

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ КРАСНОЯРСКОГО КРАЯ
КРАЕВОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«НОРИЛЬСКИЙ ТЕХНИКУМ ПРОМЫШЛЕННЫХ
ТЕХНОЛОГИЙ И СЕРВИСА»**

**СБОНИК ПРАКТИЧЕСКИХ РАБОТ
СПЕЦИАЛЬНОСТИ 09.02.07 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И
ПРОГРАММИРОВАНИЕ» ДЛЯ ОБУЧАЮЩИХСЯ 2 КУРСА
ПО ДИСЦИПЛИНЕ ОП. 13 ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

Автор работы:

Сафронова Людмила Александровна, преподаватель

Норильск
2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ПРАКТИЧЕСКАЯ РАБОТА 1. ИНФОРМАЦИОННЫЕ РЕСУРСЫ ОБЩЕСТВА.	6
ПРАКТИЧЕСКАЯ РАБОТА № 2. ОБРАЗОВАТЕЛЬНЫЕ ИНФОРМАЦИОННЫЕ РЕСУРСЫ. РАБОТА С НИМИ.....	9
ПРАКТИЧЕСКАЯ РАБОТА № 3. «АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ ОСНОВЫ РАБОТЫ КОМПЬЮТЕРА. АЛГОРИТМЫ И СПОСОБЫ ИХ ОПИСАНИЯ».....	12
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4. АЛГЕБРА ЛОГИКИ. ЗАКОНЫ ЛОГИКИ.	19
ПРАКТИЧЕСКАЯ РАБОТА 5. РЕШЕНИЕ ЛОГИЧЕСКИХ ЗАДАЧ.....	23
ПРАКТИЧЕСКАЯ РАБОТА 6. ПЕРЕХОД ОТ НЕФОРМАЛЬНОГО ОПИСАНИЯ К ФОРМАЛЬНОМУ.....	28
ПРАКТИЧЕСКАЯ РАБОТА 7. ЛИНЕЙНЫЕ АЛГОРИТМЫ, ВЕТВЛЕНИЕ И ЦИКЛЫ.....	33
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 8. СОЗДАНИЕ АРХИВА ДАННЫХ И РАБОТА С НИМ.....	43
ПРАКТИЧЕСКАЯ РАБОТА 9. ПОИСК ИНФОРМАЦИИ, ХРАНЯЩЕЙСЯ НА КОМПЬЮТЕРЕ. ПРОГРАММНЫЕ ПОИСКОВЫЕ СЕРВИСЫ.	48
ПРАКТИЧЕСКАЯ РАБОТА 10. ПОИСК ИНФОРМАЦИИ НА ГОСУДАРСТВЕННЫХ ОБРАЗОВАТЕЛЬНЫХ ПОРТАЛАХ.	55
ПРАКТИЧЕСКАЯ РАБОТА 11. «СОЗДАНИЕ ЯЩИКА ЭЛ. ПОЧТЫ. ФОРМИРОВАНИЕ АДРЕСНОЙ КНИГИ»	57
ПРАКТИЧЕСКАЯ РАБОТА 12. ФАЙЛ КАК ЕДИНИЦА ХРАНЕНИЯ ИНФОРМАЦИИ НА КОМПЬЮТЕРЕ. АТТРИБУТЫ ФАЙЛА И ЕГО ОБЪЕМ. УЧЕТ ОБЪЕМОВ ФАЙЛОВ ПРИ ИХ ХРАНЕНИИ, ПЕРЕДАЧЕ.	69
СПИСОК ЛИТЕРАТУРЫ	78

ВВЕДЕНИЕ

Сборник практических работ (далее СБОРНИК) предназначен для проведения практических занятий по программе учебной дисциплины ОП 13. Информатика и программирование, по специальности 09.02.07 Информационные системы и программирование

Данный СБОРНИК входит в комплект контрольно-измерительных материалов по учебной дисциплине ОП 13. Информатика и программирование.

СБОРНИК содержит методические указания для обучающихся по выполнению работ на компьютере. Может быть использован обучающимися при изучении дисциплины Информатика и программирование, при подготовке к дифференцированному зачету по данной учебной дисциплине, а также в дополнительном профессиональном образовании (в программах повышения квалификации и переподготовки), где необходимы знания и умения использования Информатики и программирования.

Выполнение практических работ должно способствовать решению следующих методических задач, реализуемых в ходе обучения:

- формировать умения и навыки критического мышления в условиях работы с большими объемами информации;
- формировать навыки самостоятельной работы с учебным материалом с использованием информационных технологий;
- формировать навыки самообразования обучающихся;
- формировать навыки работы в команде;
- развивать умение формулировать задачу и продуктивно решить ее в коллективе;
- формировать навыки самоконтроля;

При оценивании выполнения практических работ использован критериальный уровень оценки знаний.

Каждая практическая работа по курсу содержит: цель выполнения практической работы, задание на выполнение работы, используемые пособия и материалы, основные сведения, порядок выполнения работы. Требования к знаниям и умениям при выполнении практических работ

В результате освоения (выполнения) практических работ, предусмотренных программой по данной специальности, обучающийся **должен уметь:**

- оценивать достоверность информации, сопоставляя различные источники;
- распознавать информационные процессы в различных системах;
- использовать готовые информационные модели, оценивать их соответствие реальному объекту и целям моделирования;
- осуществлять выбор способа представления информации в соответствии с поставленной задачей;
- иллюстрировать учебные работы с использованием средств информационных

- технологий;
- просматривать, создавать, редактировать, сохранять записи в базах данных;
- осуществлять поиск информации в базах данных, компьютерных сетях и пр.;
- представлять числовую информацию различными способами (таблица, массив, график, диаграмма и пр.);
- соблюдать правила техники безопасности и гигиенические рекомендации при использовании средств ИКТ.

должен знать:

- различные подходы к определению понятия «информация»;
- методы измерения количества информации: вероятностный и алфавитный.

Знать единицы измерения информации;

- назначение наиболее распространенных средств автоматизации информационной деятельности (текстовых редакторов, текстовых процессоров, графических редакторов, электронных таблиц, баз данных, компьютерных сетей);
- назначение и виды информационных моделей, описывающих реальные объекты

или процессы;

- использование алгоритма как способа автоматизации деятельности;
- назначение и функции операционных систем.

Правила выполнения практических работ:

1. Для выполнения практических работ обучающемуся понадобятся: рабочее место для проведения работы (компьютер), конспекты, тетрадь для практических работ.

2. Прежде чем приступить к выполнению задания, необходимо прочитать рекомендации к выполнению работы; ознакомиться с перечнем рекомендуемой литературы, повторить теоретический материал, относящийся к теме работы.

3. Обучающийся должен выполнить практическую работу самостоятельно (или в группе, если это предусмотрено заданием).

4. Каждый обучающийся после выполнения работы должен представить отчет о проделанной работе с анализом полученных результатов и выводом по работе.

5. Если возникают затруднения в процессе работы или при оформлении отчета, необходимо обратиться к преподавателю.

6. Отчет о проделанной работе следует делать в тетради для практических работ.

7. Содержание отчета указано в описании практической работе.

8. Если обучающийся не выполнил практическую работу или часть работы, то он может выполнить работу или оставшуюся часть во внеурочное время, согласованное с преподавателем.

9. Оценку по практической работе обучающийся получает с учетом срока выполнения работы, если:

- работа выполнена правильно и в полном объеме;
- сделан анализ проделанной работы и вывод по результатам работы;
- обучающийся может пояснить выполнение любого этапа работы;
- отчет выполнен в соответствии с требованиями к выполнению работы.

Критерии оценки:

Оценка 5(отлично) выставляется, если:

- работа выполнена в урочное время, правильно и в полном объеме; сделан анализ проделанной работы и вывод по результатам работы;
- обучающийся может пояснить выполнение любого этапа работы;
- отчет выполнен в соответствии с требованиями к выполнению работы.

Оценка 4(хорошо) выставляется, если:

- работа выполнена в урочное время; не выполнено одно из заданий;
- правильно и в полном объеме; сделан анализ проделанной работы и вывод по результатам работы;
- обучающийся может пояснить выполнение любого, выполненного им, этапа работы;
- отчет выполнен в соответствии с требованиями к выполнению работы.

Оценка 3 (удовлетворительно) выставляется, если:

- работа выполнена во внеурочное время; не выполнено 50 % работы (в зависимости от количества заданий в работе); технологически неправильно; не сделаны анализ проделанной работы и вывод по результатам работы;
- обучающийся не может пояснить выполнение любого, выполненного им, этапа работы;

отчет выполнен не в соответствии с требованиями к выполнению работы.

Практическая работа 1. Информационные ресурсы общества.

Цель работы: научиться пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью.

Оборудование (приборы, материалы, дидактическое обеспечение): методические рекомендации к выполнению работы; задание и инструкционная карта для проведения практического занятия

Компьютерные программы: Web-браузер ...

Содержание работы.

Основные понятия.

1 "...Информационный ресурс - совокупность отдельных документов, массивов документов, обычно структурированных в базы данных и используемых определенной информационной системой..."

2 Под образовательными информационными ресурсами понимают текстовую, графическую и мультимедийную информацию, а также исполняемые программы (дистрибутивы), то есть электронные ресурсы, созданные специально для использования в процессе обучения на определенной ступени образования и для определённой предметной области.

3 К образовательным электронным ресурсам можно отнести:

- учебные материалы (электронные учебники, учебные пособия, рефераты, дипломы),
- учебно-методические материалы (электронные методики, учебные программы),
- научно-методические (диссертации, кандидатские работы),
- дополнительные текстовые и иллюстративные материалы (лабораторные работы, лекции),
- системы тестирования (тесты – электронная проверка знаний),
- электронные полнотекстовые библиотеки;
- электронные периодические издания сферы образования;
- электронные оглавления и аннотации статей периодических изданий сферы образования,
- электронные архивы выпусков.

4 При работе с образовательными ресурсами появляются такие понятия, как субъект и объект этих ресурсов. Субъекты информационной деятельности классифицируются следующим образом:

- субъект, создающий объекты (все пользователи образовательной системы - преподаватель, студент);
- субъект, использующий объекты (все пользователи образовательной системы);
- субъект, администрирующий объекты, то есть обеспечивающий среду работы с объектами других субъектов (администраторы сети);
- субъект, контролирующий использование объектов субъектами (инженеры).

Задания

Задание 1 Привести примеры:

- 1 достоверной, но необъективной информации;
- 2 объективной, но недостоверной информации;
- 3 полной, достоверной, но бесполезной информации;
- 4 неактуальной информации;
- 5 актуальной, но непонятной информации.

Задание 2 Познакомиться с каталогом образовательных ресурсов Интернет. Охарактеризовать любой раздел.

Задание 3 С помощью Универсального справочника-энциклопедии найдите ответы на вопросы по вариантам.

Порядок выполнения:

Задание 1

- 1 Достоверная, но необъективная информация – это ...

Задание 2

- 1 Загрузим поисковый сайт Интернет ... (название сайта).
- 2 В строке поиска введем фразу «каталог образовательных ресурсов Интернет».
- 3 Разделы образовательных ресурсов сети Интернет включают в себя ... (перечислите).
- 4 Раздел ... содержит ... (Охарактеризуйте любой по выбору).

Задание 3

- 1 WWW – это ...

...

Задания к практической работе.

Задание 3

Варианты с нечетным номером (ПК):

- 1 Что такое WWW?
- 2 Кто родоначальник понятия информации?
- 3 Кто такой Норберт Виннер? Годы его жизни. Место работы.
- 4 Дата первых Олимпийских игр.
- 5 Укажите смертельный уровень звука.
- 6 Какова температура плавления ртути?
- 7 Какова масса Земли?
- 8 Укажите годы правления Екатерины I.
- 9 Укажите годы правления Ивана IV.
- 10 Когда была Троянская война?

Варианты с четным номером (ПК):

- 1 Когда отмечают Всемирный день информации?
- 2 Кто разработчик первого компьютера?
- 3 Кто такой Джон Фон Нейман? Годы его жизни. Место работы.
- 4 Укажите время утверждения григорианского календаря.
- 5 Каков диаметр атома?
- 6 Какова температура кипения железа?
- 7 Укажите скорость обращения Луны вокруг Земли?
- 8 Укажите годы правления Ивана I.

9 Укажите годы правления Хрущева Н.С.

10 В каком году был изобретен первый деревянный велосипед?

Порядок выполнения задания, методические указания: -

ознакомиться с теоретическими положениями по данной теме; - выполнить задания практической работы; - сформулировать вывод и записать в тетради.

Содержание отчета:

- отчет по практической работе должен содержать:
- основные определения;
- рассуждения по выполнению заданий;
- вывод по работе

Контрольные вопросы:

- 1 Что такое информационные ресурсы?
- 2 Что такое образовательные информационные ресурсы?
- 3 Что относится к образовательным информационным ресурсам?
- 4 Каковы субъекты и объекты образовательных информационных ресурсов?

Практическая работа № 2. Образовательные информационные ресурсы. Работа с ними.

Цель: научиться пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью.

Порядок выполнения

Понятие «информационного ресурса общества» (ИРО) является одним из ключевых понятий социальной информатики. Широкое использование этого понятия началось после выхода в 1984 году книги Громова Г.Р. «Национальные информационные ресурсы: проблемы промышленной эксплуатации».

Понятие ИРО, накопленных в обществе знаний, может быть рассмотрено в узком и широком смысле слова.

ИРО в узком смысле слова – это знания, уже готовые для целесообразного социального использования, то есть отчужденные от носителей и материализованные знания.

ИРО в широком смысле слова включают в себя все отчужденные от носителей и включенные в информационный обмен знания, существующие как в устной, так и в материализованной форме.

Что же касается *информационных ресурсов*, то это понятие является сравнительно новым. Оно еще только начинает входить в жизнь современного общества, хотя в последние годы становится все более употребительным не только в научной литературе, но и в общественно-политической деятельности. Причиной этого, безусловно, является глобальная информатизация общества, в котором все больше начинает осознаваться особо важная роль информации и научных знаний.

Для *классификации информационных ресурсов* могут быть использованы следующие их наиболее важные параметры:

- тематика хранящейся в них информации;
- форма собственности – государственная (федеральная, субъекта федерации, муниципальная), общественных организаций, акционерная, частная;
- доступность информации – открытая, закрытая, конфиденциальная;
- принадлежность к определенной информационной системе – библиотечной, – архивной, научно-технической;
- источник информации – официальная информация, публикации в СМИ, статистическая отчетность, результаты социологических исследований;
- назначение и характер использования информации – массовое региональное, ведомственное;
- форма представления информации – текстовая, цифровая, графическая, мультимедийная;
- вид носителя информации – бумажный, электронный.

Под образовательными информационными ресурсами мы будем понимать текстовую, графическую и мультимедийную информацию, а также исполняемые программы (дистрибутивы), то есть электронные ресурсы, созданные

специально для использования в процессе обучения на определенной ступени образования и для определенной предметной области.

При работе с образовательными ресурсами появляются такие понятия, как *субъект* и *объект* этих ресурсов. Классификацию субъектов информационной деятельности произведем следующим образом:

- субъект, создающий объекты (все пользователи образовательной системы- преподаватель, студент);
- субъект, использующий объекты (все пользователи образовательной системы);
- субъект, администрирующий объекты, то есть обеспечивающий среду работы с объектами других субъектов (администраторы сети);
- субъект, контролирующий использование объектов субъектами (инженеры).

К образовательным электронным ресурсам можно отнести:

- учебные материалы (электронные учебники, учебные пособия, рефераты, дипломы),
- учебно-методические материалы (электронные методики, учебные программы),
- научно-методические (диссертации, кандидатские работы),
- дополнительные текстовые и иллюстративные материалы (лабораторные работы, лекции),
- системы тестирования (тесты– электронная проверка знаний),
- электронные полнотекстовые библиотеки;
- электронные периодические издания сферы образования;
- электронные оглавления и аннотации статей периодических изданий сферы образования,
- электронные архивы выпусков.

Задание №1. Познакомиться с сайтом образовательных ресурсов.

Войти в поисковый сервис в Интернете www.yandex.ru и сформировать запрос:

В строку адреса вписать <http://school-collection.edu.ru/> - это коллекция цифровых образовательных ресурсов для учебных заведений нашей страны. Ознакомиться с этой коллекцией.

Найти размещенные там электронные учебники и задания по предмету «Информатика»

Задание №2. Познакомиться с ресурсами сайта Российской государственной библиотеки.

Найти нужную информацию.

Адрес Российской государственной электронной библиотеки
<http://www.rsl.ru>

Войти по электронному адресу Российской государственной библиотеки на главную страницу и ознакомиться с ним. Найти нужное произведение в электронном виде, изученное на 1 курсе по программе предмета литература.

В программе Word составьте таблицу ссылок на сайты библиотек г. Брянска.

Задание №3. Познакомиться с сайтом образовательных ресурсов.

Раздел СПО

Используя сайт «Единое окно» (<http://windows.edu.ru>), составить список ссылок на ресурсы www.fcior.edu.ru. (разделы СПО), имеющих непосредственное отношение к подготовке по вашей будущей профессии.

Воспользоваться также подборкой сайтов по профориентации

<http://www.edu.ru/abitur/act.11/index.php>

<http://pedsovet.org/>

Задание №4. Составьте таблицу ссылок на сайты библиотек региона, в котором вы живете, используя Интернет.

Подберите коллекцию работ любимого художника, используя электронную экспозицию на сайте музея Третьяковской галереи (<http://www.tretyakovgallery.ru/>).

Задание №5.

Воспользуйтесь расширенным поиском в коллекции ФЦИОР и найдите электронные образовательные ресурсы:

- По своему профилю подготовки (специальности или профессии);
- Для использования в изучении других дисциплин;
- Для подготовки к контрольной работе по дисциплинам по профилю обучения.

Контрольные вопросы:

1. Что Вы понимаете под информационными ресурсами?
2. Перечислите параметры для классификации информационных ресурсов.
3. Что понимают под образовательными информационными ресурсами?
4. Что можно отнести к образовательным электронным ресурсам?

Практическая работа № 3. «Арифметические и логические основы работы компьютера. Алгоритмы и способы их описания»

Цели занятия:

- формировать у студентов понятие форм мышления;
- сформировать у студентов понимание арифметических и логических основ работы компьютера;
- сформировать понятия: логическое высказывание, логические величины, логические операции.
- рассмотреть основные примеры построения алгоритмов

Методические указания

Процессор выполняет арифметические и логические операции над двоичными кодами. Поэтому для получения представления об устройстве компьютера, необходимо познакомиться с основными логическими элементами, лежащими в основе его построения. Для понимания принципа работы таких элементов начнем это знакомство с основных начальных понятий алгебры логики.

Формы мышления

В основе современной логики лежат учения, созданные ещё древнегреческими мыслителями, хотя первые учения о формах и способах мышления возникли в Древнем Китае и Индии. Основоположником формальной логики является Аристотель, который впервые отделил логические формы мышления от его содержания.

Логика — это наука о формах и способах мышления. Это учение о способах рассуждений и доказательств.

Законы мира, сущность предметов, общее о них мы познаём посредством абстрактного мышления. Логика позволяет строить формальные модели окружающего мира, отвлекаясь от содержательной стороны.

Мышление всегда осуществляется через понятия, высказывания и умозаключения.

Понятие — это форма мышления, которая выделяет существенные признаки предмета или класса предметов, позволяющие отличать их от других.

Пример 1

Прямоугольник, проливной дождь, компьютер – это понятия.

Высказывание — это формулировка своего понимания окружающего мира. Высказывание является повествовательным предложением, в котором что-либо утверждается или отрицается.

По поводу высказывания можно сказать, истинно оно или ложно. *Истинным будет высказывание, в котором связь понятий правильно отражает свойства и отношения реальных вещей. Ложным высказывание будет в том случае, когда оно противоречит реальной действительности.*

Пример 2

Истинное высказывание: «Буква «а» — гласная».

Ложное высказывание: «Компьютер был изобретён в середине 19 века».

Умозаключение позволяет на основе известных фактов, выраженных в форме суждений, получить новое знание.

Умозаключение — это форма мышления, с помощью которой из одного или нескольких суждений может быть получено новое суждение (знание или вывод).

