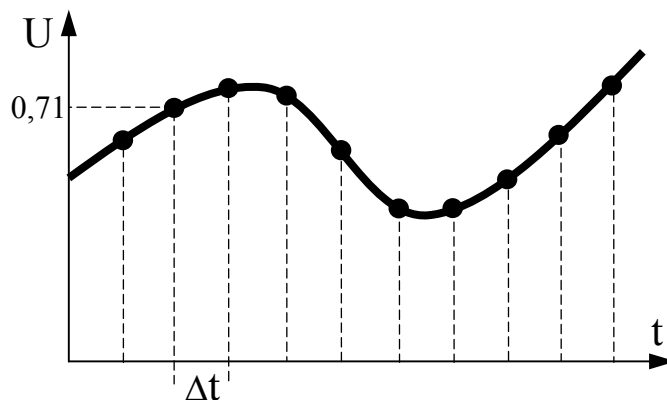
Информационный объем изображения - объем памяти, необходимый для хранения информации о всех пикселях изображения. Информационный объем зависит от экранного размера рисунка и глубины цвета. Очевидно,

что если количество точек на экране = $X \cdot Y$, а для хранения каждой точки отводится N бит, то размер графического файла = $X \cdot Y \cdot N$ бит. Например, рисунок размером 1024×768 при 32-битной палитре займет объем памяти $V = 1024 \cdot 768 \cdot 32$ бита = 25165824 бита = 3145728 байт ≈ 3 Мбайт.

4.2 Двоичное кодирование звука

Кодирование звука на примере акустической системы «микрофон - аудиокарта» выглядит следующим образом: звуковая волна преобразуется микрофоном в непрерывный электрический сигнал, который разбивается аудиокартой на короткие временные участки. Для каждого такого участка измеряется амплитуда сигнала, которая записывается двоичным числом. Рассмотрим этот процесс подробнее.

1. Значения напряжения непрерывного электрического сигнала (U) измеряются через определенные небольшие промежутки времени Δt . Например, в некоторый момент измерено $U = 0,71$ В.



2. Каждому значению U приводится в соответствие число, исходя из определенного масштаба. Например, если масштаб = $0,01$ В/единицу числа, то $U = 0,71$ В будет представлено числом 71.

3. Полученное число переводится в его двоичный код: $71_{10} = 1000111_2$, который и передается на запись, являясь, таким образом, носителем информации о напряжении электрического тока в данный момент времени.

Таким образом двоичный код звука несет информацию лишь об определенных точках (выделены на рисунке) исходной непрерывной звуковой кривой. Отсюда возможная потеря качества при оцифровке звука, которая зависит от двух параметров:

1. **Частота дискретизации** $\nu = 1/\Delta t$ - количество измерений напряжения аналогового сигнала за 1 секунду. Чем больше частота дискретизации, тем чаще измеряется напряжение исходной кривой, а следовательно лучше качество преобразования: меньшие отрезки исходного графика теряются.

2. **Глубина кодирования звука** (разрядность аудиокарты, g) - количество бит, используемое для кодирования каждой измеряемой точки кривой.

Глубина звука определяет количество отображаемых уровней громкости: аудиокарта с глубиной кодирования g позволяет отобразить 2^g уровней. Например обычная на сегодня 16-разрядная аудиокарта позволяют кодировать $2^{16} = 65536$ уровней звукового сигнала.

От этих характеристики зависит и объем памяти, требуемый для хранения 1 секунды оцифрованного звука: $V = v \cdot g$

Например, 1 секунда аудио достаточно высокого качества ($v = 44$ КГц, $g = 16$ бит) займет $44000 \cdot 16 = 704000$ бит = 88000 байт ≈ 86 Кбайт. Если учесть, что качественный звук – стерео, то есть содержит 2 аудиоканала, то 1 секунда займет $86 \cdot 2 = 172$ Кбайт. Минута звука займет соответственно $172 \cdot 60 = 10320$ Кбайт ≈ 10 Мбайт, а час - $10 \cdot 60 = 600$ Мбайт.

4.3 Двоичное кодирование видео

Видео представляет из себя большое количество отдельных кадров, которые воспроизводятся на мониторе с такой скоростью, что человеческий глаз перестает различать их дискретность, воспринимая данный кадровый поток как единое целое. Очевидно, что важнейшей характеристикой видео является его **кадровая частота** - частота смены кадров, которая измеряется в fps - frames per second (кадров в секунду). Физиологически глаз перестает замечать мелькание кадров уже при частоте их смены более 10 fps. В XX веке стандартом кино стала частота 24 fps, для телевидения был выбран стандарт 50 полукадров в секунду, в современных мониторах $\sim 60 - 70$ fps.

Очевидно, что объем памяти, занимаемый 1 секундой видео определяется как произведение объема одного кадра (графического изображения) на кадровую частоту.

Пример. Один видеокادر в стандарте PAL имеет разрешение 720×576 (414720 пикселей), пусть каждый пиксел занимает 3 байта (True Color). Итого, размер одного кадра $414720 \cdot 3 = 1244160$ байт $\approx 1,2$ Мбайт. Соответственно для хранения одной секунды видео, состоящей, как минимум из 25 кадров, потребуется $1,2 \cdot 25 = 30$ Мбайт.

Таким образом, CD емкостью 700 Мбайт вместит всего $700 / 30 \approx 23$ секунды несжатого видео; а DVD емкостью 4,7 Гбайт - $\sim 4800 / 30 = 160$ секунд (меньше 3 минут) несжатого видео. Поэтому любой видеоформат подразумевает некоторый алгоритм сжатия информации.

4.4 Алгоритмы сжатия графики, аудио и видео

Расчеты объемов, занимаемых графическими, звуковыми и особенно видеофайлами указывают на крайнюю необходимость их сжатия – уменьшения размера без или с минимальной потерей качества. На сегодняшний день существует огромное количество алгоритмов сжатия информации, некоторые из которых приведены ниже:

- **Графика (jpeg).** Группа соседних пикселей, цвет которых немного отличается, делаются одноцветными. Это позволяет не кодировать каждый пиксел отдельно, а указать в коде, что группа из некоторого количества пикселей имеет определенный цвет. Такое избавление от малозаметных глазу данных позволяет организовать сжатие файла в 10 и более раз.
- **Звук (mp3).** Человеческое ухо различает звук от 20 до 20000 Гц. Однако, низкие частоты от 20 до 2000 Гц и высокие частоты от 5000 до 20000 Гц воспринимаются плохо – они и подвергаются максимальному сжатию.
- **Видео.** Взяв один кадр за базовый, для следующих кадров можно хранить только их изменения относительно базового. Если же изменения текущего кадра относительно базового становятся значительными, то он автоматически выбирается в качестве следующего базового кадра. Таким образом, большинство кадров кодируются не полностью, а лишь изменениями от базового кадра, что существенно уменьшает объем видеофайла.

