

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Плиоджинович
Должность: И.о. ректора
Дата подписания: 19.08.2023 23:10:02
Уникальный программный ключ:
2a04bb882d7edb71479cb266eb4aaaae0eb0ee849

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФГБОУ ВПО
«Дагестанский государственный технический университет»

УЧЕБНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ №№3,4 по дисциплине «Информационные технологии» для студентов направления подготовки бакалавров 27.03.04 – «Управление в технических системах»

Махачкала 2019

УДК 004(07)

Методические указания к выполнению лабораторных работ №№3,4 по дисциплине «Информационные технологии» для студентов направления подготовки бакалавров 27.03.04 – «Управление в технических системах». – Махачкала, 2019. – 33 с.

В методических указаниях к выполнению лабораторных работ №№3,4 по дисциплине «Информационные технологии» для студентов направления подготовки бакалавров 220400.62 – «Управление в технических системах» рассматриваются основные классы современных технологий в электронном документообороте, содержатся методики выполнения лабораторных работ направленных на освоение основных программных продуктов, используемых в учрежденческой деятельности. Так же рассматриваются приемы работы и основные функции табличного процессора Microsoft Excel 2000 для работы с документами и основные процедуры программирования HTML – документов для создания рекламных материалов в Internet с помощью программы Frontpage Express.

Составитель: к.т.н., доцент Мусаева У.А.

Рецензенты: к.ф.-м.н., доцент, зав.кафедрой ПМиИ, Исабекова Т.И.

инженер ОАО НИИ «Сапфир» Гасанов О.И.

СОДЕРЖАНИЕ

Лабораторная работа № 3	
Работа с базами данных в среде СУБД Microsoft Access 2000.....	4
Лабораторная работа № 4	
Глобальная сеть Internet и создание Web-документов.....	19
Список литературы.....	33

ЛАБОРАТОРНАЯ РАБОТА №3

Работа с базами данных в среде СУБД Microsoft Access 2000

1. Цель работы

Целью работы является изучение средств разработки основных объектов базы данных Microsoft Access 2000, знакомство с конкретными приемами работы и с особенностями ее автоматизации.

2. Основные теоретические сведения.

Системы управления базами данных (СУБД) – это специальные программные средства, предназначенные для работы с файлами баз данных (файлами специального формата, содержащими информацию, структурированную заданным образом).

Современные СУБД позволяют хранить в виде файлов данные любых типов: числовые, текстовые, графические, звуковые, видео и прочие. Данные в базах хранятся в виде *таблиц*. Каждая таблица имеет *структуру*.

Структура таблицы определяется составом ее *полей* и их *свойствами*. Важнейшими свойствами полей являются: *тип поля* и *размер поля*. Для хранения разных *типов данных* используют *поля соответствующих типов*.

Данные, хранящиеся в таблице, можно изменять, удалять, сортировать, фильтровать, размножать и выполнять с ними другие операции.

Для автоматизации операций по работе с данными, в частности, для отбора нужных данных, применяют специальные объекты, которые называются *запросами*.

В СУБД Access запросы создают с помощью специального *бланка запроса по образцу*.

С помощью *запросов на выборку* производят выбор данных из базы, их обработку, выполнение итоговых вычислений и другие операции. По результатам работы запроса создается временная *результатирующая таблица*.

На основе результирующей таблицы, можно создавать новые таблицы или изменять существующие. Для этого служат *запросы на изменение*.

Для ввода данных в таблицы или для просмотра данных в наглядной форме служат специальные объекты, называемые *формами*. Формы – экранные объекты.

Структура форм состоит из *разделов* и *элементов управления*.

Проектирование формы состоит в размещении элементов управления на бланке формы и в задании связей между этими элементами и полями таблиц или запросов базы данных.

Создание форм можно выполнять автоматически (*автоформы*), полуавтоматически (с помощью *Мастера*) или вручную (в режиме *Конструктора*). Размещение элементов управления на бланке формы автоматизировано. В большинстве случаев при создании нового элемента запускается *программа-Мастер*, с помощью которой происходит настройка свойств элемента управления.

Для создания печатных документов, которые содержат информацию из базовых таблиц или из результирующих таблиц, полученных по результатам работы запросов, служат специальные объекты – *отчеты*.

Отчеты отличаются от форм тем, что предназначены не для ввода данных, а только для вывода, а также тем, что создают не экранные, а печатные документы.

Структура отчетов, как и форм, состоит из разделов и элементов управления. Проектирование отчета состоит в создании структуры его разделов и в размещении элементов управления внутри этих разделов, а также в задании связей между этими элементами и полями таблиц или запросов базы данных.

Создание отчетов может выполняться автоматически (*автоотчеты*), полуавтоматически (с помощью *Мастера*) или вручную (в режиме *Конструктора*).

Таблицы, запросы, формы и отчеты являются основными объектами базы данных. Их разрабатывает разработчик базы. Пользователь базы использует эти объекты без вмешательства в их структуру.

Разработчик базы данных имеет также два типа дополнительных объектов: *макросы* и *модули*. Эти объекты создают в тех случаях, когда стандартных средств управления базой данных оказывается недостаточно для выполнения операций, необходимых заказчику системы. С помощью *макросов* создают *макрокоманды*, упрощающие наиболее утомительные операции с базой, а с помощью *модулей*, написанных на языке программирования Visual Basic, создают программные процедуры для выполнения нестандартных операций.

СУБД Microsoft Access 2000 предоставляет несколько средств создания каждого из основных объектов базы данных. Эти средства можно классифицировать как:

- ручные (разработка объектов в режиме *Конструктора*);
- автоматизированные (разработка с помощью программ-мастеров);
- автоматические – средства ускоренной разработки простейших

объектов.