Пример 3

Дано высказывание: «Все углы равнобедренного треугольника равны». Получите высказывание «Этот треугольник равносторонний» путём умозаключений.

Доказательство:

- 1 Пусть основанием треугольника является сторона c , тогда $a = b$.
- 2 Т. к. в треугольнике все углы равны, следовательно, основанием может быть любая другая сторона, например a , тогда $b = c$.
- 3 Следовательно, $a = b = c$, треугольник равносторонний.

Логические выражения и операции

Алгебра — это наука об общих операциях, аналогичных сложению и умножению, которые выполняются не только над числами, но и над другими математическими объектами, в том числе и над высказываниями. Такая алгебра называется алгеброй логики.

Алгебра логики — это наука об общих операциях, аналогичных сложению и умножению, которые выполняются над высказываниями. Алгебра логики отвлекается от смысловой содержательности высказываний и принимает во внимание только истинность или ложность высказывания.

Можно определить понятие логической переменной, логической функции и логической операции.

Логическая переменная — это простое высказывание, содержащее только одну мысль. Её символическое обозначение — латинская буква (например, A , B , X , Y и т. д.). Значением логической переменной могут быть только константы ИСТИНА и ЛОЖЬ (1 и 0).

Составное высказывание — логическая функция, которая содержит несколько простых мыслей, соединённых между собой с помощью логических операций. Её символическое обозначение — $F(A, B, \dots)$.

На основе простых высказываний могут быть построены составные высказывания.

Логические операции — логическое действие.

Таблица истинности — таблица, определяющая значение сложного высказывания при всех возможных значениях простых высказываний.

Рассмотрим три базовые логические операции — конъюнкцию, дизъюнкцию и отрицание и дополнительные — импликацию и эквиваленцию.

Пояснение: по ходу изложения материала заполните следующую таблицу:

	Конъюнкция (от лат. conjunctio —	Дизъюнкция (от лат. disjunctio —	Инверсия (от лат. inversion —	Импликаци я (от лат. implicatio	Эквиваленци я (от лат. aeguivalens
--	--	--	-------------------------------------	---------------------------------------	--

	связываю)	различаю)	переворачиваю)	— тесно связываю)	— равноценное)									
Название	Логическое умножение	Логическое сложение	Отрицание	Логическое следование	Логическое равенство									
Обозначение	A & B или A^B	A ∨ B	Ā или ¬A	A→ B A — условие B — следствие	A ≡ B или A ↔ B									
Союз в естественном языке	A и B	A или B	Не A	Если A, то B; Когда A, тогда B; Коль скоро A, то и B; и т. п.	A тогда и только тогда, когда B									
Примеры: A – «Число 10 — чётное»; B — «Число 10 — отрицательное»	«Число 10 чётное и отрицательное» = ЛОЖЬ	«Число 10 — чётное или отрицательное» = ИСТИНА	«Неверно, что число 10 — чётное» = ЛОЖЬ «Неверно, что число 10 отрицательное» = ИСТИНА	«Если число 10 — чётное, то оно является отрицательным» = ЛОЖЬ	«Число 10 — чётное тогда и только тогда, когда отрицательное» = ЛОЖЬ									
Таблица истинности	A	B	A & B	A	B	A → B	A	B	A ≡ B					
	0	0	0	0	1	0	0	0	1					
	0	1	0	0	1	1	0	1	0					
	1	0	0	1	0	0	1	0	0					
	1	1	1	1	1	1	1	1	1					
	<u>Вывод:</u> результат будет истинным тогда и только тогда, когда оба исходных высказывания истинны			<u>Вывод:</u> результат будет ложным тогда и только тогда, когда оба исходных высказывания ложны, и истинным в остальных			<u>Вывод:</u> результат будет ложным, если исходное выражение истинно, и наоборот			<u>Вывод:</u> результат будет ложным тогда и только тогда, когда из истинности основания (A) следует ложное следствие			<u>Вывод:</u> результат будет истинным тогда и только тогда, когда оба высказывания одновременно либо ложны, либо истинны	

		случаях		(В)	
--	--	---------	--	-----	--

Если составное высказывание (логическую функцию) выразить в виде формулы, в которую войдут логические переменные и знаки логических операций, то получится логическое выражение, значение которого можно вычислить. Значением логического выражения могут быть только ЛОЖЬ или ИСТИНА.

При составлении логического выражения необходимо учитывать порядок выполнения логических операций, а именно:

- 1 действия в скобках;
- 2 инверсия, конъюнкция, дизъюнкция, импликация, эквиваленция.

Пример 4

Запишите в виде логического выражения следующее высказывание: «Летом Петя поедет в деревню и, если будет хорошая погода, то он пойдёт на рыбалку».

- 1 Проанализируем составное высказывание.

Оно состоит из следующих простых высказываний: «Петя поедет в деревню», «Будет хорошая погода», «Он пойдёт на рыбалку». Обозначим их через логические переменные:

A = Петя поедет в деревню;

B = Будет хорошая погода;

C = Он пойдёт на рыбалку.

1. Запишем высказывание в виде логического выражения, учитывая порядок действий. Если необходимо, расставим скобки:

$$F = A \& (B \rightarrow C).$$

Алгоритм

Алгоритм - точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Название "алгоритм" произошло от латинской формы имени среднеазиатского математика аль-Хорезми - Algorithmi. Алгоритм - одно из основных понятий информатики и математики.

Исполнитель алгоритма - это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Исполнителя характеризуют:

- среда;
- элементарные действия;
- система команд;
- отказы.

Основные свойства алгоритмов следующие:

Понятность для исполнителя - т.е. исполнитель алгоритма должен знать, как его выполнять.

Дискретность (прерывность, раздельность) - т.е. алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).

Определенность - т.е. каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

Результативность (или конечность). Это свойство состоит в том, что алгоритм должен приводить к решению задачи за конечное число шагов.

Массовость. Это означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

На практике наиболее распространены следующие формы представления алгоритмов:

- словесная (записи на естественном языке);
- графическая (изображения из графических символов);
- псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- программная (тексты на языках программирования).

Содержание работы

Задание 1

Есть два простых высказывания:

A – «Число 10 — чётное»;

B — «Волк — травоядное животное».

Составьте из них все возможные составные высказывания и определите их истинность.

Ответ:

$A \& B$	$A \vee B$	$\neg A$	$\neg B$	$A \rightarrow B$	$A \equiv B$
0	1	0	1	0	0

Задание 2

Запишите следующие высказывания в виде логических выражений.

- 1 Число 17 нечётное и двузначное.
- 2 Неверно, что корова — хищное животное.
- 3 На уроке физики ученики выполняли лабораторную работу и сообщали результаты исследований учителю.
- 4 Если число делится на 2, то оно — чётное. Переходи улицу только на зелёный свет.

- 5 На уроке информатики необходимо соблюдать особые правила поведения.
- 6 Если Маша — сестра Саши, то Саша — брат Маши.
- 7 Если компьютер включен, то можно на нём работать.
- 8 Ты можешь купить в магазине продукты, если у тебя есть деньги.
- 9 Компьютер выполняет вычисления, если он включен.

Задание 3

Составьте и запишите истинные сложные высказывания из простых с использованием логических операций.

- 1 Неверно, что $10 Y \geq 5$ и Z
- 2 Z является $\min(Z, Y)$.
- 3 A является $\max(A, B, C)$.
- 4 Любое из чисел X, Y, Z положительно.
- 5 Любое из чисел X, Y, Z отрицательно.
- 6 Хотя бы одно из чисел K, L, M не отрицательно.
- 7 Хотя бы одно из чисел X, Y, Z не меньше 12.
- 8 Все числа X, Y, Z равны 12.
- 9 Если X делится на 9, то X делится на 3.
- 10 Если X делится на 2, то оно чётное.

Задание 4

Найдите значения логических выражений:

- 1 $F = (0 \vee 0) \vee (1 \vee 1)$. (Ответ: 1)
- 2 $F = (1 \vee 1) \vee (1 \vee 0)$. (Ответ: 1)
- 3 $F = (0 \& 0) \& (1 \& 1)$. (Ответ: 0)
- 4 $F = \neg 1 \& (1 \vee 1) \vee (\neg 0 \& 1)$. (Ответ: 1)
- 5 $F = (\neg 1 \vee 1) \& (1 \vee \neg 1) \& (\neg 1 \vee 0)$. (Ответ: 0)

Задание 5

Задача 1

Из двух простых высказываний постройте сложное высказывание, используя логические связки «И», «ИЛИ». Запишите логические высказывания с помощью логических операций и определите их истинность.

- 1 Андрей старше Светы. Наташа старше Светы.
- 2 Один десятый класс идёт на экскурсию в музей. Второй десятый класс идёт в театр.
- 3 На полке стоят учебники. На полке стоят справочники.
- 4 Часть детей — девочки. Часть детей — мальчики.

Задача 2

Для логических выражений сформулируйте составные высказывания на обычном языке:

- 1 $(Y = 1 \text{ и } Y \text{ или } (Y \text{ и } Y = 4))$;
- 2 $(X = Y) \text{ и } (X = Z)$;
- 3 $\text{не } (X) \text{ и } X \leq 10 \text{ или } (Y = 0)$;
- 4 $(0 \text{ и } (X \leq 5) \text{ и } (\text{не } (Y)))$.

Задание 6

1 В программе БЛОКНОТ наберите кратко текст о видах информации. Сохраните этот документ под именем «Примеры построения алгоритмов» в папку со своим именем. Чтобы выполнить данное задание необходимо воспользоваться следующими вопросами:

2 Вопросы к заданию:

а) *Опишите требования к алгоритму:*

- 1 _____
- 2 _____
- 3 _____
- 4 _____

б) *Опишите способы представления алгоритмов*

2 Ответьте на вопросы теста:

1 Что такое алгоритм?

- А) пошаговое выполнение определенных действий;
- В) выполнение запрограммированных действий;
- С) нет правильного ответа.

2 Назовите исполнителей алгоритма?

- А) Человек
- В) техническое устройство
- С) все выше перечисленное.

3 Что называется системой команд исполнителя?

- А) объект или субъект, для управления которым составлен алгоритм;
- В) совокупность команд, которые могут быть выполнены исполнителем;
- С) свойство алгоритма.

4 Выберите правильный вариант, в котором перечислены свойства алгоритма.

- А) дискретность, массовость, определенность, результативность
- В) дискретность, массовость, определенность
- С) дискретность, массовость, определенность, результативность, линейность, последовательность.

5 Перечислите способы описания алгоритма.

- А) словесный, графический;
- В) алгоритмический, табличный
- С) все выше перечисленное.

Практическое занятие 4. Алгебра логики. Законы логики.

Цель: научиться строить таблицы истинности логических высказываний и преобразовывать формулы, используя основные законы логики.

Ход работы

Задание 1. В правый столбец таблицы впишите слово «да», если это высказывание, и «нет» - в противном случае:

Предложение	Ответ на вопрос «Это высказывание?»
1. Запиши название обратного алгоритма.	
2. На имя каждого кенгуру пришло разное количество писем, а каждый жираф отправил ровно два письма.	
3. Если выглянет солнце, то станет тепло.	
4. Аня пойдет в кино тогда, когда пойдут Вика и Сергей.	
5. $x^2 + 5x + 3 \geq 0$	
6. По стройке необходимо ходить в каске.	
7. Для сложения трех некомпланарных векторов можно пользоваться так называемым правилом параллелепипеда.	
8. Каждый человек – художник.	
9. Все ананасы приятны на вкус.	
10. Не нарушайте правил дорожного движения!	
11. Завтра я пойду на прогулку с друзьями или в кино, или на хоккей.	
12. Число 15 – не простое.	

Задание 2. Укажите для нижеприведенных высказываний, сложные они или простые:

Высказывание	Тип высказывания
1. Полярная звезда находится в созвездии Малой Медведицы.	
2. Неверно, что Краснодар – столица Казахстана.	
3. Две прямые на плоскости параллельны или пересекаются.	
4. Одна половина класса изучает английский язык, а вторая изучает французский язык.	
5. За площадь боковой поверхности конуса принимается площадь её развертки.	
6. Не любой вектор можно разложить по координатным векторам.	
7. Приставка есть часть слова, и она стоит перед корнем.	

8. Высказывание $A \vee B$ ложно в том и только в том случае, когда одновременно ложны оба высказывания.	
9. Если векторы a, b, c компланарны, а векторы a и b не коллинеарны, то вектор c можно разложить по векторам a и b .	
10. Толя не пришел в школу.	
11. На каникулах мы поедem в Казань или Самару.	
12. Сегодня не будет контрольной по географии.	
13. Квадрат диагонали прямоугольного параллелепипеда равен сумме квадратов трех его измерений.	
14. Исполнителем алгоритмов работы с величинами является компьютер.	

Задание 3. Укажите в нижеприведенных сложных высказываниях связующие слова или союзы и наименование связки:

Высказывание	Связка
1. Он так и не пришел.	
2. Главными потребителями йода являются фармацевтическая и химическая промышленность.	
3. Если для атомов металлов характерны сравнительно большие радиусы и небольшое число электронов на внешнем уровне, то атомам неметаллов свойственны небольшие радиусы атомов и число электронов на внешнем энергетическом уровне от 4 до 8.	
4. Неверно, что Костя учится в 10 классе.	
5. Скоро мы поедem или в Москву, или в Казань, или в Самару.	
6. Костя выиграет в шахматы тогда и только тогда, когда проиграет Саша.	
7. Папа собирается купить либо «Рено», либо «Форд», либо «Тойоте».	
8. $Zn(OH)_2$ не является щелочью.	
9. $x^2 + y^2 = r^2$ и $\sin^2(x) + \cos^2(x) = 1$.	
10. Германия и её союзники вынуждены были просить о мире.	
11. Не каждый человек ответственен.	
12. Если при каком-либо x или y некоторые из выражений не имеют смысла, то вывести сообщение об этом и сравнить значения только тех, которые имеют смысл.	

Задание 4. Составьте минимум 3 сложных высказываний, используя нижеприведенные простые высказывания и логические связки.

- 1) Завтра будет дождь.
- 2) Мы вылетаем в Париж.
- 3) Мы хотим пойти за грибами.
- 4) Миша приглашает нас на рыбалку.
- 5) Саша приглашает нас на день рождения.

- 6) Алюминий + углерод = Al_4C_3 .
- 7) Я знаю химию хорошо.
- 8) $5 \cdot 5 = 625$.
- 9) Он скоро окончит школу.
- 10) Он будет учиться в Москве.
- 11) Он – победитель олимпиады по математике.
- 12) В промышленности фтор получают электролизом расплавов.

Задание 5. Вычислить значение логической функции:

- 1) $f(c, d) = (\bar{c} \vee \bar{d}) \& (c \vee \bar{d})$, если $c=0, d=1$.
- 2) $s(a, b) = \bar{a} \& \bar{b} \vee a \& \bar{b}$, если $a=0, b=1$.
- 3) $p(a, b) = \bar{a} \& (\bar{b} \vee a)$, если $a=1, b=0$.
- 4) $f(a, b) = \overline{a \vee \bar{b}}$, если $a=1, b=0$.
- 5) $f(c, d) = \overline{(c \vee d)} \& \bar{c} \vee d$, если $c=0, d=1$.
- 6) $f(x, y, z) = \overline{(x \vee \bar{y})} \& z$, если $y=0, x=1, z=0$.
- 7) $f(a, b, c) = \overline{a \vee b} \& (\bar{c} \vee \bar{b})$, если $a=1, b=0, c=0$.
- 8) $f(a, b, c) = a \vee \bar{b} \& \bar{c}$, если $a=0, b=1, c=1$.
- 9) $f(a, b, c) = \bar{a} \& (\bar{b} \vee a \vee c)$, если $a=0, b=1, c=1$.
- 10) $f(a, b, c) = \overline{(\bar{a} \vee \bar{b})} \vee \bar{b} \& c$, если $a=1, b=0, c=0$.

Задание 6. При заданных значениях переменных $a=0, b=1, c=0$ определите истинные функции и расшифруйте слово.

T	$f = a \vee b \& \bar{c}$	Г	$f = \overline{\bar{b} \vee c} \& \bar{a}$
Б	$f = \bar{a} \& \bar{b} \& c$	М	$f = b \& c \& (a \vee b)$
P	$f = \overline{a \vee b} \vee \bar{c}$	Г	$f = \overline{(a \vee c)} \& b$
У	$f = \overline{a \vee b \vee c}$	Е	$f = \bar{a} \& (b \vee c)$
K	$f = \bar{a} \& b \& c$	A	$f = a \vee c \vee \bar{b}$
И	$f = (\bar{a} \vee c) \& b$	Л	$f = b \& a \vee b \& c$
O	$f = \bar{a} \& (\bar{b} \vee c)$	P	$f = \bar{c} \& \bar{a} \vee b \& \bar{c}$

Задание 7. Определить истинные функции при заданных значениях переменных $a=1, b=1, c=0$, расшифруйте слово.

Б	$(a \& b) \rightarrow c$	C	$\overline{a \vee c \vee \bar{b}} \leftrightarrow \bar{a} \& \bar{c} \vee b$
П	$(a \vee c) \rightarrow (c \vee b)$	T	$\bar{a} \& (a \vee b) \leftrightarrow \bar{a} \vee b$
И	$\overline{(a \vee b)} \rightarrow (a \& \bar{c})$	E	$\bar{a} \& \bar{b} \& \bar{c} \leftrightarrow \overline{(a \vee b)} \& c$
A	$(c \& a) \rightarrow c$	K	$a \& b \& \bar{c} \leftrightarrow (a \vee c) \& b$
K	$c \rightarrow (a \& b)$	Л	$\bar{c} \& b \vee a \& \bar{b} \leftrightarrow \bar{b} \vee \bar{c} \vee \bar{a}$
Й	$c \vee a \& b \leftrightarrow \bar{a} \vee \bar{c} \vee \bar{b}$	M	$(a \& c) \leftrightarrow b$
E	$(a \& c) \rightarrow \overline{(a \& b)}$	B	$(a \& c) \leftrightarrow (\bar{b} \vee \bar{a})$

Задание 8. Построить таблицу истинности для логической функции:

- 1) $f(a, b) = \overline{(a \& \bar{b} \vee a)} \vee b;$
- 2) $f(x, y) = \overline{(x \vee y) \& \bar{y} \vee \bar{x}};$
- 3) $f(x, y) = \overline{x \& (y \vee x)};$
- 4) $f(x, y) = \overline{(x \vee y) \& x \vee y};$
- 5) $f(a, b, c) = a \& (b \vee c) \& \bar{b};$
- 6) $f(x, y, z) = x \rightarrow (y \vee z)$
- 7) $f(x, y, z) = \overline{(x \& y) \vee z \rightarrow (\bar{y} \& \bar{z})}$
- 8) $f(x, y, z) = \overline{(\bar{x} \& \bar{y}) \rightarrow (x \& y) \vee \bar{z}}$
- 9) $f(x, y, z) = \overline{(x \rightarrow y) \rightarrow z \& \bar{x}}$
- 10) $f(x, y, z) = \overline{(x \vee y) \rightarrow (\bar{x} \& y)}$

Задание 9. Упростить логическую функцию, используя законы булевой алгебры.

- 1) $f(x, y) = \overline{(\bar{x} \& \bar{y}) \vee (\bar{x} \& y)};$
- 2) $f(x, y) = \overline{(x \& y \vee x \& \bar{y}) \& (\bar{x} \& \bar{y})};$
- 3) $f(x, y) = y \vee x \& \bar{x} \vee x \& (y \vee y);$
- 4) $f(x, y) = \overline{\bar{x} \& \bar{y} \vee \bar{x} \& x \vee \bar{x} \vee y};$
- 5) $f(x, y, z) = \overline{((x \& \bar{y}) \vee \bar{z}) \& x \& \bar{y}};$

Практическая работа 5. Решение логических задач

Цель: Научиться решать логические задачи

Интерпретатор Пролога можно заставить решать логические задачи, что недоступно большинству процедурных языков.

Многие логические задачи связаны с рассмотрением нескольких конечных множеств с одинаковым количеством элементов, между которыми устанавливается взаимно-однозначное соответствие. На языке Пролог эти множества можно описывать как базы данных, а зависимости между объектами устанавливать с помощью правил.