4.5 Мультимедиа

Мультимедиа (multimedia - «многие среды») - технология совместного использования различных способов представления информации: текста, графики, звука и видео. Примеры: кинофильм (видео, звук, субтитры); компьютерные презентации, игры, энциклопедии и т.д.

Практически все мультимедийные продукты являются интерактивными – предусмотрена возможность управления ими со стороны пользователя. Игры и энциклопедии интерактивны по своей природе; в презентациях есть возможность размещать управляющие элементы, которые определяют последовательность отображения слайдов; в качестве проекта рассматривается даже интерактивное кино, в котором пользователь сможет влиять на развитие сюжетной линии.

Термин «мультимедиа» иногда применяется и к техническим устройствам: компьютерам, проекторам, - подразумевая, что они способны эффективно работать со всеми видами информации.

6.1 Алгоритм. Исполнитель

Алгоритм – это конечная последовательность точно определенных действий, приводящих к решению поставленной задачи.

Исполнитель - устройство, способное понимать и исполнять команды алгоритма. Исполнитель обладает следующими свойствами:

- **Система команд исполнителя (СКИ)** - набор команд, который может выполнять данный исполнитель.
- **Среда исполнителя** - совокупность объектов, над которыми исполнитель может совершать действия.

Алгоритмами пронизана вся наша жизнь. Человек, часто не задумываясь об этом, выполняет тысячи разнообразных алгоритмов, некоторые из которых доведены до автоматизма (например, алгоритм завязывания шнурков на ботинках): любое планирование своей деятельности (поход в магазин, план действий на день) является разработкой собственного алгоритма; часто мы пользуемся и чужими алгоритмами: решение задач по математике, рецепты приготовления блюд, инструкции по работе технических устройств и т.д.

Примеры исполнителей

Исполнитель	СКИ	Среда
Человек	Широкий набор умственных и физических операций	Объекты окружающего мира
Робот-манипулятор	Перемещение вверх, вниз; повороты вправо, влево; взять, отпустить деталь и т.п.	Конвейер
Система программирования Pascal	Команды языка программирования Pascal	Компьютер

6.2 Свойства алгоритмов

Любой алгоритм должен отвечать следующим требованиям:

1. **Понятность** - в алгоритме должны содержаться только те команды, которые входят в его СКИ. Чем меньшей СКИ владеет исполнитель, тем сложнее становится описание алгоритма для него. Например, для исполнителя-десятиклассника алгоритм решения квадратного уравнения должен выглядеть так: решить уравнение, потом сообщить результат, - так как в его СКИ

должно входить умение решать подобные уравнения. А для исполнителя-семиклассника потребуются подробно расписать все этапы решения данного уравнения, так как его СКИ еще не включает подобные знания.

2. Дискретность - дискретное (пошаговое) выполнение команд алгоритма с точной фиксацией выполнения одной команды и начала выполнения следующей. Например, для объяснения человеку алгоритма нахождения какого-либо здания в незнакомом городе важно пошагово расписать последовательность его действий:

1. Проехать две остановки на автобусе.
2. Перейти через подземный переход.
3. Пройти 150 метров прямо.

Такой дискретный алгоритм гораздо надежнее, нежели сложносочиненный путанный рассказ о дороге с упоминанием красот любимого города, что вызовет лишь путаницу в голове собеседника.

Порядок выполнения действий очень важен: попробуйте, например, переставить местами шаги 2 и 3 в алгоритме перехода улицы:

1. Подойти к пешеходному перекрестку.
2. Дождаться зеленого света светофора.
3. Перейти улицу.

- и Вы рискуете попасть под машину.

3. Результативность - алгоритм должен обязательно привести к необходимому конечному результату. Например, вышеупомянутый алгоритм перехода улицы может не сработать, так как не предусматривает возможности отсутствия или поломки светофора. Усовершенствуем его:

1. Подойти к пешеходному перекрестку.
2. Если светофор работает, то дождаться зеленого света и перейти улицу.
3. Если светофор не работает, то перейти улицу по правилам для нерегулируемых пешеходных переходов.

4. Детерминированность – каждый шаг алгоритма должен быть четким и однозначным: тогда выполнение алгоритма для исполнителя не потребует никаких дополнительных указаний. Например шаг 3 предыдущего алгоритма содержит инструкцию для школьника «перейти улицу по правилам для нерегулируемых пешеходных переходов», которые он может не знать, следовательно эти инструкции также должны быть включены в алгоритм.

5. **Массовость** - возможность применения данного алгоритма для решения многих задач одного типа. Какой смысл был бы в изучении алгоритма решения квадратного уравнения, если с его помощью решалось бы только одно уравнение?

6.3 Способы записи алгоритмов

Существует достаточно большое количество форм представления алгоритмов, основные из которых перечислены ниже.

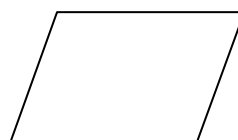
➤ **Словесная форма.** Объясняя другому человеку решение некоторой проблемы, мы описываем ее алгоритм на естественном языке. Но каждый человек делает это по-своему, поэтому один и тот же алгоритм может быть описан разными людьми по-разному, а отсутствие четких стандартов в словесном выражении алгоритма часто приводит к недопониманию.

➤ **Программа.** Если алгоритм предназначен для исполнения техническим устройством, например роботом-манипулятором или компьютером, то он представляется в виде программы на одном из языков программирования.

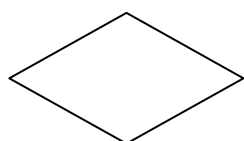
➤ **Графическая форма** является наиболее наглядной для человека. Например, на упаковках продуктов быстрого приготовления и в инструкциях для бытовой техники алгоритм их использования часто приводится в рисунках. Но особое место в информатике занимает такая графическая форма представления алгоритма как **блок-схема** – способ записи алгоритма с помощью стандартного набора (более 20-и) графических блоков, перечень и условные обозначения которых жестко зафиксированы ГОСТом (государственным стандартом). Основными графическими блоками, которые мы будем использовать, являются:



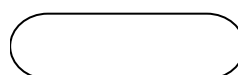
- операционный блок



- блок ввода и вывода информации



- блок проверки условия



- блок начала и конца алгоритма

7.1 Основные алгоритмические структуры

Внутри алгоритмов можно выделить группы шагов, отличающиеся внутренней структурой – алгоритмические конструкции. К основным алгоритмическим конструкциям относятся следование, ветвление и цикл.