Пример простой логической задачи (два измерения)

Условия задачи: В автомобильных гонках три первых места заняли Алеша, Петя и Коля. Какое место занял каждый из них, если Петя занял не первое и не третье место, а Коля - не третье?

Традиционным способом задача решается заполнением таблицы. По условию задачи Петя занял не первое и не третье место, а Коля - не третье. Это позволяет поставить символ '-' в соответствующих клетках. Между множеством имен участников гонки и множеством мест должно быть установлено взаимно-однозначное соответствие. Поэтому определяем занятое место сначала у Пети, затем у Коли и, наконец, у Алеши. В соответствующих клетках проставляем знак '+'. В каждой строке и каждом столбце должен быть только один такой знак.

Имя	I место	II место	III место
Петя	-	+	-
Коля	+	-	-
Алеша	-	-	+

Из таблицы следует, что Алеша занял третье место, Петя – второе, Коля – первое.

На языке Пролог структура программы представлена на рис. 5.1. В ответ Пролог выдаст имена ребят и занятые ими места.

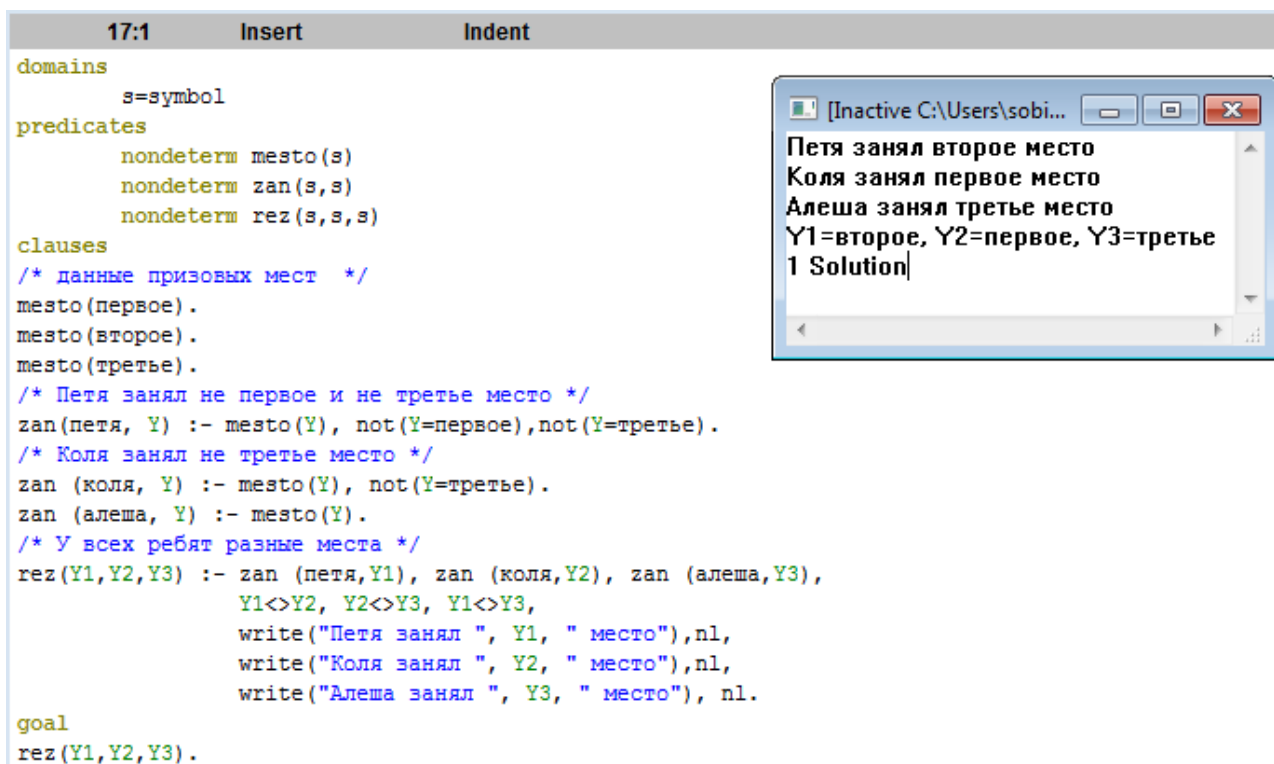


Рис. 5.1. Программа решения простой логической задачи

Практикум 5-1

- : Витя, Юра и Миша сидели на скамейке. В каком порядке они сидели, если известно, что Миша сидел слева от Юры, а Витя слева от Миши.

Практикум 5-2

- : Трое ребят вышли гулять с собакой, кошкой и хомячком. Известно, что Петя не любит кошек и живет в одном подъезде с хозяйкой хомячка. Лена дружит с Таней, гуляющей с кошкой. Определить, с каким животным гулял каждый из детей.

Пример задачи (три измерения)

Условия задачи: Четверо ребят: Лева, Коля, Миша и Петя решили позавтракать в разных местах: один – в кафе, другой - в столовой, третий – в закусочной, а четвертый – в буфете.

После завтрака все собрались вместе. Разговор зашел о том, кто как позавтракал.

Выяснилось, что все они пили разные напитки, так как в каждом из тех мест, где они завтракали, оказалось в наличие только по одному напитку: в одном месте – только кофе, в другом – только молоко, в третьем – только кефир, а в четвертом только чай.

В буфете, например, было только молоко, а в столовой не было кефира.

Петя рассказал, что он был в столовой, но пил там не чай.

Лева рассказал, что он пил кефир.

А Миша сказал, что он не был ни в закусочной, ни в буфете.

Кто где завтракал и что при этом пил?

При решении этой задачи удобно использовать списки и предикаты работы со списками (рис.5.2).

```

22:9      Insert      Indent      Modified
domains
    s=symbol
    kto_gde=zavtrak(s,s,s)/*элемент списка - структура*/
    spisok=kto_gde*      /*описание списка*/
predicates
    nondeterm pl(s)
    nondeterm napitok(s)
    nondeterm razn_pl(s,s,s,s)
    nondeterm razn_nap(s,s,s,s)
    nondeterm otvet(spisok)
    nondeterm member(kto_gde,spisok)
    nondeterm print_list(spisok)
    nondeterm rez

clauses
% Перечисление мест
pl(кафе). pl(закусочная). pl(столовая). pl(буфет).
% Перечисление напитков
napitok(кофе). napitok(молоко). napitok(чай). napitok(кефир).
% Места должны быть разными
razn_pl(M1,M2,M3,M4):-pl(M1),pl(M2),pl(M3),pl(M4),
M1<>M2,M1<>M3,M1<>M4,M2<>M3,M2<>M4,M3<>M4.
% Напитки должны быть разными
razn_nap(M1,M2,M3,M4):-napitok(M1),napitok(M2),napitok(M3),
napitok(M4),M1<>M2,M1<>M3,M1<>M4,M2<>M3,M2<>M4,M3<>M4.
%Петя рассказал, что он был в столовой, но пил там не чай.
%Лева рассказал, что он пил кефир.
%А Миша сказал, что он не был ни в закусочной, ни в буфете.
otvet([zavtrak("Петя",M1,N1),
      zavtrak("Лева",M2,N2),
      zavtrak("Миша",M3,N3),
      zavtrak("Коля",M4,N4)]) :-
      razn_pl(M1,M2,M3,M4),M1="столовая", M3<>"закусочная",M3<>"буфет",
      razn_nap(N1,N2,N3,N4),N2="кефир",N1<>"чай".

member(X,[X|_]):-!.
member(X,[_|T]):-member(X,T).

print_list([ ]).
print_list([X|Y]) :- write(X), nl, print_list(Y).
%B буфете было только молоко
rez:-otvet(L),member(zavtrak(_,буфет,молоко),L),print_list(L) .

goal
rez.

```

Рис. 5.2. Программа решения логической задачи 2

Результат выполнения программы представлен на рис. 5.3

```

zavtrak("Петя","столовая","кофе")
zavtrak("Лева","закусочная","кефир")
zavtrak("Миша","кафе","чай")
zavtrak("Коля","буфет","молоко")
yes

```

Рис. 5.3. Результат решения логической задачи 2

Практикум 5-3

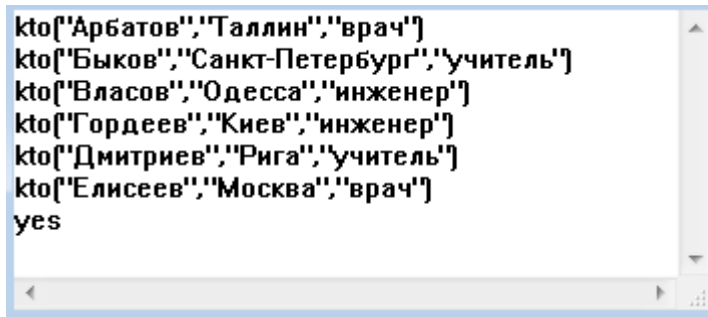
: В пансионате отдыхали Арбатов, Быков, Власов, Гордеев, Дмитриев, Елисеев. Все эти люди приехали из разных городов: Москвы, Санкт-Петербурга, Киева, Одессы, Риги и Таллина.

Кроме того, известно, что:

- Арбатов и москвич - врачи;
- Дмитриев и петербуржец - учителя;
- Власов и киевлянин - инженеры;
- Быков и Елисеев никогда не были в Киеве;
- Рижанин старше Арбатова, а житель Таллина старше Власова;
- Быков и москвич никогда не были в Одессе, а Власов и рижанин – в Таллине.

Выяснить, кто в каком городе живет и у кого какая профессия.

Ответ для проверки представлен на рис. 5.4



```
kto['Арбатов','Таллин','врач']  
кто['Быков','Санкт-Петербург','учитель']  
кто['Власов','Одесса','инженер']  
кто['Гордеев','Киев','инженер']  
кто['Дмитриев','Рига','учитель']  
кто['Елисеев','Москва','врач']  
yes
```

Рис. 5.4. Ответ к задаче из Практикума 5-2

Пример задачи (альтернативные высказывания)

В спортивных соревнованиях принимали участие пять команд: "Вымпел", "Метеор", "Нептун", "Старт" и "Чайка". Об их итогах соревнования имеется пять высказываний:

- 1). Второе место занял "Вымпел", а "Старт" оказался на третьем.
- 2). Хорошо выступала команда "Нептун", она стала победителем, а "Чайка" вышла на второе место.
- 3). Да нет же, "Чайка" заняла только третье место, а "Нептун"- был последним.
- 4). Первое место по праву завоевал "Старт", а "Метеор" был четвертым.
- 5). Да, "Метеор", действительно, был четвертым, а "Вымпел" был третьим.

Известно, что команды не делили места между собой и что в каждом высказывании одно утверждение правильное, а другое нет.

Как распределились места между командами?

Секция clauses Пролог-программы для решения этой задачи представлена на рис. 5.5.

```

name("Вымпел").name("Метеор").name("Нептун").name("Старт").name("Чайка").
% Команды не делили места между собой
razn(M1,M2,M3,M4,M5):-name(M1),name(M2),name(M3),name(M4),name(M5),
M1<>M2,M1<>M3,M1<>M4,M1<>M5,M2<>M3,M2<>M4,M2<>M5,M3<>M4,M3<>M5,M4<>M5.
% В каждом высказывании одно утверждение правильное, а другое нет
%1). Второе место занял "Вымпел", а "Старт" оказался на третьем.
prav1(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X2="Вымпел",X3<"Старт".
prav1(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X2<"Вымпел",X3="Старт".
%2). Хорошо выступала команда "Нептун", она стала победителем, а "Чайка" вышла на второе место
prav2(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X1="Нептун",X2<"Чайка".
prav2(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X1<"Нептун",X2="Чайка".
%3). Да нет же, "Чайка" заняла только третье место, а "Нептун" – был последним.
prav3(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X5="Нептун",X3<"Чайка".
prav3(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X5<"Нептун",X3="Чайка".
%4). Первое место по праву завоевал "Старт", а "Метеор" был четвертым.
prav4(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X1="Старт",X4<"Метеор".
prav4(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X1<"Старт",X4="Метеор".
%5). Да, "Метеор", действительно, был четвертым, а "Вымпел" был третьим.
prav5(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X3="Вымпел",X4<"Метеор".
prav5(X1,X2,X3,X4,X5):-razn(X1,X2,X3,X4,X5),X3<"Вымпел",X4="Метеор".
otvet(X1,X2,X3,X4,X5):-prav1(X1,X2,X3,X4,X5),prav2(X1,X2,X3,X4,X5),prav3(X1,X2,X3,X4,X5),
prav4(X1,X2,X3,X4,X5),prav5(X1,X2,X3,X4,X5).
goal
otvet(Mesto1,Mesto2,Mesto3,Mesto4,Mesto5).

```

Рис. 5.5. Секция clauses для задачи «Спортивные соревнования»

Практикум 5-4

: Найдите решение для примера рис. 5.5

Практикум 5-5

: Для четырех дежурных (Антипов, Климов, Марков и Лебедев) нужно составить график дежурств по одному с понедельника до четверга с соблюдением следующих условий:

- 1) Если Лебедев не будет дежурить в понедельник, то в понедельник согласен дежурить Климов.
- 2) Если Климов не сможет дежурить ни в понедельник, ни в четверг, то Антипов будет дежурить в понедельник.
- 3) Если Марков не сможет дежурить в четверг, то Климов будет дежурить в среду.
- 4) Если Лебедев придет дежурить во вторник, то Климов не будет дежурить в понедельник.
- 5) Если Антипов не сможет дежурить в понедельник, то Марков не сможет дежурить во вторник.

Каким должен быть график дежурств ?

Самостоятельные задания

1. На школьном вечере танцевали три пары. Юноши, как и девушки, были одеты в костюмы разных цветов: красный, зеленый и синий. Оказавшись в один из моментов рядом с девушкой в зеленом, юноша в красном обратился к ней: "Неправда ли, забавно получается: ни у кого из нас цвет костюма не совпадает с цветом костюма партнера". В костюме какого цвета был юноша, танцевавший в паре с девушкой в красном?

Ответ

Юноша в красном танцует с девушкой в синем,
а девушка в зеленом - с юношей в синем.
девушка в красном танцует с юношей в зеленом

Практическая работа 6. Переход от неформального описания к формальному.

1. Цель работы: изучить среду программирования и структуру программы Turbo Pascal, научиться составлять несложные программы в этой среде программирования, производить тестирование программы.

2. Оборудование, приборы, аппаратура, материалы: персональный компьютер, среда программирования Turbo Pascal.

Краткие теоретические сведения.

Паскаль - язык профессионального программирования, который назван в честь французского математика и философа Блеза Паскаля (1623-1662) и разработан в 1968-1971 гг. Никлаусом Виртом. Первоначально был разработан для обучения, но вскоре стал использоваться для разработки программных средств в профессиональном программировании.

Паскаль популярен среди программистов по следующим причинам:

- Прост для обучения.
- Отражает фундаментальные идеи алгоритмов в легко воспринимаемой форме, то предоставляет программисту средства, помогающие проектировать программы.
- Позволяет четко реализовать идеи структурного программирования и структурной организации данных.
- Использование простых и гибких структур управления: ветвлений, циклов.
- Надежность разрабатываемых программ.

Турбо Паскаль - это система программирования, созданная для повышения качества и скорости разработки программ (80-е гг.). Слово Турбо в названии системы программирования - это отражение торговой марки фирмы-разработчика Borland International (США).

Систему программирования Турбо Паскаль называют интегрированной (integration - объединение отдельных элементов в единое целое) средой программирования, т.к. она включает в себя редактор, компилятор, отладчик, имеет сервисные возможности.

Основные файлы Турбо Паскаля:

- Turbo.exe - исполняемый файл интегрированной среды программирования;
- turbo.hlp - файл, содержащий данные для помощи;
- Turbo.tp - файл конфигурации системы;
- Turbo.tpl - библиотека стандартных модулей, в которых содержатся встроенные процедуры и функции (SYSTEM, CRT, DOS, PRINTER, GRAPH, TURBO3, GRAPH3).

Для запуска интегрированной среды программирования нужно установить текущим каталог с Турбо Паскалем (TP7\BIN) и ввести команду: turbo.exe.

Окно программы содержит полосу меню, область окна и строку статуса.



Для входа в меню можно воспользоваться одним из способов:

- с помощью "мышки";
- с помощью клавиши F10;
- с помощью комбинации Alt+<выделенная буква>. О том, что мы в меню свидетельствует курсор - прямоугольник зеленого цвета.

Интегрированная среда программирования Турбо Паскаль позволяет иметь любое количество открытых окон, но в любой момент времени активным может быть только одно.

Активное окно - это окно с которым вы в настоящий момент работаете.

Общие горячие клавиши:

- F1 - выводит окно подсказки;
- F2 - сохраняет файл активного окна;
- F3 - появление диалогового окна и возможность открыть файл;
- F4 - запускает программу до строки, на которой стоит курсор;
- F5 - масштабирует диалоговое окно;
- F6 - переходит к следующему открытому окну;
- F7 - запускает программу в режиме отладки с заходом внутрь процедур;
- F8 - запускает программу в режиме отладки, минуя вызов процедур;
- F9 - компилирование программы в текущем окне;
- F10 - возвращение в меню.

Команды меню File:

- Open-F3 - открыть существующий файл (при активизации этой опции появляется окно со списком файлов, где можно выбрать необходимый),
- New - создать новый файл (очищает память редактора и переводит в режим создания нового файла, которому присваивается имя

- Noname.pas; имя можно изменить при записи файла на диск),
- Save-F2 - сохранить файл (переписывает файл из памяти редактора на диск),
- Save as - сохранить с новым именем,
- Save all - сохранить все в окнах (записывает содержимое всех окон редактора в соответствующие файлы),
- Change dir - смена каталога (позволяет изменить установленный по умолчанию диск или каталог),
- Print - печать файла,
- Get info - выдача информации о текущем состоянии программы и используемой памяти,
- DOS Shell - выход в DOS без выгрузки из памяти (для возврата ввести команду exit),
- Exit - выход и выгрузка из памяти.

Программы на языке Паскаль имеют блочную структуру:

1. Блок типа PROGRAM - имеет имя, состоящее только из латинских букв и цифр. Его присутствие не обязательно, но рекомендуется записывать для быстрого распознавания нужной программы среди других листингов.
2. Программный блок, состоящий в общем случае из 7 разделов:
3. раздел описания модулей (uses);
- раздел описания меток (label);
- раздел описания констант (const);
- раздел описания типов данных (type);
- раздел описания переменных (var);
- раздел описания процедур и функций;
- раздел описания операторов.

Общая структура программы на языке Паскаль:

```

Program ИМЯ..; {заголовок программы}
Uses ...; {раздел описания модулей}
Var ...; {раздел объявления переменных}
...
Begin {начало исполнительной части программы}
...                               {последовательность
...                               операторов}
End. {конец программы}

```

Пример программы, которая осуществляет сложение двух чисел и выводит сумму на экран:

```

Program Summa;
Uses
Crt; {Подключаем модуль Crt}
Var
  number1, {переменная, в которой будет содержаться первое число}
  number2, {переменная, в которой будет содержаться второе число}

```

```

rezult {переменная, в которой будет содержаться результат}
:integer; {указывает тип целых чисел}
Begin
  ClrScr; {Используем процедуру очистки экрана из модуля Crt}
  Write ('Введите первое число ');
  {Выводим на экран символы, записанные между апострофами}
  Readln (number1);
  {Введенное пользователем число считываем в переменную number1}
  Write ('Введите второе число ');
  {Выводим на экран символы, записанные между апострофами}
  Readln (number2);
  {Введенное пользователем число считываем в переменную number2}
  rezult := number1 + number2;
  {Находим сумму введенных чисел и присваиваем переменной rezult}
  Write ('Сумма чисел ', number1, ' и ', number2, ' равно ', rezult);
  {Выводим на экран строку, содержащую ответ задачи}
  Readln; {Процедура задержки экрана}
End.

```

1. Задание

Задание 1. Изучите внешний вид системы программирования Турбо Паскаль.

Задание 2. Откройте файл, в который Вы запишите программу, выполняющую сложение двух чисел. Для этого нажмите клавишу F10, чтобы выйти в главное меню, затем клавишами перемещения курсора выберите опцию File, а в выпавшем меню команду New.

Найдите в этой программе заголовок, раздел описания переменных, признак начала программы, признак конца программы, тело программы, комментарий.

Ответьте на вопросы:

1. Какое назначение переменных number1, number2, rezult?
2. Что обозначает строка: number1, number2, rezult : integer; ?
3. Если присвоить переменным number1 и number2 соответственно значение 5 и 7, то какую строку выдаст компьютер при исполнении последней процедуры Write? Запишите ее в отчет.
4. Переведите с английского языка слова: Write, Read. Как вы думаете, что должен делать операторы с таким названием?
5. Как вы понимаете запись: readln(number1); ?
6. Чему равно значение переменной rezult после выполнения оператора: rezult := number1 + number2; ?
7. Что делает оператор присваивания в этой программе?

Задание 3. Измените программу, выполненную в задании 2 так, чтобы она находила произведение двух чисел. Сохраните текст программы в файле Proizv.pas. Результат покажите преподавателю.

Задание 4. Измените программу, выполненную в задании 3 так, чтобы

она выполняла расчет площади прямоугольника по его длине и ширине.
Заполните по результатам работы программы таблицу:

№ п/п	Параметры прямоугольника		
	Длина	Ширин а	Площадь
1.	1	10	
2.	7	14	
3.	5	7	
4.	8	8	
5.	4	12	

5. Содержание отчета

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Ответы на контрольные вопросы
5. Вывод по работе.

Практическая работа 7. Линейные алгоритмы, ветвление и циклы.

Цель работы: сформировать умения по составлению алгоритма программы; научить записывать его с помощью блок-схем.

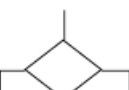
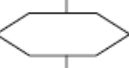
Оборудование, технические и программные средства: персональный компьютер, приложение DiagramDesigner.

Порядок выполнения работы

Теоретические сведения

Предпочтительнее до записи на алгоритмическом языке представить алгоритм в виде блок-схемы. Для построения алгоритма в виде блок-схемы необходимо знать назначения каждого из блоков. В таблице 1. приводятся типы блоков и их назначение.

Таблица 1

№	Блок	Назначение блока
1		Начало или конец блок-схемы
2		Ввод или вывод данных
3		Процесс (в частности вычислительный)
4		условие
6		Цикл с параметром (for)

Основные типы алгоритмов

Алгоритмизация вычислений предполагает решение задачи в виде последовательности действий, т.е. решение, представленное в виде блок-схемы. Можно выделить типичные алгоритмы. К ним относятся: линейные алгоритмы, разветвляющиеся алгоритмы, циклические алгоритмы.

Линейные алгоритмы

Линейный алгоритм является наиболее простым. В нём предполагается последовательное выполнение операций. В этом алгоритме не предусмотрены проверки условий или повторений.

Пример: Вычислить функцию $z = (x-y)/x + y^2$.

Составить блок-схему вычисления функции по линейному алгоритму. Значения переменных x , y могут быть любые, кроме нуля, вводить их с клавиатуры.

Решение: Линейный алгоритм вычисления функции задан в виде блок-схемы на рис.1. При выполнении линейного алгоритма значения переменных вводятся с клавиатуры, подставляются в заданную функцию, вычисляется результат, а затем выводится результат.

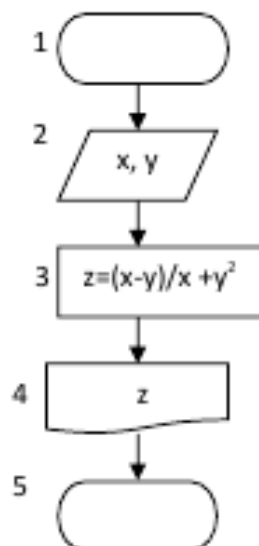


Рис.1. Линейный алгоритм

Назначение блоков в схеме на рис.1:

- Блок 1 в схеме служит в качестве логического начала.
- Блок 2 соответствует вводу данных.
- Блок 3 представляет арифметическое действие.
- Блок 4 выводит результат.
- Блок 5 в схеме служит в качестве логического завершения схемы.

Алгоритмы ветвлений

Разветвляющийся алгоритм предполагает проверку условий для выбора решения. Соответственно в алгоритме появятся две ветви для каждого условия.

В примере рассматривается разветвляющийся алгоритм, где в зависимости от условия выбирается один из возможных вариантов решений.

Алгоритм представляется в виде блок-схемы.

Пример: При выполнении условия $x > 0$ вычисляется функция: $z = x + y$, иначе, а именно, когда $x = 0$ или $x < 0$, вычисляется функция: $z = x^2 + y^2$.

Составить блок-схему вычисления функции по алгоритму ветвления. Значения переменных x , y могут быть любые, вводить их с клавиатуры.

Решение: На рис.2 представлен разветвляющийся алгоритм, где в зависимости от условия выполнится одна из веток. В блок-схеме появился новый блок 3, который проверяет условие задачи. Остальные блоки знакомы из линейного алгоритма.

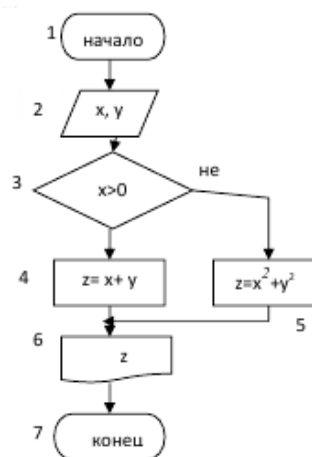


Рис. 2. Алгоритм ветвления

Пример: Найти максимальное значение из трёх различных целых чисел, введенных с клавиатуры. Составить блок-схему решения задачи.

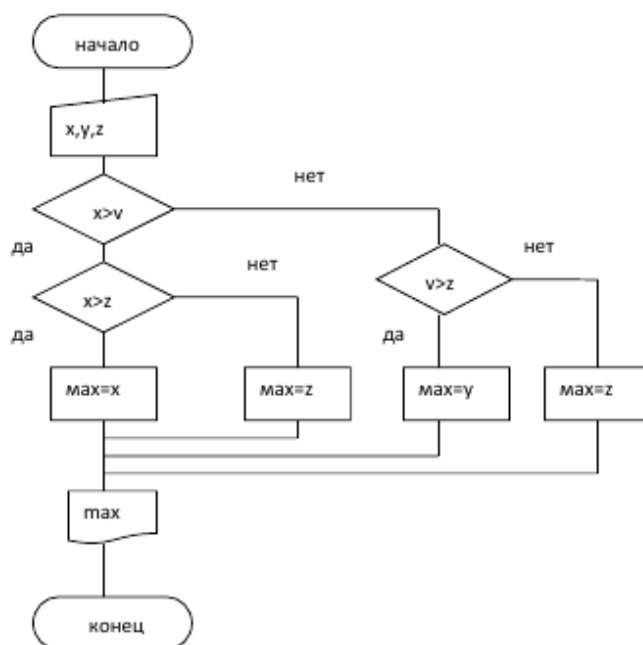


Рис. 3. Блок-схема поиска максимума

Решение: Данный алгоритм предполагает проверку условия. Для этого выбирается любая из трёх переменных и сравнивается с другими двумя. Если она больше, то поиск максимального числа окончен. Если условие не выполняется, то сравниваются две оставшиеся переменные. Одна из них будет максимальной. Блок-схема к этой задаче представлена на рис 3.

Циклические алгоритмы

Циклический алгоритм предусматривает повторение одной операции или нескольких операций в зависимости от условия задачи.

Из циклических алгоритмов выделяют два типа:

- 1) с заданным количеством циклов или со счётчиком циклов;
- 2) количество циклов неизвестно.

Пример: В цикле вычислить значение функции $z=x \cdot y$ при условии, что одна из переменных x меняется в каждом цикле на единицу, а другая переменная y не меняется и может быть любым целым числом. В результате выполнения цикла при начальном значении переменной $x=1$ можно получить таблицу умножения. Количество циклов может быть любым. Составить блок-схему решения задачи.

Решение: В примере количество циклов задаётся. Соответственно выбирается алгоритм циклов первого типа. Алгоритм этой задачи приводится на рис. 4.

Во втором блоке вводятся количество циклов n и любые целые числа x, y .

В блок-схеме появился новый блок 3, в котором переменная i считает количество циклов, после каждого цикла увеличиваясь на единицу, пока счётчик не будет равен $i=n$. При $i=n$ будет выполнен последний цикл.

В третьем блоке указывается диапазон изменения счётчика цикла (от $i=1$ до $i=n$).

В четвёртом блоке изменяются значения переменных: z, x .

В пятом блоке выводится результат. Четвёртый и пятый блоки повторяются в каждом цикле.

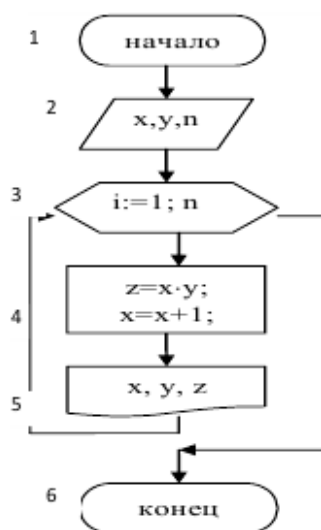


Рис.4 . Циклический алгоритм со счётчиком циклов

Этот тип циклических алгоритмов предпочтителен, если дано количество циклов.

Если количество циклов неизвестно, то блок-схемы циклических алгоритмов могут быть представлены в виде рисунков 5, 6.

Пример: Вычислить $y=y-x$ пока $y>x$, если $y=30, x=4$. Подсчитать количество выполненных циклов, конечное значение переменной y . В цикле вывести значение переменной y , количество выполненных циклов. Составить блок-схему решения задачи.

Решение: В примере количество циклов неизвестно. Соответственно выбирается алгоритм циклов второго типа. Алгоритм этой задачи приводится на рис. 5.

Условие проверяется на входе в цикл. В теле цикла выполняется два блока:

- 1) $y = y - x$; $i = i + 1$;
- 2) вывод значений переменных i , y .

Цикл выполняется до тех пор, пока выполняется условие $y > x$. При условии равенства этих переменных $y = x$ или $y < x$ цикл заканчивается.

Алгоритм, представленный на рис.5, называется циклический алгоритм с предусловием, так как условие проверяется в начале цикла или на входе в цикл.

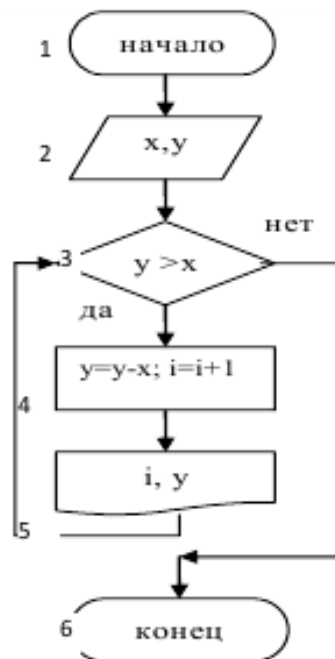


Рис.5. Блок-схема циклического алгоритма с предусловием

Во втором блоке вводятся $y=30$, $x=4$.

В третьем блоке проверяется условие $y > x$ на входе в цикл. Если условие выполняется, то переход к блоку 4, иначе на блок 6.

В четвёртом блоке вычисляется значение переменной y , подсчитывается количество выполненных циклов $i=i+1$.

В пятом блоке выводится результат:

- значение переменной y ,
- количество выполненных циклов i .

Пример: Составить блок-схему примера (рисунок 5), проверяя условие выхода из цикла. В этом примере условие задачи не меняется, и результат выведется тот же, но блок-схема будет другой.

Решение: В этом случае проверяется условие на выход из цикла: $y \leq x$. При этом условии цикл не выполняется. Условие в блок-схеме следует перенести в конец цикла, после вывода на печать. Цикл выполняется до тех пор, пока выполняется условие $y > x$.

Алгоритм, если условие перенести в конец цикла, называется *алгоритмом цикла с постусловием*.

Алгоритм этой задачи приводится на рис. 6.

Во втором блоке вводятся $y=30$, $x=4$.

В третьем блоке вычисляется значение переменной y , подсчитывается количество выполненных циклов $i=i+1$.

В четвёртом блоке выводится результат:

- значение переменной y ,
- количество выполненных циклов i .

В пятом блоке проверяется условие $y \leq x$ на выход из цикла. Если условие выполняется, то переход к блоку 6, иначе на блок 3 и цикл повторяется.

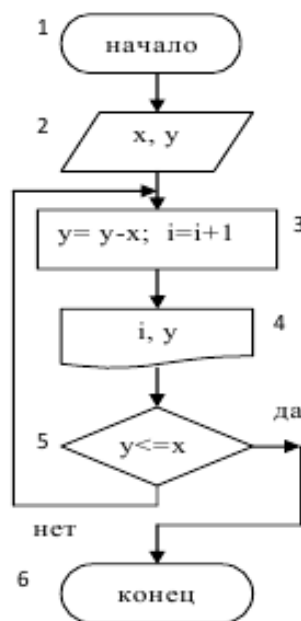
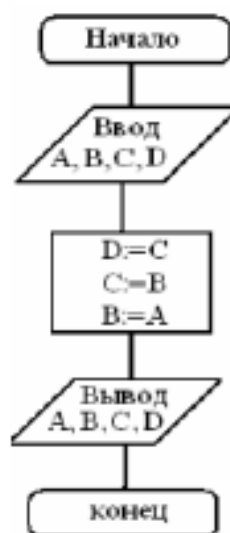


Рис.6 . Алгоритм цикла с постусловием

Индивидуальные задания к работе:

1. Найти результат работы алгоритма:



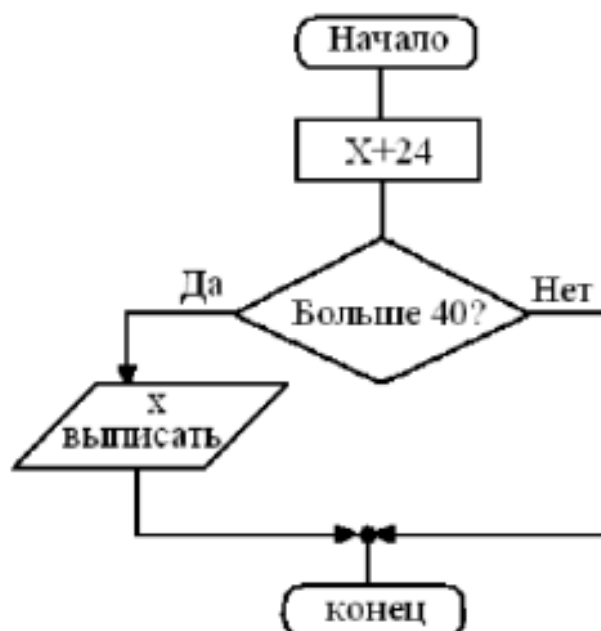
Входные данные по вариантам

№	A	B	C	D
1	0	-1	-2	-3
2	1	0	-1	-2
3	2	1	0	-1
4	3	2	1	0
5	4	3	2	1
6	5	4	3	2
7	6	5	4	3
8	7	6	5	4
9	-3	7	6	5
10	-4	-3	7	6
11	-5	-4	-3	7
12	-6	-5	-4	-3
13	-7	-6	-5	-4
14	9	-7	-6	-5
15	8	7	-7	-6

2. При заданном X условие выполняется? Написать результат вычисления и ответ попадаем в условие или нет.

Входные данные по вариантам

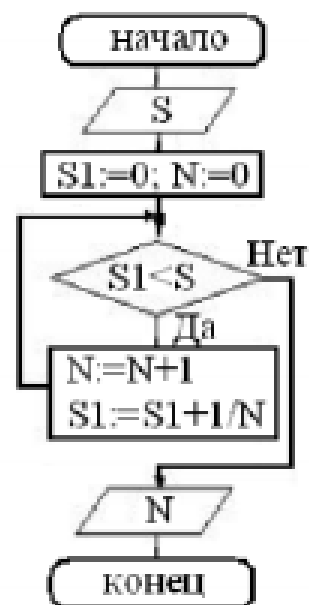
№	X1	X1
1	55	12
2	85	13
3	24	17
4	65	15
5	17	54
6	15	67
7	26	3
8	27	21
9	92	34
10	12	23
11	45	22
12	66	45
13	71	46
14	13	76
15	45	67



3. Написать результат выполнения алгоритма с указанными входными данными

Входные данные по вариантам

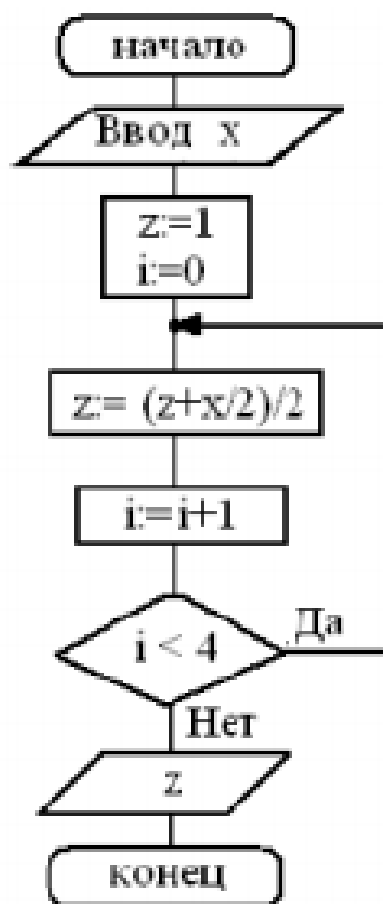
№	S
1	1,5
2	1,8
3	2,4
4	1,6
5	1,7
6	1,3
7	2,6
8	2,37
9	1,92
10	1,12
11	1,45
12	2,66
13	2,71
14	2,13
15	1,45



4. Написать результат выполнения алгоритма с указанными входными данными

Входные данные по вариантам

№	X
1	-1
2	0
3	1
4	2
5	3
6	4
7	5
8	6
9	7
10	-3
11	-4
12	-5
13	-6
14	-7
15	7



5. Построить блок схему к задаче(по вариантам). Указать тип алгоритма, что дано и что нужно найти.

№	Задача
1	Дано двузначное число. Найти: число десятков в нем;
2	Дано двузначное число. Найти: число единиц в нем;
3	Дано двузначное число. Найти: сумму его цифр;
4	Дано двузначное число. Найти: произведение его цифр.
5	Дано двузначное число. Получить число, образованное при перестановке цифр заданного числа.
6	Дано трехзначное число. Найти: число единиц в нем;
7	Дано трехзначное число. Найти: число десятков в нем;
8	Дано трехзначное число. Найти: сумму его цифр;
9	Дано трехзначное число. Найти: произведение его цифр.
10	Дано трехзначное число. Найти число, полученное при прочтении его цифр справа налево.
11	Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее в конце. Найти полученное число.
12	Дано трехзначное число. В нем зачеркнули последнюю справа цифру и приписали ее в начале. Найти полученное число.
13	Дано трехзначное число. Найти число, полученное при перестановке первой и второй цифр заданного числа
14	Дано трехзначное число. Найти число, полученное при перестановке второй и третьей цифр заданного числа.
15	Дано трехзначное число, в котором все цифры различны. Получить шесть чисел, образованных при перестановке цифр заданного числа.

6. Построить блок схему к задаче (по вариантам). Указать тип алгоритма, что дано и что нужно найти.

№	Задача
1	Определить максимальное и минимальное значения из двух различных вещественных чисел.
2	Известны два расстояния: одно в километрах, другое — в футах (1 фут 0,305 м). Какое из расстояний меньше?
3	Известны две скорости: одна в километрах в час, другая — в метрах в секунду. Какая из скоростей больше?
4	Даны радиус круга и сторона квадрата. У какой фигуры площадь больше?
5	Даны объемы и массы двух тел из разных материалов. Материал какого из тел имеет большую плотность?
6	Известны сопротивления двух несоединенных друг с другом участков электрической цепи и напряжение на каждом из них. По какому участку протекает меньший ток?
7	Даны вещественные числа a , b , c ($a \neq 0$). Выяснить, имеет ли квадратное уравнение с данными параметрами решение
8	Известны площади круга и квадрата. Определить: уместится ли круг в квадрате?
9	Известны площади круга и квадрата. Определить: уместится ли квадрат в круге?
10	Известны площади круга и равностороннего треугольника. Определить: уместится ли круг в треугольнике?
11	Известны площади круга и равностороннего треугольника. Определить: уместится ли треугольник в круге?
12	Дано двузначное число. Определить: какая из его цифр больше: первая или вторая;
13	Дано двузначное число. Определить: одинаковы ли его цифры
14	Дано двузначное число. Определить: кратна ли трем сумма его цифр;
15	Дано двузначное число. Определить: кратна ли сумма его цифр числу a .