1. **Линейная** последовательность (следование) - группа шагов алгоритма, выполняемых последовательно друг за другом без каких-либо условий. Если весь алгоритм представляет собой линейную последовательность шагов, то его называют линейным.

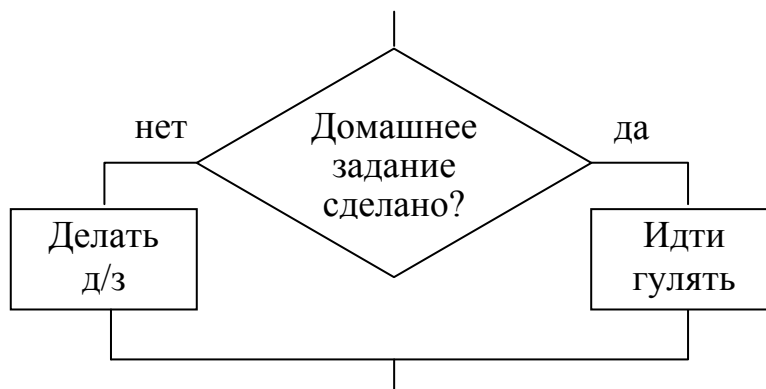
Пример: упрощенный алгоритм перехода улицы ⇒



Пример линейного алгоритма для языка Pascal – в разделе 8.

2. **Ветвление** – алгоритмическая конструкция, в которой выполнение того или иного действия зависит от истинности некоторого условия.

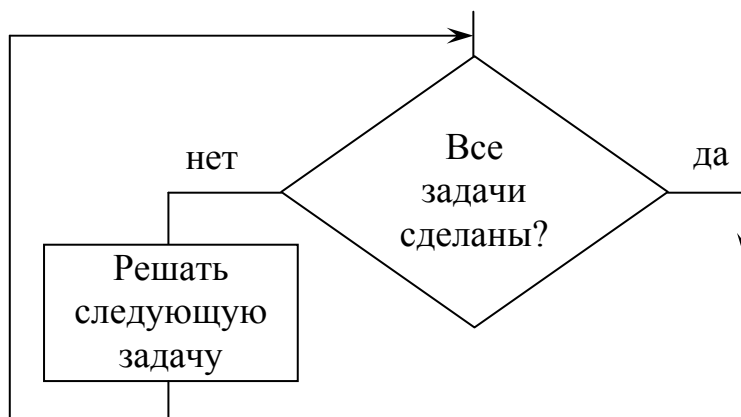
Пример: фрагмент алгоритма действий учащегося в послеурочное время ⇒



Пример блок-схем ветвления для языка Pascal – в разделе 10.

3. **Цикл** - алгоритмическая конструкция, в которой многократно выполняется одна и та же последовательность действий, называемая телом цикла. Каждое однократное исполнение тела цикла называется итерацией.

Пример: фрагмент алгоритма действий учащегося на контрольной работе ⇒



Пример блок-схем цикла для языка Pascal – в разделе 10.

Вспомогательный алгоритм

В последнем алгоритмическом примере инструкция «Решать следующую задачу» может оказаться непосильной для учащегося, если его уровень знаний по данному предмету невысок.

В этом случае для достижения поставленной цели (отличной отметки за контрольную) данный операционный блок придется вынести в отдельный (вспомогательный) алгоритм, в котором на понятном учащемуся языке сформулировать все этапы решения данной задачи.

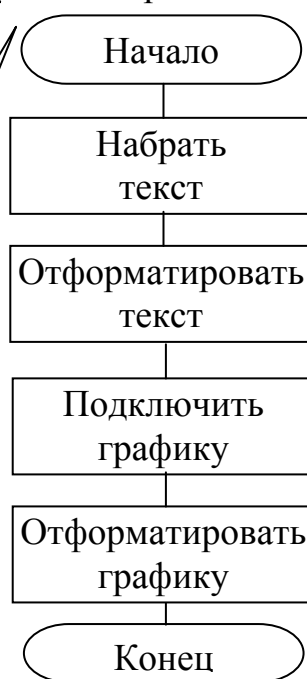
Процесс решения сложной задачи довольно часто сводится к решению нескольких более простых подзадач. Соответственно при разработке сложного алгоритма он может разбиваться на отдельные алгоритмы, которые называются вспомогательными. При этом создание вспомогательных алгоритмов можно поручить разным разработчикам. Важно только, чтобы каждый такой вспомогательный алгоритм описывал решение какой-либо подзадачи с использованием команд, понятных конкретному исполнителю.

Например, в школе создается web-сайт. Приведенный, хоть и крайне упрощенный основной алгоритм решения данной задачи, вряд ли под силу большинству учащихся. Но отдельные алгоритмические шаги, если они затем детализируются (конечно, более тщательно, чем в приведенном примере) во вспомогательном алгоритме «Создание web-страницы» учащимся под силу и они могут взять на себя эти трудоемкие операции.

Основной алгоритм
«Создание web-сайта»



Вспомогательный алгоритм
«Создание web-страницы»



8.1 Язык программирования Pascal

Программа – это запись алгоритма для исполнения техническим устройством, например компьютером. Программа пишется на одном из языков программирования, общее число которых приближается к 10 тысячам. Мы рассмотрим возможности одного из них, широко применяющегося как в обучении, так и промышленном программировании, - языка Pascal.

В основе Pascal, как и любого другого языка, лежит **алфавит**:

1. Латинские (английские) буквы A - Z: запись всех команд программы (строчные и прописные буквы при этом не различаются).
2. Цифры 0 - 9. Математическая запись чисел (запись чисел на Pascal):
11 (11); -9 (-9); 3,5 (3.5); 0,000617 (0.000617 или .000617 или 6.17E-4);
310000 (310000 или 3.1E+5).
3. Знаки операций: + - * /
mod - остаток от деления целых чисел, например: 14 mod 5 = 4
div - целая часть от деления целых чисел, например 14 div 5 = 2
4. Знаки отношения: = > < >= <= <>
5. Служебные символы, в т.ч. ; - разделение операторов;
, - разделение элементов списка;
{ } - скобки комментария и т.д.