Практическое занятие 8. Создание архива данных и работа с ним.

Цели:

Учебная: сформировать понятия сжатия и архивации файлов; выполнять алгоритм архивации и разархивации файлов.

Развивающая: учить анализировать, выделять главное, обобщать и систематизировать.

Воспитательная: воспитывать информационную культуру у студентов, самостоятельность и ответственность за выполненную работу.

1. Сжатие данных

Система двоичного кодирования, которая используется в компьютерах, очень удобна для хранения, передачи и обработки данных с точки зрения надежности этих процессов. Однако двоичное кодирование увеличивает размеры файлов по сравнению с другими системами кодирования. Поэтому возникает необходимость в уменьшении размеров файлов.

Цели сжатия:

- Уменьшить место, которое занимают файлы на диске;
- Уменьшить объем данных, которые передаются через Интернет.

Сжатие данных – это процесс перекодирования данных, который осуществляется с целью уменьшения размеров файлов.

Типы сжатия:

1. Без потерь: сжатый файл можно восстановить в исходном виде, если знать алгоритм сжатия:

- Тексты;
- Программы;
- Данные.

2. С потерями: при сжатии часть информации безвозвратно теряется:

- Графическое изображение;
- Звук;
- Видео.

2. Архивация

Создание копий данных с помощью специальных программ, которые могут сжимать данные, называется архивацией.

Архивация – это процесс сжатия информации, которая сохраняется в файлах

Цель архивации:

- уменьшить место, которое занимают файлы на диске;
- создать резервную копию данных;
- уменьшить объем данных, которые передаются через Интернет;
- объединить группу файлов в один архив;
- зашифровать данные с паролем.

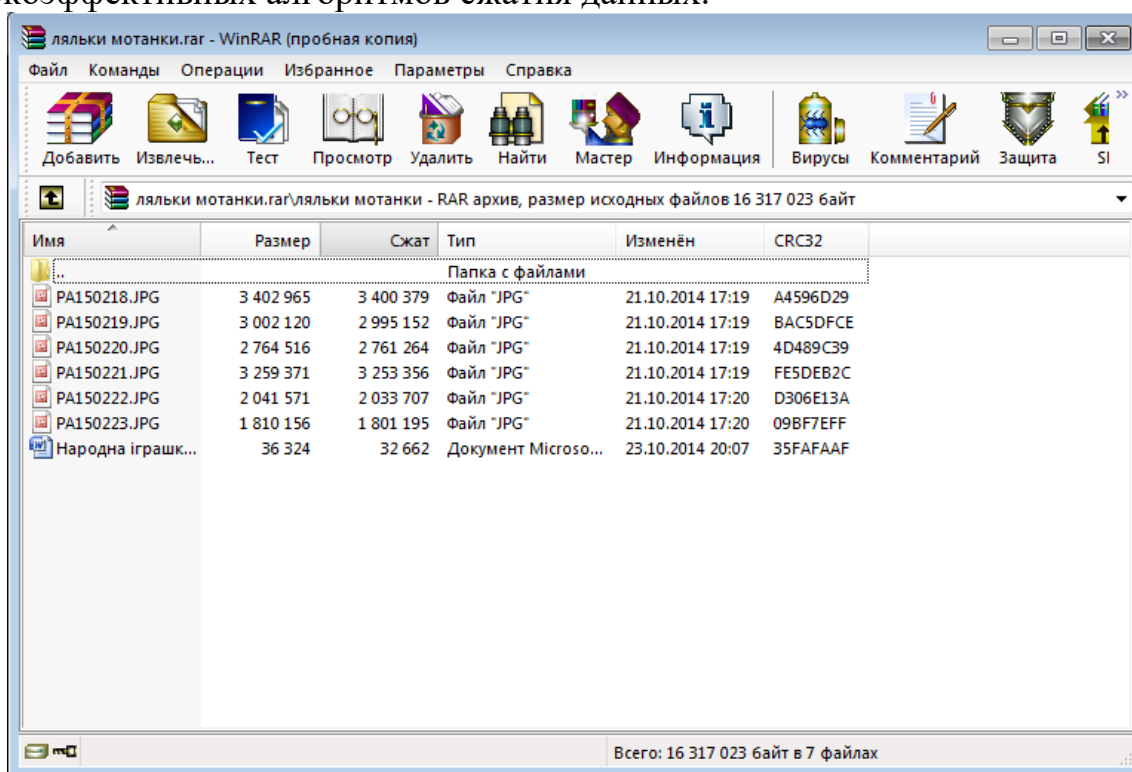
Программы, которые используются для выполнения архивации, называются архиваторами.

Архиватор – это программа, которая сжимает и объединяет файлы.

Результатом работы этих программ является архивный файл, или просто архив, который содержит в сжатом и не в сжатом состоянии файлы и папки. В зависимости от алгоритмов, по которым осуществляется архивация данных, различают такие форматы архивных файлов: ZIP, RAR, ARJ, CAB, LZH, ACE, ISO. Чаще всего, особенно в сети Интернет, используются архивные файлы формата ZIP. Формат RAR обеспечивает самое эффективное сжатие. Примерами архиваторов являются программы WinZip, WinRAR, 7-Zip, Winace, PowerArchiver, ArjFolder, BitZipper, Gnochivebzip2.

3. Программа-архиватор WinRAR

Программа-архиватор WinRAR была создана российскими братьями-программистами Евгением и Александром Рошал с целью использования высокоэффективных алгоритмов сжатия данных.



Основные функции этой программы такие:

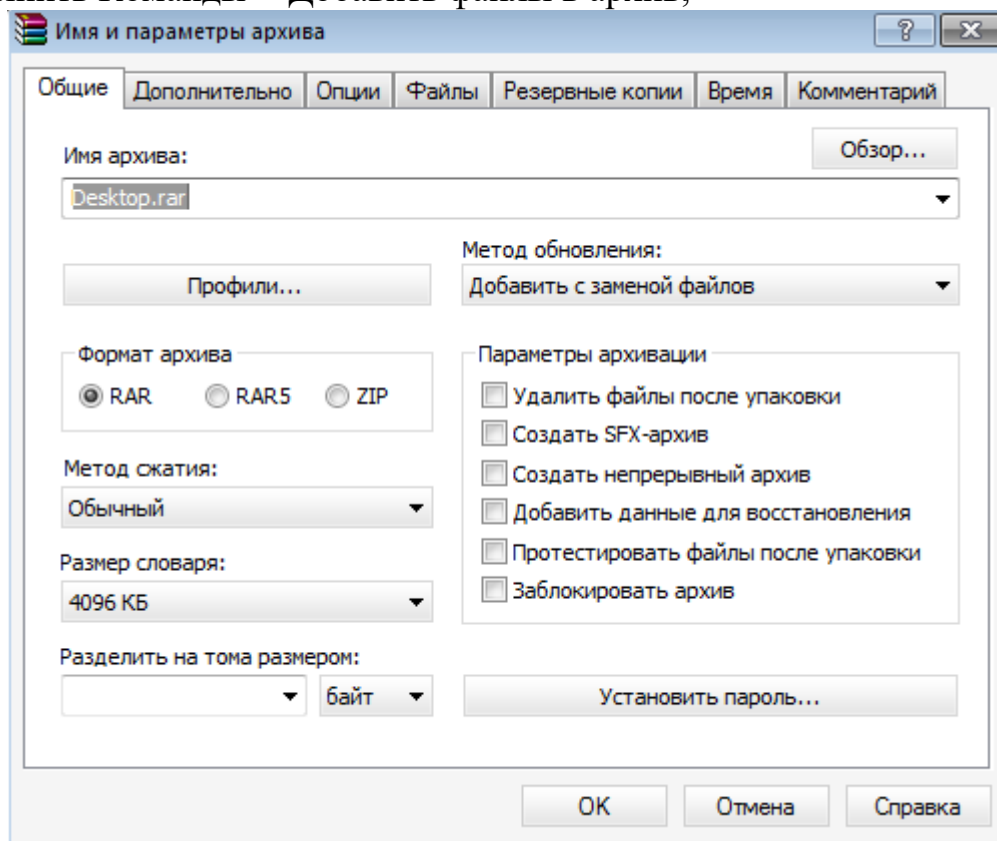
- создание архивов файлов и папок с возможным сжатием данных;
- добавление файлов и папок к уже существующим архивам;
- просмотр содержимого архивов;
- создание многотомных архивов (архив разбивается на несколько отдельных файлов –томов), размер томов устанавливает пользователь;
- создание обычных и многотомных архивов, которые содержат программы самостоятельного извлечения файлов и папок, без участия программы-архиватора – так называемых SFX – архивов;
- проверка целостности в архивах;

- шифровка данных и имен файлов в архивах.

Создание архива с использованием программы-архиватора WinRAR можно выполнить несколькими способами:

1. Выполнить следующий алгоритм:

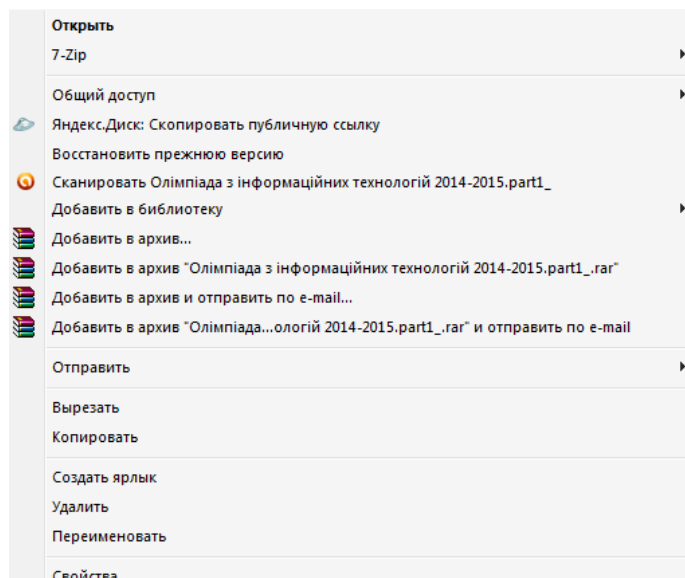
- Запустить программу WinRAR на выполнение (Пуск => Все программы => WinRAR => WinRAR);
- Выполнить Команды=>Добавить файлы в архив;



- Выбрать вкладку файлы;
- Выбрать нужные объекты для архивации;
- Выбрать вкладку общие;
- Ввести в поле имя архива имя архивного файла;
- Указать папку, в которой будет сохранен архив (кнопка Обзор);
- Выбрать метод сжатия;
- Указать значение параметров архивации (при необходимости);
- Выбрать формат архивного файла (RAR или ZIP);
- Создать многотомный архивный файл (при необходимости);
- Выбрать кнопку ОК.

Многотомные архивы создаются для разделения архивного файла на несколько частей, если полностью архив не помещается на одном носителе, например на диске, на флеш-карте, или если нужно передать файл по сети с низкой скоростью передачи данных.

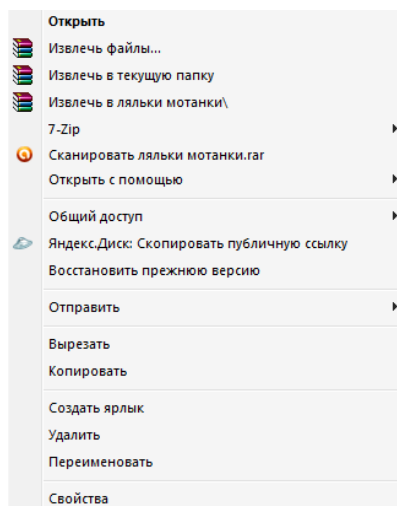
2. При помощи контекстного меню: выбор команды **Добавить в архив** или **Добавить в архив и отправить по e-mail** открывает окно установки режимов архивации.



Для извлечения объектов из архива необходимо:

1. Запустить архив, объекты которого нужно извлечь;
2. Выполнить Команды => Извлечь в указанную папку (или выбрать извлечь на Панели инструментов);
3. Указать в диалоговом окне Путь и параметры извлечения папку, в которую будет осуществляется извлечение;
4. Установить значения параметров извлечения;
5. Выбрать кнопку ОК.

Можно использовать контекстное меню файла архива: **Извлечь файлы**, **Извлечь в текущую папку**...



Задание

1. изучить метод сжатия информации архивацией данных.
2. выяснить, какие файлы целесообразно подвергать архивации.

Ход работы:

1. На компьютере в папке студентам общая – Практическая работа 13, создать папку с именем «Архивация и своя фамилия».

2. В нее скопировать следующие типы файлов: графический (jpg), текстовый, звуковой, презентация, табличный.
3. Заархивировать каждый файл и всю папку, используя программу архиватор RAR.

Технология создания архива при помощи программы WinRAR.

1. Выделить необходимый файл
2. Открыть контекстное меню, выбрать команду *WinRAR – добавить в архив*.
3. Выбрать формат архива RAR, метод сжатия *обычный*.
4. Посмотреть, как изменится размер файла по отношению к размеру архива.
5. Результаты занести в таблицу и посчитать коэффициент сжатия по формуле

$$K_c = (V_c / V_o) * 100\%$$

Степень сжатия файлов характеризуется коэффициентом K_c , определяемым как отношение объема сжатого файла V_c к объему исходного файла V_o , выраженное в процентах

6. Степень сжатия зависит от используемой программы, метода сжатия и типа исходного файла.

Имя файла	Тип файла	Размер файла	Размер архива	Коэффициент сжатия

7. Ответить в тетради на контрольные вопросы
8. Сделать вывод по работе. В выводе указать, какие файлы следует сжимать и почему. (Докажите или опровергните утверждение, приведенное ниже)

Для справки! Наиболее хорошо сжимаются файлы графических образов, текстовые файлы и файлы данных, для которых коэффициент сжатия может достигать 5 - 40%, меньше сжимаются файлы исполняемых программ и загрузочных модулей $K_c = 60 - 90\%$. Почти не сжимаются архивные файлы.

Контрольные вопросы

1. Какие способы сжатия информации вы знаете?
2. Что такое архивация файлов?
3. С какою целью создаются архивы?
4. Как называются программы, которые выполняют архивацию данных?

Практическая работа 9. Поиск информации, хранящейся на компьютере. Программные поисковые сервисы.

Цель работы. Изучение способов поиска информации в тексте, в файловых структурах, в базах данных, в сети Интернет.

План

- 1) Изучить основные понятия поиска информации
- 2) Выполнить поиск информационного объекта в тексте
- 3) Выполнить поиск информационного объекта в файловых структурах
- 4) Выполнить поиск информационного объекта в сети Интернет
- 5) Ответить на контрольные вопросы

Краткие сведения

Поиск информации - задача, которую человечество решает уже многие столетия. По мере роста объема информационных ресурсов, потенциально доступных одному человеку, были выработаны все более изощренные и совершенные поисковые средства и приемы, позволяющие найти необходимый документ. Обширные возможности для работы с большими массивами информации дают поисковые сервисы Internet.

При наличии первичных сведений по теме поиска, документы можно разыскивать в поисковых системах. При этом следует различать приемы *простого, расширенного, контекстного и специального поиска*.

Под **простым поиском** понимается поиск Web-ресурсов по одному или нескольким ключевым словам. Недостаток простого поиска заключается в том, что обычно он выдает слишком много документов, среди которых трудно выбрать наиболее подходящие.

При использовании **расширенного поиска** ключевые слова связывают между собой операторами логических отношений. Расширенный поиск применяют в тех случаях, когда приемы простого поиска дают слишком много результатов. С помощью логических отношений поисковое задание формируют так, чтобы более точно детализировать задание и ограничить область отбора, например по дате публикации или типу данных.

Контекстный поиск – это поиск по точной фразе. Он удобен для реферативного поиска информации, но доступен далеко не во всех поисковых системах. Прежде всего, чтобы обеспечивать такую возможность, система должна работать не только индексированными файлами, но и с полноценными образами Web-страниц. Эта операция достаточно медленная, и ее выполняют не все поисковые системы.

Специальный поиск применяют при розыске Web-страниц, содержащих ссылки на заданные адреса URL, содержащих заданные данные в служебных полях, например в поле заголовка и т.п.

Расширенный поиск. Кроме средства простого поиска обычно поисковые службы предоставляют средства расширенного поиска. Эти средства позволяют более точно формулировать поисковое задание, но требуют определенного опыта и работают заметно медленнее. В большинстве поисковых систем команды расширенного поиска формируются с помощью логических команд. Удобство использования логических команд в частности связано с тем, что команды простого поиска у многих поисковых систем реализованы по-разному. Каждая система стремится сделать средства простого поиска наиболее удобными, а средства расширенного поиска – наиболее стандартными. Тем не менее, для обозначения логических операторов в различных поисковых системах используются разные обозначения. Поэтому желательно перед осуществлением расширенного поиска желательно изучить синтаксис поисковых запросов выбранной поисковой системы.

Рассмотрим подробнее операторы логических отношений (логические команды).

Логическая оператор **OR (ИЛИ)** служит для формирования поискового запроса, если искомый текст должен содержать хотя бы один из терминов, соединенных данным оператором. Этот оператор в различных поисковых системах может обозначаться одним из следующих способов: **| ; OR; ИЛИ.**

Например, результат запроса «Чёрное OR море» - будет представлен списком ссылок на документы, в которых есть слово «Чёрное», или слово «море», или оба этих слова вместе.

В некоторых поисковых системах, как отмечалось выше, по умолчанию ключевые слова в запросе связаны именно этим логическим отношением.

С помощью логического оператора **AND (И)** осуществляется поиск документов, содержащих все термины, соединенные данным оператором. Этот оператор может обозначаться одним из следующих способов: **+; AND; &; И.**

Например, по запросу – «Чёрное AND море» - будут найдены документы, в которых содержатся слова «черное» и «море».

Логической оператор **NOT (НЕ)** позволяет производить поиск документов, в тексте которых отсутствуют термины, следующие за данным оператором. Этот оператор может обозначаться одним из следующих способов: **not; !; ~; НЕ.**

Например, по запросу – «Чёрное NOT море», результат - документы, в которых есть слово «Чёрное» и нет слова «море».

С помощью логических операций можно создавать достаточно сложные запросы. Запрос из нескольких слов, перемежающихся операторами, будет истолкован в соответствии с их приоритетом. Операторы AND и NOT традиционно имеют более высокий приоритет, поэтому запрос из нескольких слов при обработке сначала группируется по операторам AND и NOT, и лишь потом по операторам OR. Например, по запросу «Чёрное AND море OR Крым» будут найдены документы, либо содержащие обязательно слова: «Чёрное» и «море», либо слово «Крым», либо все три слова.

Изменить порядок группировки можно использованием [скобок](#). Оператор, стоящий в скобках, будет выполняться в первую очередь. Использование скобок позволяет строить вложенные запросы и передавать их операторам в качестве аргументов. Так по запросу «Чёрное AND (море OR Крым)» будут найдены документы, в которых обязательно содержится слово «Чёрное» и одно из двух слов «море» или «Крым».

С помощью вложенных запросов можно значительно ограничивать область отбора, освобождая результирующий список от ненужных ссылок. Так, например, если нас интересует информация об отдыхе на юге на море, но исключительно на российском побережье, то можно попробовать использовать примерно такой запрос – «отдых AND ((Азовское OR Чёрное) AND море) NOT (Крым OR Турция OR Болгария)».

Использование круглых скобок для управления порядком исполнения задания на поиск разрешается большинством крупнейших поисковых систем.