Русские буквы А – Я можно использовать только в комментариях и при выводе информации на экран.

8.2 Структура программы

Структура программы	Пример
<ul style="list-style-type: none"> • Заголовок программы. 1. Раздел описаний, используемых в программе: - библиотек; - переменных величин и т.д. 2. Тело программы. Начинается с оператора begin. Последовательность операторов, которые реализуют алгоритм решения поставленной задачи. Заканчивается оператором end. 	<pre> program primer; uses wincrt; var x: integer; begin write ('Введите отметку '); read (x); if x=2 then write('неудовлетворительно'); if x=3 then write('удовлетворительно'); if x=4 then write('хорошо'); if x=5 then write('отлично'); if (x<2) or (x>5) then write('такой отметки нет'); end. </pre>

- Заголовок программы для экономии времени можно не набирать.
- До тела программы (до оператора begin) все обрабатываемые в ней объекты должны быть описаны (объявлены) в разделе описаний. Мы будем ограничиваться описанием библиотеки wincrt, которая подключает дополнительные возможности системы программирования, и описанием переменных величин.
- Алгоритм решения задачи записывается в теле программы посредством операторов - зарезервированных слов языка, отражающих те или иные алгоритмические конструкции.
- Программа может оформляться произвольно, например, нет необходимости следовать правилам машинописного набора текста. Однако, для наглядности программы можно следовать следующим правилам:
 - размещать каждый оператор на отдельной строке; только в случае, если два-три последовательно идущих оператора логически связаны друг с другом, они также могут быть размещены на одной строке;
 - операторы, входящие в состав других операторов, если они размещаются на отдельной строке, должны печататься с отступом слева.

8.3 Вывод информации. Запись арифметических выражений

Вывод информации на экран дисплея осуществляется при помощи операторов **WRITE** и **WRITELN**:

- Выводимая информация заключается в круглые скобки, а текст должен дополнительно браться в одинарные штрих-кавычки. Например:
 - writeln ('мама мыла раму') - вывод на экран текста «мама мыла раму».
 - writeln (x) - вывод на экран содержимого ячейки x.
 - writeln ('x') - вывод на экран буквы x.
- При выводе на экран разнотипной информации она разделяется через запятую. Например, writeln ('Вы получили оценку ',x, ' по информатике').
- Writeln, в отличие от write, дополнительно переводит курсор в начало следующей строки, то есть следующая выводимая на экран информация появится с новой строки.

Запись арифметических выражений осуществляется в соответствии с общепринятым в математике приоритетом выполнения операций: скобки - умножение и деление - сложение и вычитание. Например, для вычисления значения выражения $\frac{1,15 \cdot 3,3 + 3}{230}$ необходимо набрать $(1.15 * 3.3 + 3) / 230$.

При получении нецелого числа компьютер выдает результат в нормальном виде (для вышеуказанного примера: 2.9543478261E-02, что означает 0,029543478261). Обычно такая точность вычисления является излишней. Более наглядный способ вывода нецелых чисел с точностью до N знаков после запятой: WRITELN (<арифметическое выражение>:0:N). Например, WRITELN ((1.15*3.3+3)/230:0:2) даст результат 0.03.

При записи арифметических выражений могут быть использованы следующие **стандартные арифметические функции**:

Функция	Назначение	Пример	
		запись	значение
PI	3.14159...	2*PI	6.28
SQR	квадрат	SQR (9)	81
SQRT	квадратный корень	SQRT (9)	3
SIN	синус	SIN (PI/4)	0.71
COS	косинус	COS (PI/3)	0.5
ABS	модуль	ABS (-5.3)	5.3
FRAC	дробная часть числа	FRAC (13.79)	0.79
TRUNC	целая часть числа	TRUNC (13.79)	13

При записи арифметических выражений надо помнить, что:

- Отрицательные числа необходимо заключать в скобки.
- Аргумент тригонометрических функций должен выражаться не в градусной, а в радианной мере. Например, для нахождения $\cos 60^\circ$ нужно набирать не $\cos (60)$, а $\cos (\pi/3)$. Общая формула для записи α° : $\pi * \alpha / 180$. Например, $\cos 46^\circ$ должен быть записан как $\cos (\pi*46/180)$.

- Пример записи сложного арифметического выражения: $21 + \frac{35,8^2 - \sqrt{7}}{1 + \sin 34^\circ} \cdot 2,3 :$
`writeln (21 + (sqr(35.8) - sqrt(7)) / (1 + sin(pi*34/180))*2.3 :0:2)`

```

uses wincrt;
begin
  writeln ((1.15*3.3+3)/230:0:2);
  writeln (21+(sqr(35.8)-sqrt(7))/(1+sin(pi*34/180))*2.3:0:2);
end.

```

8.4 Величины и их типы

Величина - элемент данных, хранящийся в памяти под определенным именем. В процессе выполнения программы величины могут менять свои значения (переменные) или быть постоянными (константы).

Константа – величина, не меняющая своего значения в процессе работы программы. Используемые в программе константы описываются после зарезервированного слова **CONST**. Например, `const g = 9.81; e = 5.`

Переменная – величина, которая может изменяться в процессе работы программы. Используемые в программе переменные описываются после зарезервированного слова **VAR**, где также указывается тип, к которому принадлежит переменная. Например, `var x,y: integer; z: real; s: string;`

Каждая величина имеет 3 основных характеристики: имя, тип и значение.

Имя (идентификатор) величины может состоять не только из одной английской буквы, как принято в математике. Например, для обозначения суммы можно использовать переменную *s*, а можно назвать ее *sum* или *summa*.

Тип величины – характеристика величины, определяющая множество ее допустимых значений и операций над нею. К основным (стандартным) типам величин в Паскале относятся:

Тип	Обозначение	Множество допустимых операций	Диапазон допустимых значений
Целый	INTEGER	Арифметические операции, операции сравнения	-32768 .. 32767
Вещественный	REAL		$\pm 2.9 \cdot 10^{-39} \dots 1.7 \cdot 10^{38}$
Символьный	CHAR	Операции сравнения	1 символ
Строковый	STRING	Строковые функции	строка до 256 символов
Логический	BOOLEAN	Логические операции	true, false

➤ Если в разделе описаний объявляется переменная одного типа, а в программе она используется как переменная другого типа, компьютер выдаст сообщение об ошибке типов: «Type mismatch» или «Invalid numerical format».