Расширенный поиск yandex.ru

Оператор	Описание
"	Обнаруживает точные слова в кавычках или фразы
	Найти любое из слов. Достаточно поставить между словами символ , и вы получите страницы, где содержится хоть одно из слов запроса.
~	Исключает страницы, содержащие слово или фразу.
()	Вы можете строить сколь угодно сложные конструкции, подставляя в каждом из операторов вместо отдельного слова целые выражения. Чтобы Яндекс при этом правильно понимал вас, заключайте выражения в круглые скобки.
&	Ограничить поиск страницами, где слова запроса находятся в пределах предложения
&&	Если вам нужны документы, где присутствуют заданные слова — неважно, на каком расстоянии и в каком порядке — соедините их оператором
!	Слова с большой и маленькой буквы считаются разными формами одного слова, поэтому все равно, какой регистр использовать в запросе. Исключением является оператор точной формы. Это полезно, если искомое имя собственное совпадает с распространенным словосочетанием, например, группа !Черный кофе . Все слова, которые вы приводите в запросе, по умолчанию ищутся с учетом морфологии. Чтобы отключить ее, используйте оператор ! перед словом (без пробела).
/	Вы можете указать максимально допустимое расстояние между двумя любыми словами запроса, поставив после первого слова символ /, сразу за которым идет число, означающее расстояние.
*	Замена части слова. Журналист*
?	Замена любого символа. Журналистик?

Расширенный поиск [Rambler.ru](http:// Rambler.ru)

Оператор	Описание
+	Используется для включения общих слов
"	Слова запроса, заключенного в двойные кавычки, ищутся в документах именно в том порядке и в тех формах, в которых они встретились в запросе. Таким образом, двойные кавычки можно использовать и просто для поиска слова в заданной форме (по умолчанию слова находятся во всех формах).
NOT	Оператор NOT позволяет сформировать запрос, которому отвечают документы, удовлетворяющие левой части запроса и не удовлетворяющие правой. Так, результатом поиска по запросу собака NOT кошка будут все документы, в которых есть слово "собака" и нет слова "кошка". Это особенно полезно в случаях, если искомая словоформа является одновременно формой другого слова: Женя - имя собственное, а также деепричастие. Если мы ищем именно человека по имени Женя, в запросе можно написать Женя NOT женить.
&&	Два запроса, соединенные оператором &&, образуют сложный запрос, которому удовлетворяют только те документы, которые одновременно удовлетворяют обоим этим запросам. Иными словами, по запросу собака && кошка найдутся только те документы, которые содержат и слово "собака", и слово "кошка".
	Сложному запросу, состоящему из двух запросов, соединенных оператором , удовлетворяют все документы, удовлетворяющие хотя бы одному из этих двух запросов. По запросу <i>собака кошка</i> найдутся документы, в которых есть хотя бы одно из двух слов - слово "собака" или слово "кошка" (или оба эти слова вместе).
()	Использование скобок позволяет строить вложенные запросы и передавать их операторам в качестве аргументов, а также перекрывать приоритеты операторов, принятые по умолчанию.

Организация специального поиска. С помощью средств специального поиска можно осуществлять поиск документов: содержащих искомые данные в различных полях (заголовке, ключевых словах, описании); расположенных на каком-либо сайте; содержащих те или иные ссылки и т.п. Рассмотрим подробнее некоторые возможности специального поиска.

Многие поисковые системы позволяют разыскивать Web-документы по тексту, содержащемуся в заголовках. Поиск по заголовкам существенно уменьшает количество найденных ссылок, но очень точно выводит на нужные материалы. Ведь каждая Web-страница может иметь заголовок, если её автор не поленился его создать. И заголовок Web-страницы обычно точно характеризует тему материала, который содержится на ней.

Например, если вам нужна информация о дистанционном обучении, то целесообразно искать страницы, в которых это сочетание присутствует в

заголовке. Таким образом, в отличие от простого запроса мы отсекаем те документы, где эти слова не являются значимыми, т.е. не определяют тему статьи.

Оператором или командой такого поиска является `title`. Этот оператор может обозначаться одним из следующих способов: `title:`, `t:`, `title =`, `$title` и т.п. После оператора следуют ключевые слова. В некоторых поисковых системах ключевые слова следует заключать в скобки, в других они пишутся без скобок. Например, в Яндекс команда поиска в заголовке записывается так: `$title (очное обучение)`.

Поиск по сайту. С помощью поисковых систем можно осуществлять поиск информации не во всем Web пространстве, а на каком то конкретном сайте (если конечно последний проиндексирован поисковой системой). Соответствующий оператор может обозначаться следующим одним из следующих способов: `url=`, `url:`, `u:`, `#url=`. Далее следует адрес Web-узла, некоторые системы требуют заключения адреса в кавычки.

Если в запросе просто записать данный оператор с адресом какого-либо Web-узла, то будет получен список документов, проиндексированных поисковой системой на данном сайте. Но этот оператор можно комбинировать с другими, тем самым, осуществляя поиск информации по всем правилам построения запросов на данном сайте. Например, если мы хотим найти информацию об очном обучении на сайте БелГУ, то запрос в поисковой системе Апорт будет выглядеть следующим способом: `URL=http://www.bsu.edu.ru + "очное обучение"`.

Кроме того, поисковые системы могут предлагать другие возможности специального поиска: поиск по тексту ссылок, поиск в описании документа, поиск в списке ключевых слов Web-страниц, поиск по подписям к рисункам и т.п. Следует знать, что синтаксис поисковых запросов, да и состав доступных операторов, отличается в различных поисковых указателях. Поэтому перед осуществлением поиска в той или иной поисковой системе, следует изучить страницу справки по поиску в данной системе.

Специальный поиск **aport.ru**

Оператор	Описание
<code>title=</code>	Указанное после равенства слово или конструкция в круглых скобках должны искаться в заголовках документов (перед круглыми скобками знак равенства можно опускать). По запросу <code>title=(папа или мама)</code> будут найдены документы, содержащие в заголовке слово папа или слово мама, или оба слова одновременно.
<code>anchor=</code>	Указанное после равенства слово или конструкция в круглых скобках должны искаться в тексте ссылок. По запросу <code>anchor=(ненавижу Интернет)</code> будут найдены документы, в тексте ссылок на которые, встречаются оба слова: ненавидеть и Интернет.
<code>text=</code>	Указанное после равенства слово или конструкция в круглых скобках должны искаться только в обычном тексте. По умолчанию слова

	запроса ищутся как в тексте, так и во всех указанных выше полях. Чтобы искать только по тексту, выдаваемому в основное окно браузера, следует использовать данный оператор. По запросу text=(оглавление или содержание) будут найдены документы, в которых любое из указанных слов встречается в пределах основного текста документа.
--	---

Задание к работе

Задание 1. Вставить в текстовом процессоре Microsoft Word любой текст и выполнить поиск информационного объекта в тексте документа в текстовом процессоре Microsoft Word.

Порядок работы

1. Запустить на выполнение текстовый процессор Microsoft Word и открыть документ, в котором будет выполняться поиск фразы.
2. Выполнить команду Правка-Найти и указать искомый текст, выделить найденный текст красным цветом.

Задание 2. Выполнить поиск информационного объекта с расширением PDF в файловых структурах диска C и D:

Порядок работы

1. В операционной системе Windows выполнить команду Пуск – Поиск
2. Указать искомый объект *.PDF
3. Указать место поиска – локальный диск D:

Задание 3. Выполнить поиск информационного объекта в сети Интернет

Порядок работы

1. Произведите простой поиск информации по ключевым словам «информационные технологии» в поисковых указателях Яндекс (<http://www.yandex.ru>), rambler (<http://www.rambler.ru>), АПОРТ (<http://www.aport.ru>). Результаты поиска по каждой поисковой системе запишите в отчёт.
2. Осуществите контекстный поиск (поиск по точной фразе) информации в поисковой машине rambler по ключевым словам «информационные технологии». В файл отчета запишите запрос, удовлетворяющий данному условию и результат выполнения запроса в rambler.
3. Выполните поиск информации в поисковой машине Aport по ключевым словам «информационные технологии», но с помощью операторов расстояния задайте поиск так, чтобы слова «информационные» «технологии» стояли рядом.
4. Используя любую поисковую систему, найдите материал, который помог бы Вам ответить на следующие вопросы:
 1. Сколько медалей было завоевано российскими спортсменами на летней олимпиаде 2012 года в Лондоне?
 2. Годы царствования Петра I.
 5. Выполните поиск документов, в которых ключевые слова «информационные технологии» находятся в заголовке. В файл отчета

запишите запрос, удовлетворяющий данному условию и результат выполнения запроса в Яндекс.

6. Перейдите на сайт по адресу <http://www.edu.ru/>. В разделе учреждения выполните поиск информации о вашем техникуме. Запишите последовательность поиска в файл отчета.

Контрольные вопросы

1. Как выполнить поиск информационного объекта в текстовом процессоре MS Word?
2. Как выполнить поиск информационного объекта в файловых структурах Windows?
3. Перечислите известные вам поисковые машины.
4. Для чего в некоторых поисковых системах используется расширенный поиск?
5. Каким логическим оператором связаны ключевые слова в простом запросе в рассмотренных поисковых машинах?
6. Как в поисковой машине Яндекс осуществить поиск точной формы слов?
7. Какие области поиска можно определить в Яндекс?
8. Как в Яндекс указать расстояние между ключевыми словами?

Практическая работа 10. Поиск информации на государственных образовательных порталах.

Цели:

научиться пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью; овладеть навыками установки программного обеспечения.

изучить лицензионные и свободно распространяемые программные продукты; научиться осуществлять обновление программного обеспечения с использованием сети Интернет.

Оборудование: ПК, интернет-браузер.

Ход работы

Используя правила формирования запросов, с помощью поисковых систем найти ответы на вопросы

Ответить на контрольные вопросы

ЗАДАНИЕ № 1. Работа с поисковыми системами

Используя правила формирования запросов в поисковой системе, найдите ответы на вопросы

- 1) создайте на диске X:\ документ Microsoft Office Word, назовите его ЛР32
 - 2) запустите поисковую систему, скопируйте вопрос, найдите ответ на вопрос
 - 3) вставьте вопрос в документ ЛР32, скопируйте адрес страницы и вставьте в документ
 - 4) скопируйте ответ и вставьте в документ
- Что такое информационное общество?
- Что такое информационные ресурсы?
- Чем характеризуются национальные ресурсы общества?
- Что такое инсталляция (деинсталляция) программного обеспечения?
- Порядок инсталляция (деинсталляция) программного обеспечения?
- Какие программы называют лицензионными?
- Какие программы называют условно бесплатными?
- Какие программы называют свободно распространяемыми?
- В чем состоит различие между лицензионными, условно бесплатными и бесплатными программами?
- Как можно зафиксировать свое авторское право на программный продукт?
- Какие используются способы идентификации личности при предоставлении доступа к информации?
- Почему компьютерное пиратство наносит ущерб обществу?
- Какие существуют программные и аппаратные способы защиты информации?
- Чем отличается простое копирование файлов от инсталляции программ?
- Назовите стадии инсталляции программы.
- Что такое инсталлятор?
- Как запустить установленную программу?
- Как удалить ненужную программу с компьютера?

ЗАДАНИЕ № 2. Работа с Универсальным справочником-энциклопедией
<http://www.sci.aha.ru/ALL/index.htm>

Найдите в Универсальном справочнике-энциклопедии ответы на следующие вопросы и запишите ответы в документ ЛР32:

ВНИМАНИЕ! Используйте для поиска на странице сочетание клавиш Ctrl + F

Укажите время утверждения григорианского календаря.

Каков диаметр атома?

Укажите смертельный уровень звука.

Какова температура кипения железа?

Какова температура плавления ртути?

Укажите скорость обращения Луны вокруг Земли?

Какова масса Земли?

Какая гора в России является самой высокой?

Дайте характеристику народа кампа.

Укажите годы правления Екатерины I.

Укажите годы правления Ивана IV.

Укажите годы правления Хрущева Н.С.

В каком году был изобретен первый деревянный велосипед?

ЗАДАНИЕ № 3. Работа с законом РФ «Об информации, информатизации и защите информации»

Найти в Интернет закон РФ «Об информации, информатизации и защите информации», скопировать и вставить определения понятий в документ ЛР32

информация;

информационные технологии;

информационно-телекоммуникационная сеть;

доступ к информации;

конфиденциальность информации;

электронное сообщение;

документированная информация.

ЗАДАНИЕ № 4. Работа с поисковыми системами

Используя правила формирования запросов, найдите ответы на вопросы, скопируйте вопрос, адрес страницы и ответ в документ под своей фамилией

1. Какой вид поиска является самым быстрым и надежным?

2. Где пользователь может найти адреса Web-страниц?

3. Каково основное назначение поисковой системы?

4. Из каких частей состоит поисковая система?

5. Какие поисковые системы вы знаете?

6. Какова технология поиска по рубрикатору поисковой системы?

7. Какова технология поиска по ключевым словам?

8. Когда в критерии поиска надо задавать + или -?

Практическая работа 11. «Создание ящика эл. почты. Формирование адресной книги»

Цель работы: создать и оформить ящик электронной почты.

Ход работы:

- 1) Изучить теоретический материал
- 2) Выполнить практические задания

Обработка почтовой корреспонденции с помощью программы Microsoft Outlook Express

Рекомендации для преподавателя:

Перед началом практической работы необходимо отправить

- на все рабочие столы электронное письмо с темой «Ваше мнение?», содержащие вопросы
- на первый рабочий стол письмо с темой «Ознакомиться всем!» об изменении даты проведения педагогического совета;
- на все рабочие столы письмо с темой «Сохраните вложение!», к которому прикреплен файл с текстовым документом (например, план работы образовательного учреждения на текущий месяц).

Примечание. Упражнение 7 практической работы выполняется по мере поступления сообщения с темой «Ознакомиться всем!».

Упражнение 1. Знакомство с основными возможностями и элементами интерфейса клиентской почтовой программы Microsoft Outlook Express

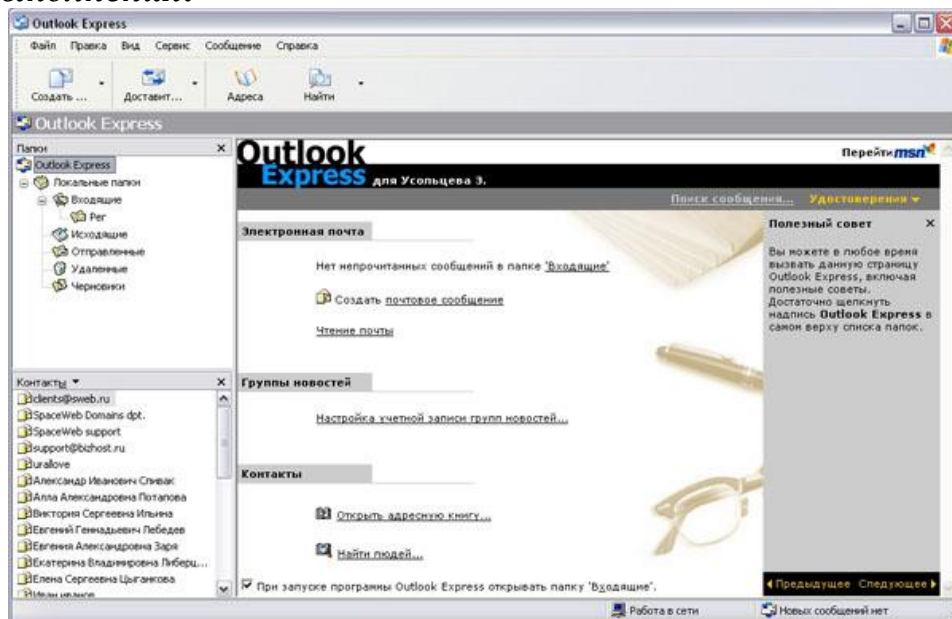
Цель упражнения:

Формирование первоначальных навыков оптимальной работы с клиентской программой Microsoft Outlook Express.

Задание:

Загрузите почтового клиента Microsoft Outlook Express, изучите основные элементы интерфейса.

Порядок выполнения:



1. Запустите клиентскую программу Microsoft Outlook Express с помощью значка



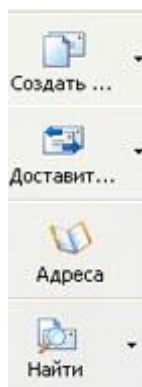
на **Рабочем столе** или соответствующей кнопки на **Панели задач**. После запуска программы появится окно:

Основными элементами интерфейса программы Microsoft Outlook Express окна являются:

Строка заголовка. Содержит стандартные элементы окна Windows - приложения (кнопки *Свернуть*, *Восстановить* и *Заккрыть*) и название приложения Outlook Express.

Строка меню. Содержит пункты меню, предоставляющие доступ ко всем функциям, необходимым при работе с почтовым клиентом (создание, отправка и получение сообщений, настройка интерфейса и проч.).

Панель инструментов. Предназначена для быстрого доступа к некоторым наиболее часто используемым командам:



Создать сообщение — открывает окно для формирования нового письма

Доставить почту — получение и (или) отправка почтовой корреспонденции.

Адреса — открывает доступ к адресной книге.

Поиск — поиск почтового сообщения или адресата по атрибутам.

Панель Локальные папки. Позволяет вывести на экран списки почтовых сообщений (и их содержимое), хранящихся в одной из стандартных папок почтового клиента:

Входящие. В эту папку поступает вся новая почта. Впоследствии можно создать дополнительные папки (в соответствии с выбранной пользователем логической структурой) и настроить программу так, чтобы при поступлении новых писем вся почта автоматически сортировалась по папкам.

Исходящие. Эта папка предназначена для временного хранения отправляемых писем.

Отправленные. Здесь по умолчанию хранятся копии отправленных сообщений.

Удаленные. Для временного хранения удаленных сообщений (на случай, если сообщение потребуется восстановить). Очистка папки приведет к удалению сообщения без возможности восстановления.

Черновики. Для хранения «недописанных» писем.

Панель Контакты. В этом окне фиксируются имена клиентов, адреса которых внесены в адресную книгу.

Область просмотра. Позволяет обозревать список сообщений в текущей папке и содержимое отмеченного письма.

2. Выделите папку **Входящие** на панели **Локальные папки**. Область просмотра при этом делится на две части. Вверху отображается список сообщений

электронной почты из текущей папки, а в нижней части окна показывается содержимое выделенного письма. Последовательно выделяя заголовки писем, просмотрите их содержимое.

3. Аналогично просмотрите содержимое остальных локальных папок.

Упражнение 2. Знакомство с основными приемами доставки и сохранения почтовых сообщений с помощью программы Microsoft Outlook Express.

Цель упражнения:

Формирование навыков получения почтовой корреспонденции и ее размещения в локальных папках при работе с клиентской программой Microsoft Outlook Express.

Задание:

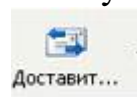
Доставьте почтовую корреспонденцию и сохраните некоторые сообщения в специально созданных папках.

Порядок выполнения:

1. В папке **Входящие** создайте папку **Моя корреспонденция**:

- выберите пункт меню **Файл - Создать - Папка** ;
- в поле ввода введите имя новой папки **Моя корреспонденция**, проверьте правильность ее местонахождения (должна быть выделена папка **Входящие**) и подтвердите действия кнопкой **ОК** (или нажав **Enter**).

2. Проверьте наличие новых сообщений, воспользовавшись пунктом меню



Сервис - Доставить почту или соответствующей кнопкой на панели инструментов.

3. Сохраните одно из поступивших (например, с темой «**Ваше мнение?**») сообщений в папке **Моя корреспонденция**:

- выделите соответствующее сообщение;
- указав на него, вызовите контекстное меню, нажав правую кнопку мыши;



Примечание: В контекстном меню представлены возможные действия с почтовым сообщением. Проанализируйте назначение основных (Открыть, Ответить отправителю, Переслать, Переместить (скопировать) в папку, Удалить) и спрогнозируйте результат их выполнения. в контекстном меню выберите пункт **Переместить в папку**; укажите на папку **Моя корреспонденция** для сохранения в ней почтового сообщения; подтвердите действия клавишей **ОК**.

4. Удалите одно из ненужных сообщений (по согласованию с преподавателем).

Упражнение 3. Формирование подписи к электронному сообщению.