➤ Мы, в основном, будем работать с числовой информацией:

- Числа могут представляться как целые и любые (вещественные). Целые числа мы будем относить к типу `integer`, любые – к типу `real`. Кроме этого, для хранения целых чисел можно использовать типы `byte` (0...255), `shortint` (-128...127), `word` (0...65535), `longint` (-2147483648...2147483647). Самый широкий диапазон для чисел предоставляет тип `extended` ($\pm 3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{4932}$).

- Тип `real` дает возможность работать с любыми числами в большом диапазоне, `integer` – только с целыми в гораздо меньшем диапазоне. Тем не менее тип `integer` не бесполезен: целые числа занимают в памяти компьютера меньше места и гораздо быстрее обрабатываются, поэтому если такая воз-

возможность есть, то число необходимо относить к целому типу.

- Результат целого типа дают только следующие операции: сложение, вычитание и умножение переменных целого типа, а также операции div и mod. Результат деления целых чисел, даже заведомо целый, является величиной вещественного типа.

8.5 Оператор ввода информации

READ (< имена вводимых переменных >)

Используемые в программе переменные должны быть объявлены в разделе описаний после слова VAR.

Для более наглядного ввода информации перед оператором read ставится оператор write с подсказывающим текстом.

```

Найти класс, в котором учится школьник, по его возрасту.
uses wincrt;
const x=6;
var y:integer;
begin
  write ('Возраст = '); read (y);
  write(Ты учишься в ',y-x,' классе');
end.

```

8.6 Оператор присваивания значения переменной

<имя переменной> := <значение переменной>

Например:

- a:=21; b:=5+a {a = 21; b = 26}
- a:=5; b:=2; a:=b {a = 2; b = 2}
- a:=5; b:=2; b:=a {a = 5; b = 5}
- a:=a+1 {a ↑ на 1}

Оператор присваивания не аналогичен простому арифметическому равенству. Например, математическая запись a=a+1 является ложной при любых значениях a. А в программах запись типа a:=a+1 встречается достаточно часто: переменной с именем А присваивается значение суммы числа, которое ранее хранилось в этой переменной, и единицы, то есть содержимое ячейки увеличивается на 1 (так называемый «счетчик»).

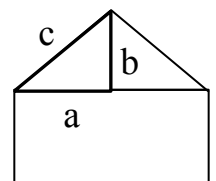
```

Найти сумму и частное двух целых чисел
uses wincrt;
var a,b,s:integer; x:real;
begin
  write ('1-ое число = '); read (a);
  write ('2-ое число = '); read (b);
  s:=a+b; x:=a/b;
  writeln ('Сумма = ',s);
  writeln ('Частное = ',x:0:2);
end.

```

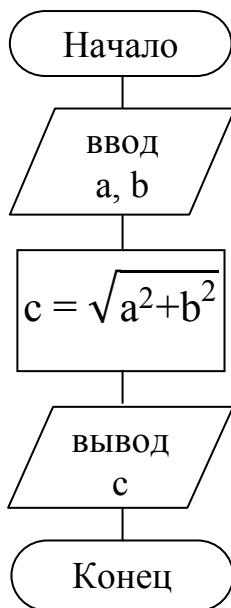
8.7 Этапы решения задачи на компьютере

1. Постановка задачи. Необходимо измерить длину стропила крыши (с), считая что прямое измерение затруднительно по техническим причинам.
2. Определение метода решения задачи - создание математиче-



ской модели. Предположим, что измерение величин отрезков a и b не составляет большого труда. А длину стропила можно найти как гипотенузу прямоугольного треугольника по теореме Пифагора: $c = \sqrt{a^2 + b^2}$

3. Разработка алгоритма решения задачи



4. Создание программы по алгоритму

```
uses wincrt;  
var a,b,c: real;  
  
begin  
  
  write ('a = '); read (a);  
  write ('b = '); read (b);  
  
  c := sqrt((sqr(a)+sqr(b)));  
  
  write ('результат = ',c:0:2);  
  
end.
```

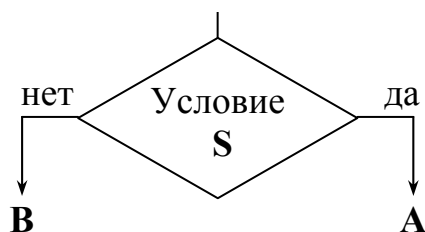
- 5. Ввод программы: набор с клавиатуры или загрузка готовой программы.
- 6. Отладка программы: исправление ошибок.
- 7. Тестирование программы - составление теста, с помощью которого можно убедиться в правильности работы программы. Например, при вводе $a=3$, $b=4$ гипотенуза должна получиться $c = \sqrt{3^2 + 4^2} = 5$.
- 8. Ввод данных и получение конечного результата.

10.1 Программирование разветвленных алгоритмов

Для программирования алгоритмической конструкции ветвления используется оператор проверки условия, который применяется в двух форматах:

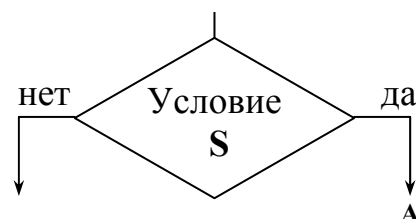
Полный формат оператора условия

IF S THEN A ELSE B



Неполный формат оператора условия

IF S THEN A



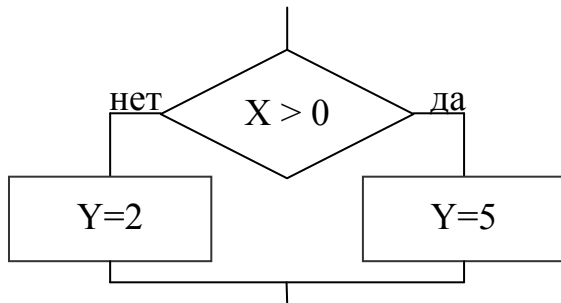
Здесь S - логическое выражение, истинность которого проверяется;

A - операторы, которые выполняются, если выражение S истинно;

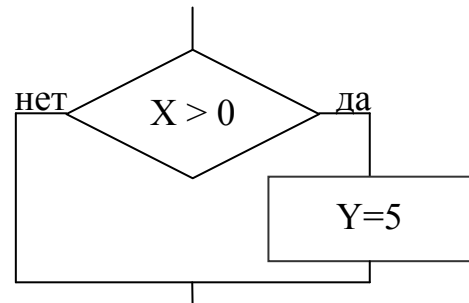
B - операторы, которые выполняются, если выражение S ложно.