Цель упражнения:

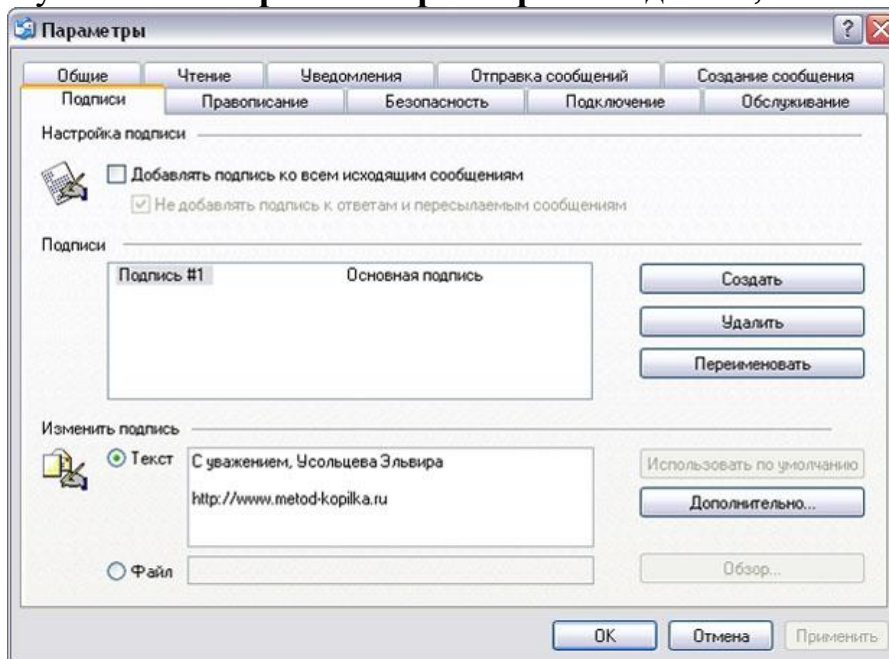
Формирование навыков оптимизации подготовки сообщений за счет автоматического добавления подписи к отправляемым и пересылаемым сообщениям.

Задание:

Создайте собственную подпись, которая будет автоматически добавляться ко всем отправляемым сообщениям.

Порядок выполнения:

1. Выберите пункт меню **Сервис - Параметры - Подпись** ;



2. Введите текст подписи (желательно с указанием электронного почтового адреса).

3. Поставьте флажок **Добавлять подпись ко всем исходящим сообщениям** и снимите флажок **Не добавлять подпись к ответам и пересылаемым сообщениям**.

4. Подтвердите действия клавишей **ОК**. Теперь подпись будет добавляться автоматически ко всем отправляемым вами сообщениям. Убедиться в эффективности такого приема можно при выполнении последующих упражнений.

Упражнение 4. Создание и отправка почтовых сообщений.

Цель упражнения:

Формирование навыков подготовки и отправки электронных писем.

Задание:

Создайте почтовое сообщение, содержащее анонс мероприятий, проводимых в образовательном учреждении на следующей неделе, и перешлите на соседние компьютеры и компьютер преподавателя.

Порядок выполнения:

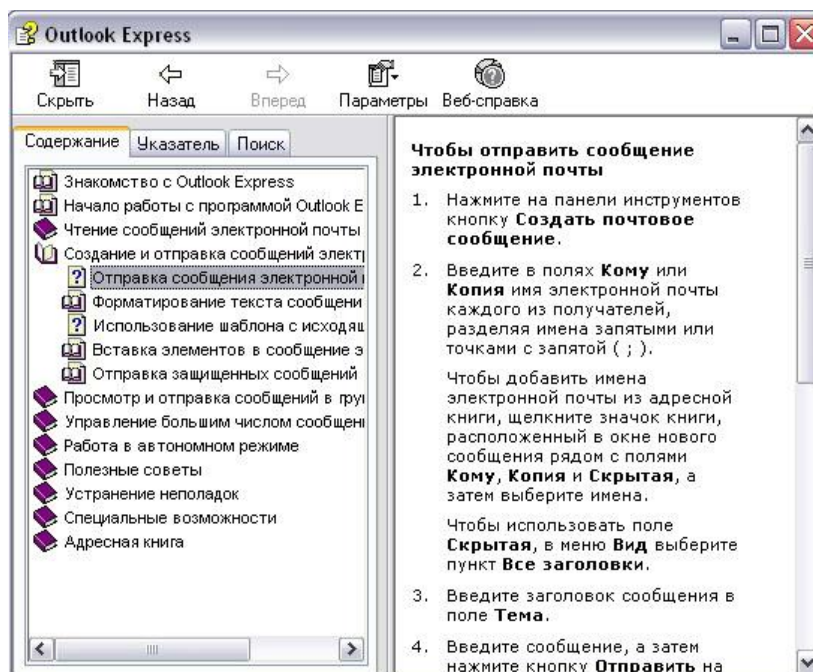
1. Выберите пункт меню **Сообщение - Создать** или воспользуйтесь



соответствующей кнопкой на панели инструментов;

2. **Примечание:** Более подробно изучить основы работы с почтовым клиентом можно, воспользовавшись встроенной системой помощи, вызвав ее по нажатию клавиши **F1** на функциональной клавиатуре или выполнив команду меню **Справка - Содержание и указатель**. Для выполнения данного упражнения целесообразно раздел **Создание и отправка почтовых сообщений**.

3.



2. Заполните все заголовки сообщения: **Кому**, **Копия**, **Скрытая**, **Тема** следующим образом: в заголовке **Кому** укажите электронный адрес преподавателя, **Копия** – адрес соседа слева, **Скрытая** – соседа справа. В качестве **Темы** укажите «**Анонс мероприятий МОУ СОШ №**».

Примечание. Если отсутствует заголовок **Скрытая**, то выберите пункт меню **Вид - Все заголовки**.

3. Впишите текст сообщения.

4. Отправьте сообщение, выполнив команду меню **Файл - Отправить** или



нажмите кнопку

Примечание. Проверьте, как выглядит сообщение, если его отправить в формате HTML. Для этого дайте команду **Формат - Формат HTML**. Убедитесь, что в этом случае (в отличие от режима **Обычный текст**) в окне подготовки сообщения появляется дополнительная панель форматирования, элементы управления которой позволяют управлять выбором шрифта, его начертанием и цветом, оформлением маркированных и нумерованных списков и т.п.

Упражнение 5. Подготовка и отправление почтового сообщения на бланке с вложением.

Цель упражнения:

Формирование навыков отправления электронных документов в качестве вложения в почтовое сообщение.

Задание:

Созданный вами текстовый документ (например, приказ) отправьте в качестве вложения на соседние компьютеры и компьютер преподавателя.

Порядок выполнения:

1. Подготовьте текстовый документ, содержащий приказ о награждении победителей районной научно-практической конференции, и сохраните его на локальном диске **D :/ Приказы - prikaz_N.doc**.
2. Используя команду меню **Сообщение - Создать с использованием - Выбор бланка**, выберите фоновый рисунок для вашего сообщения.
3. В заголовке **Кому** укажите электронный адрес преподавателя, **Копия** – свой собственный адрес. Впишите текст сообщения. В качестве **Темы** укажите **«Итоги конференции»**.
4. В это письмо вложите для пересылки файл **D:/Приказы - prikaz_N.doc**. Для этого выполните команду меню **Вставка - Вложение файла** или воспользуйтесь



соответствующей кнопкой. Укажите местонахождение файла **D:/Приказы - prikaz_N.doc** и дайте команду **Вложить**.

5. Организуйте отправку сообщения.
6. Убедитесь, что сообщение с вложением находится у вас и в папке **Отправленные**, и в папке **Входящие** (так как копию вы адресовали на свой компьютер).

Упражнение 6. Сохранение документов, полученных в качестве почтовых вложений с электронной почтой.

Цель упражнения:

Формирование навыков сохранения электронных документов, вложенных в почтовое сообщение.

Задание:

Полученный вами в качестве вложения электронный документ сохраните на локальном диске компьютера.

Порядок выполнения:

1. Перейдите в папку **Входящие**. Выделите сообщение с темой **«Сохраните вложение!»** (обратите внимание на маркировку сообщений с вложением символом «скрепка»).
2. Выполните команду меню **Файл - Сохранить**.
3. В открывшемся диалоговом окне выделите сохраняемое вложение. С помощью кнопки **Обзор** выберите диск и папку (например, **D :/ Рабочая**), где будет сохранено вложение.
4. Отправьте преподавателю ответ с подтверждением получения вложения. Выполните команду меню **Сообщение - Ответить отправителю** или



воспользуйтесь соответствующей кнопкой на панели инструментов. Обратите внимание, что поля **Кому** и **Тема** заполняется автоматически.

5. Впишите текст и отправьте сообщение.

6. Проверьте результат сохранения вложения, воспользовавшись программой **Проводник**.

Упражнение 7. Пересылка почтовых сообщений.

Цель упражнения:

Формирование навыков оптимальных приемов транспортировки почтовых сообщений.

Задание:

Полученное вами почтовое сообщение перешлите новому адресату.

Порядок выполнения:

1. Перейдите в папку **Входящие**.
2. Выделите почтовое сообщение с темой «**Ознакомиться всем !**»:
3. Выберите пункт меню **Сообщение - Переслать** (можно воспользоваться



соответствующей кнопкой **Переслать** на панели инструментов).

4. Заполните поле **Кому**, вписав адрес соседа справа, и отправьте сообщение.

Упражнение 8*. Заполнение адресной книги.

Цель упражнения:

Формирование навыков заполнения электронных баз данных.

Задание:

Занесите в Адресную книгу новых абонентов.

Порядок выполнения:

1. Пополните **Адресную книгу**, воспользовавшись пунктом меню **Сервис -**



Адресная книга или соответствующей кнопкой **Адреса** на панели инструментов.

2. Внесите в **Адресную книгу** преподавателя и одного из «соседей». Для этого выполните команду **Файл - Создать контакт** (или щелкните левой кнопкой мыши на кнопке **Создать** и выберите пункт меню **Создать контакт**). Внимательно изучите вкладки, представленные в данном диалоговом окне. Обратите внимание на то, что в нем имеются средства для ввода как личной, так и служебной информации (для практической деятельности, как правило, достаточно заполнить лишь несколько полей на вкладке **Имя**).

3. Начните заполнение полей вкладки **Имя** с поля **Имя в книге**. Введите сюда такую запись, которую хотели бы видеть в списке контактов, например Сорокин И.И.;

4. Заполните поля **Фамилия** (Сорокин), **Имя** (Иван) и **Отчество** (Иванович);
5. В поле **Адреса электронной почты** введите его электронный адрес, например: metod-kopilka@mail.ru
6. Занесите введенные данные в **Адресную книгу**, нажав на кнопку **Добавить**.

Примечание. Если необходимо изменить внесенные данные, следует щелкнуть на записи правой кнопкой мыши, в контекстном меню выбрать пункт **Свойства** и перейти на вкладку **Имя**.

Упражнение 9. Создание группового адреса в адресной книге для проведения массовой рассылки.**

Цель упражнения:

Формирование навыков оптимальных приемов рассылки почтовой корреспонденции.

Задание:

Создайте в Адресной книге групповое имя для оптимизации рассылки корреспонденции.

Порядок выполнения:

1. Откройте Адресную книгу.
2. Выполните команду меню **Файл - Создать группу** (или щелкните левой кнопкой мыши на кнопке **Создать** и выберите пункт меню **Создать группу**).
3. В соответствующей форме введите **Название группы** (например, **Коллеги**).
4. С помощью кнопки **Выбрать** занесите в нее из адресной книги преподавателя и «соседа слева».
5. Воспользовавшись кнопкой **Создать контакт**, «соседа справа» одновременно занесите и в **Адресную книгу**, и в **группу**.
6. С помощью кнопки **Добавить** пополните группу еще двумя записями.
7. Подготовьте (**Файл - Создать сообщение**) и отправьте сообщение в группу (в поле **Кому** укажите название группы **Коллеги**).

Примечание. Просмотрите, как заполнено поле **Кому** данного сообщения в папке **Отправленные**.

Упражнение 10.** Настройка панели инструментов программы Microsoft Outlook Express.

Цель упражнения:

Формирование навыков оптимальной работы с почтовым клиентом.

Задание:

Расположите кнопки на панели инструментов в определенном порядке.

Порядок выполнения:

Примечание. Настройка панели инструментов осуществляется аналогично настройке панели инструментов в браузере **Internet Explorer**. Для этого

1. Вызовите контекстное меню, щелкнув правой кнопкой мыши на свободном месте панели инструментов.
2. В контекстном меню выберите пункт **Настройка**.
3. Выберите **Текст кнопки** (например, **Выводить подписи**) и **Размер значка** (например, **Мелкие значки**).
4. Из **Имеющихся кнопок** сформируйте **Панель инструментов**, используя кнопки **Добавить** и **Удалить**, в следующей последовательности: **Создать сообщение**, **Ответить**, **Ответить всем**, **Переслать – Разделитель – Печать**, **Удалить – Разделитель – Доставить почту – Разделитель – Адреса**. Для изменения порядка расположения кнопок используйте кнопки **Вверх** и **Вниз**.

Упражнение 11.** Изучение дополнительных возможностей программы Microsoft Outlook Express.

Цель упражнения:

Формирование навыков оптимальной работы с почтовым клиентом и самостоятельного изучения возможностей программы.

Задание:

Изучите сервисные возможности программы **Microsoft Outlook Express**.

Порядок выполнения:

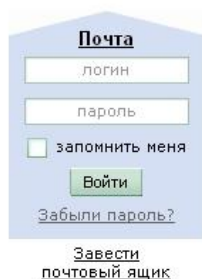
Выполните команду меню **Сервис - Параметры**. Не изменяя параметров, самостоятельно просмотрите основные возможности настройки программы, имеющиеся на вкладках:

- **Правописание ;**
- **Создание сообщения ;**
- **Отправка сообщений.**

Работа с электронной почтой на почтовых WWW-серверах (рекомендуется для самостоятельной работы)

Существует большое количество WWW -серверов, которые предлагают завести бесплатный почтовый ящик и позволяют работать с почтой, используя только браузер. Чтобы получить бесплатный почтовый ящик на таком сервере, необходимо зарегистрироваться. Для этого нужно заполнить несколько обязательных полей – ввести свой логин, пароль, возраст, пол и т.д. В случае успешной регистрации, за Вами будет закреплен бесплатный почтовый электронный адрес.

Упражнение 1 . Регистрация на бесплатном почтовом сервере.



Цель: Освоение приемов работы с электронной почтой через браузер. **Задание:** Зарегистрироваться на одном из бесплатных серверов www.yandex.ru, www.mail.ru, www.nm.ru, www.rambler.ru, www.ok.ru, www.pochta.ru, www.gmail.com, <http://www.nextmail.ru> и т.п.

Порядок выполнения:

1. Запустите программу **Internet Explorer** через кнопку **Пуск - Программы - Internet Explorer** или с помощью значка на Рабочем столе (Панели задач).
2. В адресной строке браузера введите адрес сайта www.yandex.ru.

3. Выберите ссылку **Почта - Зарегистрироваться** или **Завести почтовый ящик**.

4. Заполните форму регистрации

Примечание. Помните, что

- При введении **Вашего имени и Фамилии** будут предложены автоматически свободные логины, понравившийся вы можете выбрать или придумать собственный, который будет проверен почтовым сервером, занят ли он другим пользователем.
 - поля **Логин, Пароль и Подтверждение пароля** должны заполняться латинскими буквами, причем пароль должен содержать не менее 4-х символов;
 - обязательные поля для заполнения отмечены звездочками.
5. Подтвердите данные, нажав кнопку **Зарегистрировать**.
 6. После успешной регистрации появляется ваш личный адрес.
 7. Подтвердите согласие, нажав кнопку **Сохранить**.

Примечание:

Аналогично, можно зарегистрировать бесплатную почту на сайте www.mail.ru:

Упражнение 2. Знакомство с основными возможностями и элементами интерфейса Web-mail.

Цель упражнения:

Формирование первоначальных навыков оптимальной работы с электронными сообщениями на бесплатных почтовых серверах.

Задание:

Откройте свой почтовый ящик на бесплатном почтовом сервере и изучите основные элементы интерфейса.

Порядок выполнения:

Откройте свой почтовый ящик. Примерно так выглядит интерфейс вашего почтового ящика:



Примечание:

Папка **Входящие** содержит всю поступившую к вам корреспонденцию (на ваш почтовый ящик).

Папка **Отправленные** содержит всю отправленную вами другим адресатам в Internet корреспонденцию.

В папку **Рассылки** складываются письма, которые были одновременно разосланы большому числу пользователей.

Папка **Удаленные** хранит удаленные письма из любой другой папки.

Упражнение 3. Работа с почтовыми сообщениями.

Цель упражнения:

Освоение основных приемов оптимальной работы с электронной почтой на бесплатном почтовом сервере.

Задание:

- создайте и отправьте по электронной почте одно почтовое сообщение;
- напишите ответ на полученное письмо;
- создайте сообщение и вложите в него файл любого формата;
- сохраните вложенный в почтовое сообщение файл на локальном диске;
- полученное сообщение с вложением перешлите преподавателю.

Порядок выполнения:

1. Откройте свой почтовый ящик на бесплатном почтовом сервере, (например www.yandex.ru), введя логин и пароль в соответствующую форму:

Логин:

Пароль: [Забыли пароль?](#)

2. Создайте сообщение с темой «**Распоряжение**» :

- Щелкните по кнопке панели инструментов **написать автору**  [Написать](#) ;
- заполните заголовки сообщения: **Кому**, **Копия**, **Тема** следующим образом: в заголовке **Кому** укажите адрес преподавателя, **Копия** – адрес «соседа слева». В качестве **Темы** укажите «**Распоряжение**» ;
- впишите текст сообщения.

3. Отправьте сообщение с помощью кнопки **Отправить** или воспользовавшись соответствующей гиперссылкой.

4. Перейдите в папку **Входящие**. Для того, чтобы прочесть полученное сообщение, необходимо нажать на ссылку в поле **От кого**

5. В появившемся окне нажать на кнопку **Ответить**  [Ответить](#) . Напишите ответ на это письмо и нажмите на кнопку **Отправить**.

6. Создайте новое сообщение и вложите в него файл:

- в редакторе **Microsoft Word** создайте файл-проект приказа по персоналу с именем **prikaz_N.doc** и сохраните его в своем каталоге (**D:\Рабочая**);
- вернитесь в свой электронный ящик;

- щелкните по кнопке панели инструментов **Написать**  [Написать](#) ;
- заполните заголовки сообщения: **Кому**, **Копия**, **Тема** следующим образом: в заголовке **Кому** укажите адрес преподавателя, **Копия** – адрес «соседа справа». В качестве **Темы** укажите «**Приказ по персоналу**» ;

• нажмите на кнопку **Обзор**, укажите местонахождение файла (**D:\Рабочая**);

- напишите текст сообщения.

7. Отправьте сообщение, нажав на соответствующую кнопку.

8. Перейдите в папку **Входящие**. В списке сообщений найдите электронное письмо с темой «**Приказ по персоналу**», отправленное «соседом слева». Значок

в виде скрепки свидетельствует о наличии в полученном письме вложения. Сохраните вложенный файл в папке **D :\Рабочая\Ваша фамилия :**

- откройте полученное сообщение;
 - щелкните по значку вложенного файла левой кнопкой мыши;
 - в появившемся окне нажмите на кнопку **Сохранить** ;
 - укажите путь сохранения **D:\Рабочая\Ваша фамилия**.
9. Сообщение с темой **«Приказ по персоналу»** перешлите преподавателю.



- откройте нужное письмо и нажмите на кнопку **Переслать** Переслать ;
- заполните поле **Кому**, впишите электронный адрес преподавателя и отправьте сообщение.

Примечание. Такое использование почты имеет определенные достоинства. Можно легко менять провайдеров, не меняя свой адрес электронной почты. Можно просматривать почту с любого компьютера, подключенного к Интернету.

Разумеется, у такого способа есть и свои недостатки. вы не можете при работе с почтой через браузер минимизировать время подключения к Интернету в той мере, в какой это позволяют почтовые программы. Кроме того, общедоступные почтовые сервера часто перегружены.

Практическая работа 12. Файл как единица хранения информации на компьютере. Атрибуты файла и его объем. Учет объемов файлов при их хранении, передаче.

Цель занятия: осуществлять учет объемов файлов при их хранении, передаче.

Задание: Ознакомиться с теоретическими положениями по данной теме, выполнить задания практического занятия, сформулировать вывод.