Примеры блок-схем и реализующих их операторов условия:

IF X>0 THEN Y:=5 ELSE Y:=2

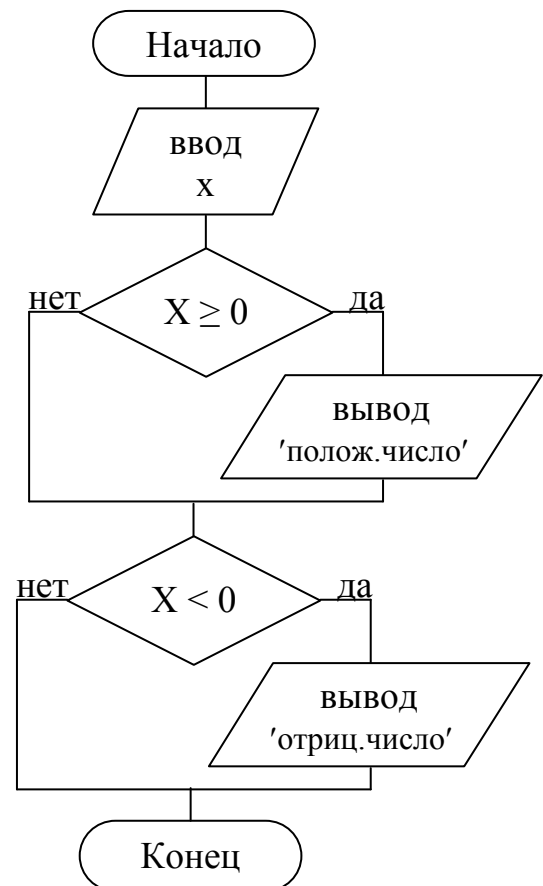
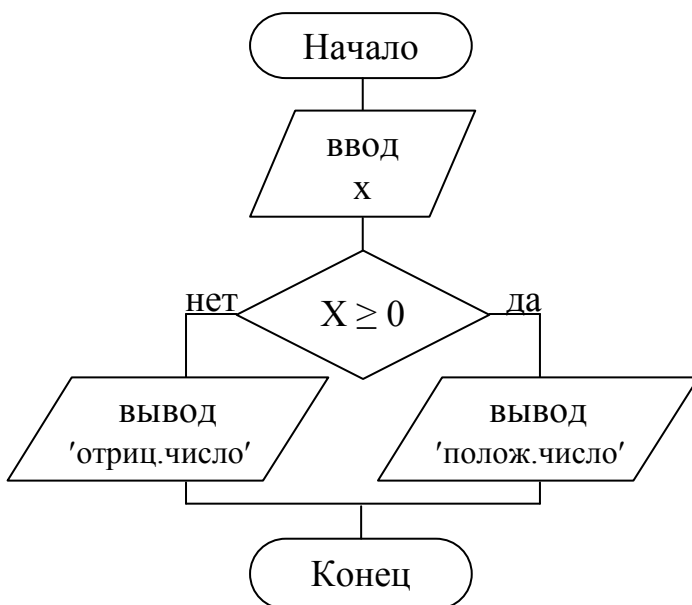


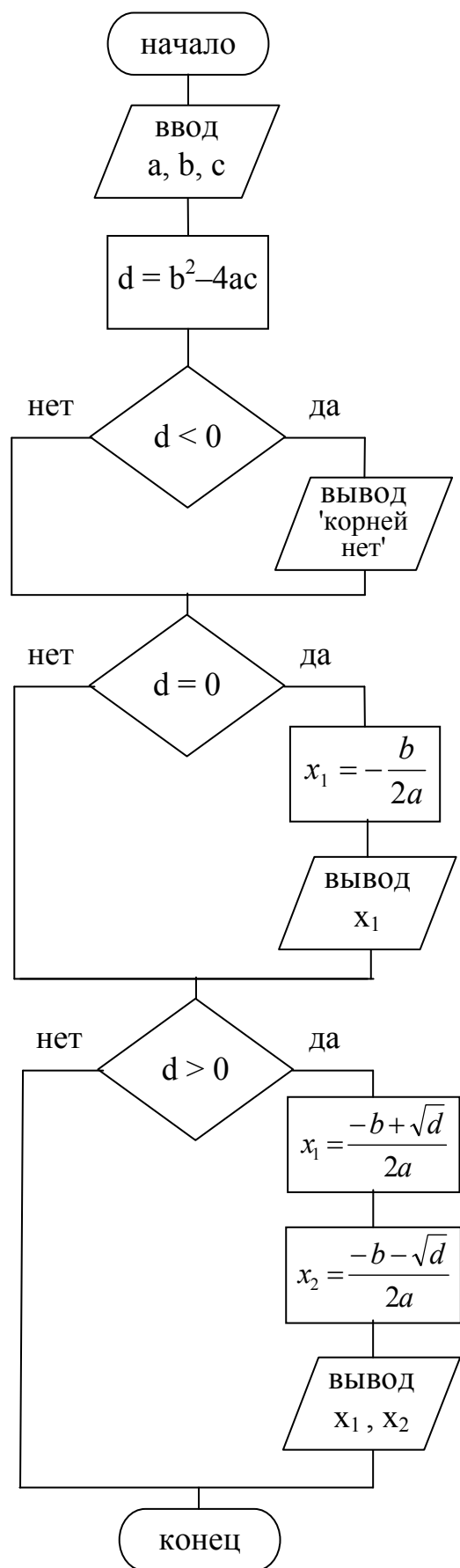
IF X>0 THEN Y:=5



Задача. В зависимости от введенного целого числа вывести на экран сообщение «положительное число» или «отрицательное число».

<pre>uses wincrt; var x:integer; begin write ('Введите число = '); read (x); if x>=0 then write('Положительное число'); if x<0 then write('Отрицательное число'); end.</pre>	<pre>uses wincrt; var x:integer; begin write ('Введите число = '); read (x); if x>=0 then write('Положительное число'); else write('Отрицательное число'); end.</pre>
--	--





Составной оператор – линейная последовательность операторов, заключенная в операторные скобки: между словами `begin` и `end`. Если к одной из ветвей алгоритма, между которыми осуществляет выбор оператор условия (т.е. к слову `THEN` или слову `ELSE`), относится более одного оператора, то всю эту группу операторов необходимо заключить в «операторные скобки» (между `begin` и `end`);).

Пример.

Найти корни квадратного уравнения вида $ax^2 + bx + c = 0$ по вводимым a , b и c .

```

program kw_ur;
  uses wincrt;
  var a,b,c,d,x1,x2: real;
begin
  write ('a = '); read (a);
  write ('b = '); read (b);
  write ('c = '); read (c);
  d:=sqr(b) - 4*a*c;
  if d<0 then write ('корней нет');
  if d=0 then
    begin
      x1:=-b/(2*a); write ('x1=',x1:0:2)
    end;
  if d>0 then
    begin
      x1:=(-b+sqrt(d))/(2*a);
      x2:=(-b-sqrt(d))/(2*a);
      writeln ('x1 = ',x1:0:2);
      write ('x2 = ',x2:0:2)
    end;
end.
  
```

С другими примерами программ на оператор условия можно ознакомиться в рекомендациях по выполнению практических экзаменационных работ.

10.2 Программирование циклических алгоритмов

Цикл – это многократное повторение группы операторов (тела цикла) до выполнения некоторого условия.

1. **Цикл с предусловием:** условие выполнения цикла проверяется до тела цикла. Реализуется оператором **WHILE S DO A**. Здесь A - тело цикла, которое выполняется, пока истинно логическое выражение S.

2. **Цикл с постусловием:** условие выхода из цикла проверяется после тела цикла. Реализуется оператором **REPEAT A UNTIL S**. Здесь A - тело цикла, которое выполняется до тех пор, пока ложно логическое выражение S.

3. **Оператор цикла с параметром**

FOR I := X1 TO X2 DO A;

FOR I := X1 DOWNTO X2 DO A;

Здесь I - имя переменной (параметр цикла); X1, X2 – арифметические выражения, определяющие начальное и конечное значения переменной; A – тело цикла.

Оператор цикла с параметром может реализовывать любую циклическую алгоритмическую конструкцию (как с пред-, так и с постусловием), при этом являясь наиболее удобным из операторов цикла.

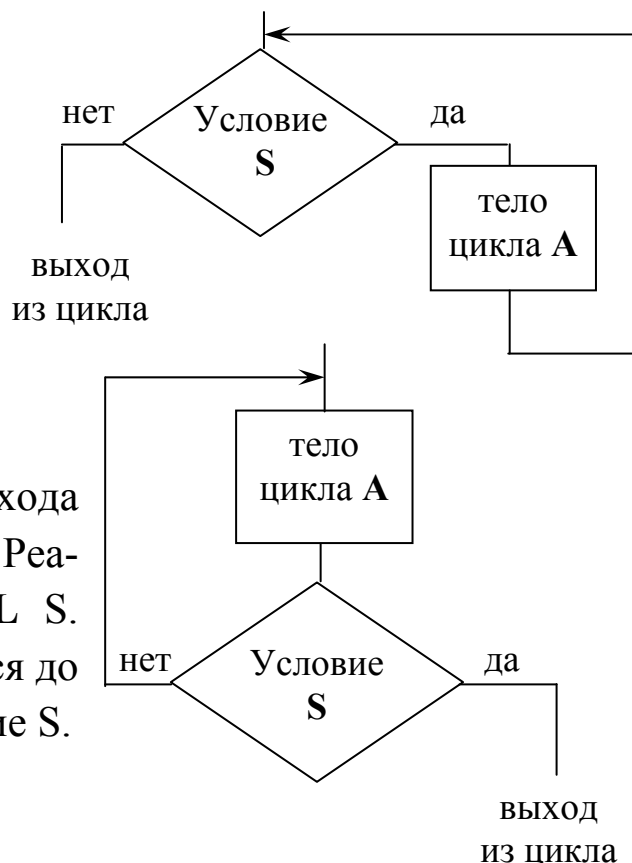
Однако использовать оператор цикла с параметром можно при выполнении следующих условий:

- параметр цикла должен быть переменной целого типа;
- шаг изменения параметра: +1 (to) или -1 (downto).

Трассировка программы

При безмашинном решении задач часто возникают проблемы с проверкой правильности созданной программы. Для этого можно использовать трассировку – пошаговое выполнение программы с прослеживанием изменений значений переменных.

Пример. Вывести на экран числовой ряд, начиная с 5 до 11 с шагом 3. Подсчитать сумму выведенных элементов.



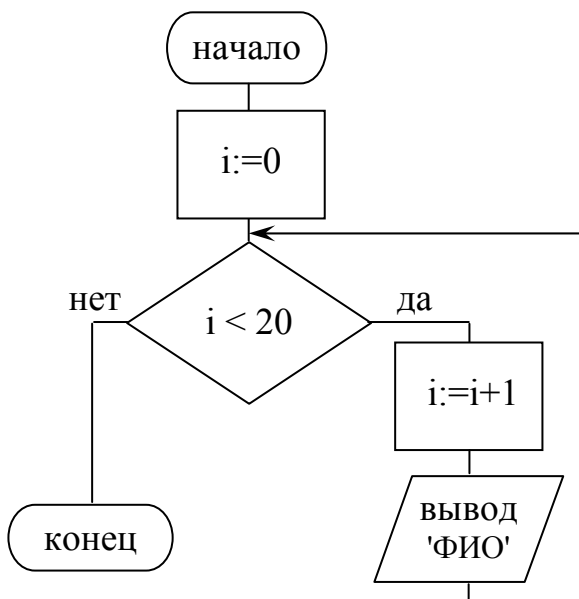
Проведем трассировку программы:

a	s	экран
5	0	
8	5	5
11	13	8
14	24	11
		сумма = 24

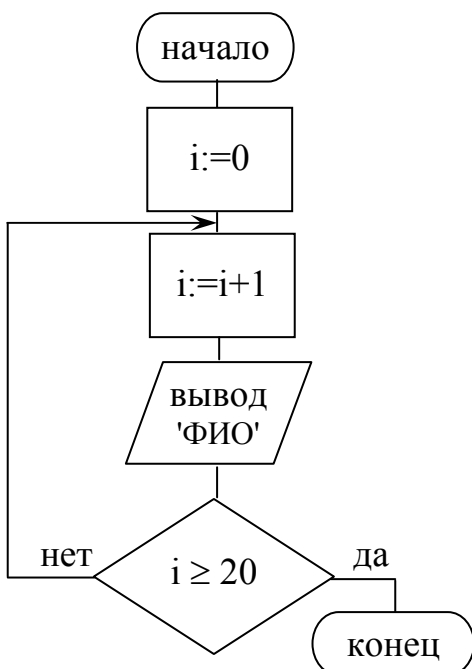
Данная трассировка позволяет сделать вывод о корректности работы данной программы.