Содержание отчета по результатам выполнения практического занятия

Отчет должен содержать:

1. Название работы
2. Цель работы
3. Результаты выполнения заданий
4. Вывод по работе (необходимо указать виды выполняемых работ, достигнутые цели, какие умения и навыки приобретены в ходе ее выполнения)

Методические указания к выполнению:

1. Краткие теоретические сведения

Файл - это определенное количество информации, имеющие имя, хранящиеся в долговременной памяти компьютера.

Имя файла разделено на две части точкой: имя файла (префикс) и расширение (суффикс), определяющее его тип (программа, данные и т.д.).

Имя файлу дает пользователь, а его тип обычно задается программой автоматически.

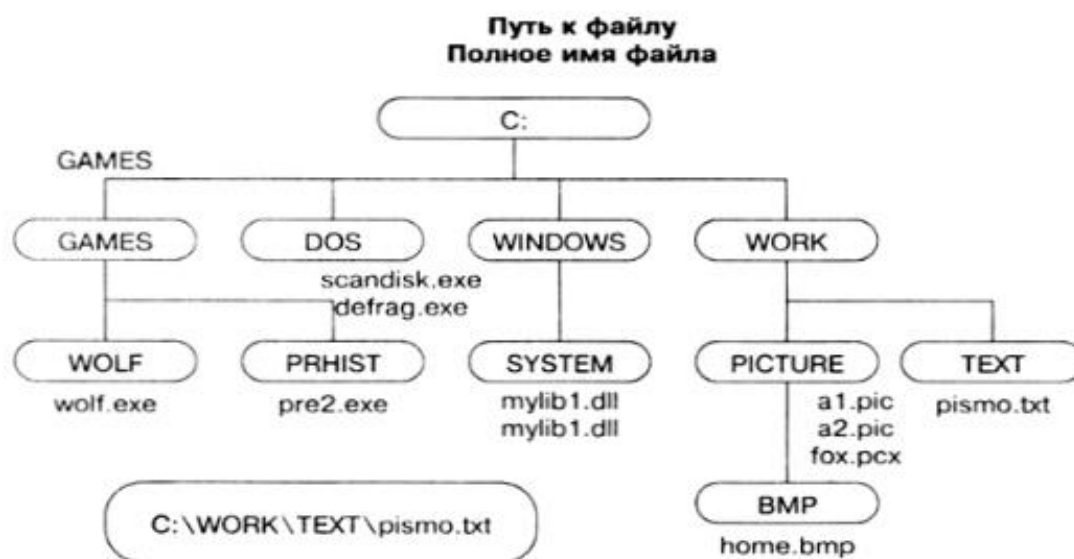
Таблица. Расширения в именах файлов

<i>Тип файла</i>	<i>Расширения</i>
Исполнимые файлы	.exe, .com, .bat
Текстовые файлы	.txt, .doc, .rtf
Графические файлы	.gif, .bmp, .jpg, .jpeg, .tif
Звуковые файлы	.wav, .midi, .mp3, .wma
Видеофайлы	.avi, .mpeg
Web-страницы	.htm, .html
Программы на языках программирования	.pas, .bas
Файлы данных	.dat, .dbf
Архиваторы данных	.arj, .rar, .zip

Файловая система - это функциональная часть операционной системы, обеспечивающая выполнение операций над файлами. Файловая система позволяет работать с файлами и директориями (каталогами) независимо от их содержимого, размера, типа и т. д.

Файловая система определяет общую структуру именования, хранения и организации файлов в операционной системе.

Иерархическая файловая система:



Над файлами могут производиться различные операции:

- Копирование (копия файла помещается из одного каталога в другой)
- Перемещение (сам файл перемещается в другой каталог)
- Удаление (запись о файле удаляется из каталога)
- Переименование (изменяется имя файла) и т.д.

Понятие «файл» включает не только хранимые им данные и имя, но и атрибуты. Атрибуты - это информация, описывающая свойства файла.

Примеры возможных атрибутов файла:

- Тип файла (обычный файл, каталог, специальный файл и т. п.) >
- Владелец файла ' - Создатель файла;
- Пароль для доступа к файлу;
- Информация о разрешенных операциях доступа к файлу;
- Время создания, последнего доступа и последнего изменения;
- Текущий размер файла;
- Максимальный размер файла;
- Признак «только «для чтения»;
- Признак «скрытый файл»;
- Признак «системный файл»;
- Признак «архивный файл»;
- Признак «двоичный / символьный»;
- Признак «временный файл» (удалить после завершения процесса);
- Признак блокировки;
- Длина записи в файле;
- Указатель на ключевое поле в записи;
- Длина ключа.

Набор атрибутов файла определяется спецификой файловой системы; в файловых системах разного типа для характеристики файлов могут

использоваться разные наборы атрибутов. Например, в файловых системах, поддерживающих неструктурированные файлы, нет необходимости использовать три последних атрибута в приведенном списке, связанных со структуризацией файла. В однопользовательской ОС в наборе атрибутов будут отсутствовать характеристики, имеющие отношение к пользователям и защите, такие как владелец файла, создатель файла, пароль для доступа к файлу, информация о разрешенном доступе к файлу.

Пользователь может получать доступ к атрибутам, используя средства, предоставленные для этих целей файловой системой. Обычно разрешается читать значения всех атрибутов, а изменять — только некоторые. Например, пользователь может изменить права доступа к файлу (при условии, что он обладает необходимыми для этого полномочиями), но изменять дату создания или текущий размер файла ему не разрешается;

Значения атрибутов файлов могут непосредственно содержаться в каталогах, как это сделано в файловой системе MS - DOS. Другим вариантом является размещение атрибутов в специальных таблицах, когда в каталогах содержатся только ссылки на эту таблицы. Такой подход реализован, например, в файловой системе ufs ОС UNIX. В этой файловой системе структура каталога очень простая. Запись о каждом файле содержит короткое символьное имя файла и указатель на индексный дескриптор файла, (так называется в ufs, таблица, в которой сосредоточены значения атрибутов файла).

В том и в другом вариантах каталоги обеспечивают связь между именами файлов и собственно файлами. Однако подход, когда имя файла отделено от его атрибутов, делает систему более гибкой. Например, файл может быть легко включен сразу в несколько каталогов. Записи об этом файле в разных каталогах могут содержать разные простые имена, но в поле ссылки будет указан один и тот же номер индексного дескриптора.

Логическая организация файлов

В общем случае данные, содержащиеся в файле, имеют некоторую логическую структуру. Эта структура является базой при разработке программы, предназначенной для обработки этих данных. Например, чтобы текст мог быть правильно выведен на экран, программа должна иметь возможность выделить отдельные слова, строки, абзацы и т. д. Признаками, отделяющими один структурный элемент от другого, могут служить определённые кодовые последовательности или просто известные программе значения смещений этих структурных элементов относительно начала файла. Поддержание структуры данных может быть либо целиком возложено на приложение, либо в той или иной степени эту работу может взять на себя файловая система.

В первом случае, когда все действия, связанные со структуризацией и интерпретацией содержимого файла, целиком относятся к ведению приложения, файл представляется ФС неструктурированной последовательностью данных. Приложение формулирует запросы к файловой системе на ввод - вывод, используя общие для всех приложений системные средства, например, указывая смещение от начала файла и количество байт которые необходимо считать или

записать. Поступивший к приложению поток байт интерпретируется в соответствии с заложенной в программе логикой. Например, компилятор генерирует, а редактор связей воспринимает вполне определённый формат объектного модуля программы.

При этом формат файла, в котором хранится объектный модуль, известен только этим программам. Подчеркнём, что интерпретация данных никак не связана с действительным способом их хранения в файловой системе.

Модель файла, в соответствии с которой содержимое файла представляется неструктурированной последовательностью (поток), байт, стала популярной вместе с ОС UNIX, а теперь она широко используется в большинстве современных ОС, в том числе в MS -DOS, Windows NT / 2000, Net Ware. Неструктурированная модель файла позволяет легко организовать разделение файла между несколькими приложениями: разные приложения могут по-своему структурировать и интерпретировать данные, содержащиеся в файле.

Другая модель файла, которая применялась в ОС OS/360, DBS RSX, VMS, а в настоящее время используется достаточно редко, - это структурированный файл. В этом случае поддержание структуры файла поручается файловой системе. Файловая система видит файл как упорядоченную последовательность логических записей. Приложение может обращаться к ФС с запросами на ввод - вывод на уровне записей, например, «считать запись 25 из файла File. Doc». ФС должна обладать информацией о структуре файла, достаточной для того, чтобы выделить любую запись. ФС предоставляет приложению доступ к записи, а вся дальнейшая обработка данных, содержащихся в этой записи, выполняется приложением. Развитием этого подхода стали системы управления базами данных (СУБД), которые поддерживают не только сложную структуру данных, но и взаимосвязи между ними.

Логическая запись является наименьшим элементом данных, которым может оперировать программист при организации обмена с внешним устройством. Даже, если физический обмен с устройством осуществляется большими единицами, операционная система должна обеспечивать программисту доступ к отдельной логической записи.

Файловая система может использовать два способа доступа к логическим записям: читать или записывать логические записи последовательно (последовательный доступ) или позиционировать файл на запись с указанным номером (прямой доступ).

Очевидно, что ОС не может поддерживать все возможные способы структурирования данных в файле, поэтому в тех ОС, в которых вообще существует поддержка логической структуризации файлов, она существует для небольшого числа широко распространенных схем логической организации файла.

К числу таких способов структуризации относится представление данных в виде записей, длина которых фиксирована в пределах файла. В таком случае доступ к некоторой записи А осуществляется либо путём последовательного

чтения ($A - 1$) предшествующих записей, либо прямо по адресу, вычисленному по её порядковому номеру. Например, если R – длина записи, то начальный адрес записи $A = R * A$. Заметим, что при такой логической организации размер записи фиксирован в пределах файла, а записи в разных файлах, принадлежащих одной и той же файловой системе, могут иметь разный размер.

Другой способ структуризации состоит в представлении данных в виде последовательности записей, размер которых изменяется в пределах одного файла. При такой организации данных для поиска нужной записи система должна последовательно считать все предшествующие записи. Вычислить адрес нужной записи по её номеру при такой логической организации невозможно, а, следовательно, не может быть применён более эффективный метод прямого доступа.

Файлы, доступ к записям которых осуществляется последовательно, по номерам позиций, называются неиндексированными, или последовательными.

Индексная логическая организация

Другим типом файлов являются индексированные файлы, они допускают более быстрый прямой доступ к отдельной логической записи. В индексированном файле записи имеют одно или более ключевых (индексных) полей и могут адресоваться путём указаний значений этих полей. Для быстрого поиска данных в индексированном файле предусматривается специальная индексная таблица, в которой значениям ключевых полей ставится в соответствие адрес внешней памяти. Этот адрес может указывать либо непосредственно на искомую запись, либо на некоторую область внешней памяти, занимаемую несколькими записями, в число которых входит искомая запись. В последнем случае говорят, что файл имеет индексно — последовательную организацию, так как поиск включает в себя два этапа: прямой доступ по индексу к указанной области диска, а затем последовательный просмотр записей в указанной области. Ведение индексных таблиц берёт на себя файловая система. Понятно, что записи в индексированных файлах могут иметь произвольную длину.

Всё вышесказанное в большей степени относится к обычным файлам, которые могут быть как структурированными, так и неструктурированными. Что же касается других типов файлов, то они обладают определённой структурой, известной файловой системе. Например, файловая система должна понимать структуру данных, хранящихся в файле - каталоге или файле типа «символьная связь».

Физическая организация файловой системы

Представление пользователя о файловой системе как об иерархически организованном множестве информационных объектов имеет мало общего с порядком хранения файлов на диске. Файл, имеющий образ цельного, непрерывающегося набора байт, на самом деле очень часто разбросан «кусочками» по всему диску, причем это разбиение никак не связано с логической структурой файла, например, его отдельная логическая запись может быть расположена в несмежных секторах диска. Логически объединённые файлы из одного каталога совсем не обязаны соседствовать на диске. Принципы

размещения файлов, каталогов и системной информации на реальном устройстве описываются физической организацией файловой системы. Очевидно, что разные файловые системы имеют разную физическую организацию.

Диски, разделы, секторы, кластеры

Основным типом устройства, которое используется в современных вычислительных системах для хранения файлов, являются дисковые накопители. Эти устройства предназначены для считывания и записи данных на жесткие и гибкие магнитные диски. Жёсткий диск состоит из одной или нескольких стеклянных или металлических пластин, каждая из которых покрыта с одной или двух сторон магнитным материалом; Таким образом, диск в общем случае состоит из пакета пластин.

На каждой стороне каждой пластины размечены тонкие концентрические кольца - дорожки (tracks) на которых хранятся данные. Количество дорожек зависит от типа диска. Нумерация дорожек начинается с 0 от внешнего края к центру диска. Когда диск вращается, элемент, называемый головкой, считывает двоичные данные с магнитной дорожки или записывает их на магнитную дорожку.

Головка может позиционироваться над заданной дорожкой. Головки перемещаются над поверхностью диска дискретными шагами, каждый шаг соответствует сдвигу на одну дорожку. Запись на диск осуществляется благодаря способности головки изменять магнитные свойства дорожки. В некоторых дисках вдоль каждой поверхности перемещается одна головка, а в других - имеется по головке на каждую дорожку. В первом случае для поиска информации головка должна перемещаться по радиусу диска. Обычно все головки закреплены на едином перемещающем механизме и двигаются синхронно. Поэтому, когда головка фиксируется на заданной дорожке одной поверхности, все остальные головки останавливаются над дорожками с такими же номерами. В тех же случаях, когда на каждой дорожке имеется отдельная головка, никакого перемещения головок с одной дорожки на другую не требуется, за счёт этого экономится время, затрачиваемое на поиск данных.

Совокупность дорожек одного радиуса на всех поверхностях всех пластин пакета называется цилиндром (cylinder). Каждая дорожка разбивается на фрагменты, называемые секторами (sectors), или блоками (blocks), так что все дорожки имеют равное число секторов, в которые можно максимально записать одно и то же число байт. Сектор имеет фиксированный для конкретной системы размер, выражающийся степенью двойки. Чаще всего размер сектора составляет 512 байт. Учитывая, что дорожки разного радиуса имеют одинаковое количество секторов, плотность записи становится тем выше, чем ближе дорожка к центру.

Сектор наименьшая адресуемая единица обмена данными дискового устройства с оперативной памятью. Для того чтобы контроллер мог найти на диске нужный сектор, необходимо задать ему все составляющие адреса сектора: номер цилиндра, номер поверхности и номер сектора. Так как прикладной программе в общем случае нужен не сектор, а некоторое количество байт, не обязательно кратное размеру сектора, то типичный запрос включает чтение

нескольких секторов, содержащих требуемую информацию, и одного из двух секторов, содержащих наряду с требуемыми и избыточные данные.

Операционная система при работе с диском использует, как правило, собственную единицу дискового пространства, называемую кластером (cluster). При создании файла на диске место ему выделяется кластерами. Например, если файл имеет размер 2560 байт, а размер кластера в файловой системе определён в 1024 байта, то файлу будет выделено на диске три кластера.

Дорожки и секторы создаются в результате выполнения процедуры физического, или низкоуровневого, форматирования диска, предшествующей использованию диска. Для определения границ блоков на диск записывается идентификационная информация. Низкоуровневый формат диска не зависит от типа операционной системы, которая этот диск будет использовать.

Разметку диска под конкретный тип файловой системы выполняют процедуры высокоуровневого, или логического, форматирования. При высокоуровневом форматировании определяется размер кластера и на диск записывается информация, необходимая для работы файловой системы, в том числе информация о доступном и неиспользуемом пространстве, о границах областей, отведённых под файлы и каталоги, информация о повреждённых областях. Кроме того, на диск записывается загрузчик операционной системы - небольшая программа, которая начинает процесс инициализации операционной системы после включения питания или рестарта компьютера.

Прежде чем форматировать диск под определённую файловую систему, он может быть разбит на разделы. Раздел - непрерывная часть физического диска, которую операционная система представляет пользователю как логическое устройство (используются также названия логический диск и логический раздел). Логическое устройство функционирует так, как если бы это был отдельный физический диск. Именно с логическими устройствами работает пользователь, обращаясь к ним по символьным именам, используя, например, обозначения a, b, s, sys и т. п. Операционные системы разного типа используют единое для всех них представление о разделах, но создают на его основе логические устройства, специфические для каждого типа ОС. Так как файловая система, с которой работает одна ОС, в общем случае не может интерпретироваться ОС другого типа, логические устройства не могут быть использованы операционными системами разного типа. На каждом логическом устройстве может создаваться только одна файловая система.

В частном случае, когда всё дисковое пространство охватывается одним разделом, логическое устройство представляет физическое устройство в целом. Если диск разбит на несколько разделов, то для каждого из этих разделов может быть создано отдельное логическое устройство. Логическое устройство может быть создано и на базе нескольким разделов, причём эти разделы не обязательно должны принадлежать одному физическому устройству. Объединение нескольких разделов в единое логическое устройство может выполняться разными способами и преследовать разные цели, основные из которых

увеличение общего объёма логического раздела, повышение производительности и отказоустойчивости. Примерами организации совместной работы нескольких дисковых разделов являются так называемые RAID - МАССИВЫ, подробнее о которых будет сказано ниже.

На разных логических устройствах одного и того же физического диска могут располагаться файловые системы разного типа.

Все разделы одного диска имеют одинаковый размер блока, определённый для данного диска в режиме низкоуровневого форматирования. Однако и в результате высокоуровневого форматирования в разных разделах одного и того же диска, предоставленных разными логическими устройствами, могут быть установлены различные файловые системы, в которых определены кластеры разных размеров.

Операционная система может поддерживать разные статусы разделов, особым образом отмечая разделы, которые могут быть использованы для загрузки модулей операционной системы, и разделы, в которых можно устанавливать только приложения и хранить файлы данных. Один из разделов диска помечается как загружаемый (или активный). Именно из этого раздела считывается загрузчик операционной системы.

Правила создания имени файла:

1. Нельзя использовать следующие символы, которые зарезервированы для специальных функций: ? . , ; : = + * / \ “ | < > [] ПРОБЕЛ

2. В длинных именах нельзя использовать следующие символы: ? : * / \ “ > < |

Задание 1

Предложите варианты имен и типов для перечисленных ниже файлов. Перенесите в тетрадь таблицу и заполните ее.

Содержание	Имя	Тип	Полное имя файла
Открытое письмо Биллу Гейтсу	BillG	doc	BillG.doc
Фото моей семьи			
Рецепт яблочного пирога			
Буклет «Мой колледж»			
Семейный альбом «Моя родословная»			
Репродукция картины Малевича «Черный квадрат»			
Петиция директору колледжа об увеличении числа уроков информатики			
Реферат по истории			
Реклама концерта рок-группы			

Задание 2. Предложите варианты программ, открывающих файлы с тем или иным расширением. Перечертите таблицу в тетрадь и заполните ее.

Расширение имени файла	Программа
TXT	
DOC	
RTF	
BMP	
ARJ	
HTML	

«неудовлетворительно» - менее 45%.

Задание 3.

Выполните задания в тетради.

- Придумай имя текстового файла, в котором будет содержаться информация о твоём доме. Подчеркни собственное имя файла.
- Придумай имя графического файла, в котором будет содержаться рисунок твоего дома. Подчеркни расширение файла.
- Выпиши в один столбик правильные имена файлов, а во второй правильные имена каталогов:

Письмо.18, letter.txt, WinWord, письмо.doc, Колледж?12, Мои документы, роза.bmp, crop12.exe, 1C, red.com

Вопросы для самоконтроля:

- Назовите основные возможности архиваторов?
- Что такое электронный архив?
- В чём суть разархивирования?
- Как задать имя файла?
- Что такое файловая система?
- Что такое иерархическая файловая система?
- Какие операции могут производиться над файлами?

СПИСОК ЛИТЕРАТУРЫ

- 1 Алексеев В. А. Информатика. Практические работы. М.: Лань, 2024. 256 с.
- 2 Астафьева С. А. Информатика и ИКТ. Практикум для профессий и специальностей технического и социально-экономического профилей. / Н. Е. Астафьева. С.А. Гаврилова, М.С. Цветкова под ред. М.С. Цветковой, Издательский центр «Академия», 2020.- 272 с.
- 3 Волк В. К. Информатика. М.: Юрайт, 2024. 227 с.