Пример. Вывести на экран в столбик свою фамилию, имя и отчество 20 раз.

Цикл с предусловием



Цикл с постусловием



```

uses wincrt;
var a, s: integer;
begin
a:=5; s:=0;
while a <= 11 do
  begin
    write (a, ' '); s:=s+a; a:=a+3;
  end;
  writeln; write ('сумма = ',s);
end.
    
```

Цикл с предусловием

```

uses wincrt;
var i: integer;
begin
i := 0;
while i < 20 do
  begin
    i := i+1;
    writeln ('Иванов Иван Иванович');
  end;
end.
    
```

Цикл с постусловием

```

uses wincrt;
var i: integer;
begin
i := 0;
repeat
  i := i+1;
  writeln ('Иванов Иван Иванович');
until i >= 20;
end.
    
```

Цикл с параметром

```

uses wincrt;
var i: integer;
begin
for i:=1 to 20 do
  writeln ('Иванов Иван Иванович');
end.
    
```

С другими примерами программ на операторы цикла можно ознакомиться в рекомендациях по выполнению практических экзаменационных работ.

9.1 Основы алгебры логики

Алгебра логики (булева алгебра) - раздел математической логики, в которой рассматриваются операции на двухэлементном множестве (0 и 1). Основа данного раздела математики была заложена в середине XIX века английским математиком Джорджем Булем. Алгебра логики оперирует логическими высказываниями.

Логическое высказывание - это любое предположение, в отношении которого имеет смысл утверждение об его истинности или ложности. Примеры простых логических высказываний: «Москва - столица России» - истинное; «2 умножить на 2 равно 5» - ложное. При этом истинному высказыванию сопоставляется 1, ложному высказыванию - 0.

Основные логические операции

В алгебре логики над простыми логическими высказываниями определена некоторая совокупность логических операций, в результате которых получаются составные логические высказывания.

Операция	Обозначение	Таблица истинности	Пояснения															
Логическое отрицание; Операция НЕ; Инверсия	$y = \neg a$ NOT	<table border="1"> <thead> <tr> <th>a</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	a	y	0	1	1	0	Логическое отрицание делает истинное высказывание ложным и, наоборот, ложное - истинным: $\bar{0} = 1 \quad \bar{1} = 0$									
a	y																	
0	1																	
1	0																	
Логическое сложение; Операция ИЛИ; Дизъюнкция	$y = a \vee b$ OR	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	y	0	0	0	0	1	1	1	0	1	1	1	1	Результат логического сложения истинен тогда, когда истинно хотя бы одно слагаемое: $0 \vee 1 = 1 \quad 1 \vee 0 = 1 \quad 1 \vee 1 = 1$ Но: $0 \vee 0 = 0$
a	b	y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Логическое умножение; Операция И; Конъюнкция	$y = a \wedge b$ AND	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	a	b	y	0	0	0	0	1	0	1	0	0	1	1	1	Результат логического умножения истинен тогда, когда истинны все множители: $1 \wedge 1 = 1$ Но: $0 \wedge 0 = 0 \quad 0 \wedge 1 = 0 \quad 1 \wedge 0 = 0$
a	b	y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

Основные логические законы

1. Операции с константами: $A \vee 0 = A \quad A \vee 1 = 1 \quad A \wedge 0 = 0 \quad A \wedge 1 = A$
2. Операции переменной со своей инверсией: $A \vee \bar{A} = 1 \quad A \wedge \bar{A} = 0$
3. Закон двойного отрицания: $\bar{\bar{A}} = A$
4. Законы идемпотенции: $A \vee A = A \quad A \wedge A = A$
5. Правила де Моргана: $\overline{A \vee B} = \bar{A} \wedge \bar{B} \quad \overline{A \wedge B} = \bar{A} \vee \bar{B}$

Примеры упрощения логических выражений

➤ $(\bar{A} \vee B) \wedge A = \bar{A} \wedge A \vee B \wedge A = 0 \vee B \wedge A = B \wedge A$

Последовательность действий: раскрытие скобок + правило 2 + правило 1.

➤ $\overline{\bar{A} \vee \bar{B}} = \bar{\bar{A}} \wedge \bar{\bar{B}} = A \wedge B$ Последовательность действий: правило 5 + правило 3.

➤ $A \wedge B \vee A \wedge B \wedge C \vee A \wedge C \wedge D = A \wedge (B \vee B \wedge C \vee C \wedge D) =$
 $= A \wedge (B \wedge (1 \vee C) \vee C \wedge D) = A \wedge (B \vee C \wedge D)$. Последовательность действий: дважды выносится общий множитель за скобки + правило 1.

Пример составления таблицы истинности логического выражения

Пусть требуется решить логическое выражение $\overline{A \wedge B} \vee A \wedge \bar{B}$, то есть найти его значения при всех возможных комбинациях переменных A и B.

Переменные		Промежуточные значения				Результат
A	B	$A \wedge B$	$\overline{A \wedge B}$	\bar{B}	$A \wedge \bar{B}$	$\overline{A \wedge B} \vee A \wedge \bar{B}$
0	0	0	1	1	0	1
0	1	0	1	0	0	1
1	0	0	1	1	1	1
1	1	1	0	0	0	0

Более рациональным решением здесь можно признать следующее: сначала упростить данное выражение, а уже затем строить таблицу истинности.

$$\overline{A \wedge B} \vee A \wedge \bar{B} = \bar{A} \vee \bar{B} \vee A \wedge \bar{B} = \bar{A} \vee \bar{B} \wedge (1 \vee A) = \bar{A} \vee \bar{B} \wedge 1 = \bar{A} \vee \bar{B}$$

9.2 Pascal: логические операции

- В условном операторе
- **IF S THEN A ELSE B**
- и в операторах цикла
- **WHILE S DO A**
- **REPEAT A UNTIL S**
- условие S – это логическое выражение, следовательно:
- может принимать всего два значения: true (истина) и false (ложь);
- в нем могут применяться логические операции (and, or, not).

Найти максимальное из трех вводимых неравных целых чисел.

```
uses wincrt;
var a, b, c, max: integer;
begin
  write ('1-е число = '); read (a);
  write ('2-е число = '); read (b);
  write ('3-е число = '); read (c);
  if (a>b) and (a>c) then max:= a;
  if (b>a) and (b>c) then max:= b;
  if (c>a) and (c>b) then max:= c;
  write ('максимальное = ', max);
end.
```