Соотношения между этими средствами понятны: ручные средства являются наиболее трудоемкими, но обеспечивают максимальную гибкость; автоматизированные и автоматические средства являются наиболее производительными, но и наименее гибкими. Методической особенностью изучения программы Microsoft Access является тот факт, что в учебных целях для создания разных объектов целесообразно пользоваться разными средствами:

1. при разработке учебных таблиц и запросов рекомендуется использовать ручные средства – работать в режиме *Конструктора*. Использование мастеров ускоряет работу, но не способствует освоению понятий и методов работы;

2. при разработке учебных форм, отчетов и страниц доступа наоборот лучше пользоваться автоматизированными средствами, предоставляемыми мастерами, так как для этих объектов большую роль играет внешний вид. Дизайн этих объектов весьма трудоемок, поэтому его лучше поручить программе, а студентам сосредоточиться на содержательной части работы.

2.1. Работа с таблицами

Создание таблиц. Работа с любыми объектами начинается с окна *База данных*. На левой панели данного окна сосредоточены элементы управления для вызова всех семи типов объектов программы. Создание таблиц начинается с выбора элемента управления *Таблицы*.

На правой панели представлен список таблиц, уже имеющихся в составе базы, и приведены элементы управления для создания новой таблицы. Чтобы создать таблицу вручную, следует использовать значок *Создание таблицы* в режиме *Конструктора*.

Окно *Конструктора таблиц* фактически является графическим бланком для создания и редактирования структуры таблиц. В первом столбце вводят имена полей. Если свойство *Подпись* для поля не задано, то *Имя поля* станет одновременно и именем столбца будущей таблицы. Тип для каждого поля выбирают из раскрывающегося списка, открываемого кнопкой выбора типа данных. Эта кнопка – скрытый элемент управления. Она отображается только после щелчка на поле бланка. Это надо иметь в виду – в Microsoft Access очень много таких скрытых элементов управления, которые не отображаются, пока ввод данных не начат.

Нижняя часть бланка содержит список свойств поля, выделенного в верхней части. Некоторые из свойств уже заданы по умолчанию.

Свойства полей не являются обязательными. Их можно настраивать по желанию, а можно и не трогать.

При создании таблицы целесообразно (хотя и не обязательно) задать поле. Это поможет впоследствии, при организации связей между таблицами. Для задания ключевого поля достаточно щелкнуть на его имени правой кнопкой мыши и в открывшемся контекстном меню выбрать пункт *Ключевое поле*.

Если первичный ключ необходим для связи с другими таблицами, но ни одно из полей не является уникальным, то первичный ключ можно создать на базе двух (или более полей). Эта операция выполняется точно так же, через контекстное меню. Групповое выделение выполняют при нажатой клавише SHIFT щелчками на квадратных маркерах слева от имен полей.

Закончив создание структуры таблицы, бланк закрывают (при этом система выдает запрос на сохранение таблицы), после чего дают таблице имя, и с этого момента она доступна в числе прочих таблиц в основном окне *База данных*. Оттуда ее и можно открыть в случае необходимости.

Созданную таблицу открывают в окне *База данных* двойным щелчком на ее значке. Новая таблица не имеет записей – только названия столбцов, характеризующие структуру таблицы. Заполнение таблицы данными производится обычным порядком. Курсор ввода устанавливается в нужную ячейку указателем мыши. Переход к следующей ячейке можно выполнить клавишей TAB. Переход к очередной записи выполняется после заполнения последней ячейки.

В нижней части таблицы расположена *Панель кнопок перехода*. Ее элементами управления удобно пользоваться при навигации по таблице, имеющей большое число записей.

Если данные не всегда уместаются в ячейках таблицы, то следует изменить ширину столбцов методом перетаскивания их границ. Удобно использовать автоматическое форматирование столбцов «по содержанию». Для этого надо установить указатель мыши на границу между столбцами (в строке заголовков столбцов), дождаться, когда указатель сменит форму, и выполнить двойной щелчок.

Это общесистемный прием Windows, и им можно пользоваться в данной программе, как и во многих других.

После наполнения таблицы данными сохранять их не надо – все сохраняется автоматически. Однако если при работе с таблицей произошло редактирование ее макета (например, изменялась ширина столбцов), СУБД попросит подтвердить сохранение этих изменений.

Если возникнет необходимость изменить структуру таблицы (состав полей или их свойства), таблицу надо открыть в режиме *Конструктора*. Для этого ее следует выделить в окне *База данных* и щелкнуть на кнопке *Конструктор*.

Если на этапе проектирования базы данных была четко разработана структура таблиц, то создание таблиц с помощью *Конструктора* происходит очень быстро и эффективно. Даже без использования автоматизированных средств создание основы для достаточно крупных проектов происходит в считанные минуты – это ценное свойство СУБД Microsoft Access, но оно реализуется при обязательном условии тщательной предварительной подготовки.

Создание межтабличных связей. Если структура базы данных продумана заранее, а связи между таблицами намечены, то создание реляционных отношений между таблицами выполняется очень просто. Вся необходимая работа происходит в специальном окне *Схема данных* и выполняется с помощью мыши. Окно *Схема данных* открывают кнопкой на панели инструментов или командой *Сервис / Схема данных* (если в меню *Сервис* не видно соответствующего пункта, следует раскрыть расширенную часть меню).

Образовавшаяся межтабличная связь отображается в окне *Схема данных* в виде линии, соединяющей два поля разных таблиц. При этом одна из таблиц считается *главной*, а другая – *связанной*. Главная – это та таблица, которая участвует в связи своим ключевым полем (название этого поля на схеме данных отображается полужирным шрифтом). У связи два основных назначения. Первое – обеспечение целостности данных, а второе – автоматизация задач обслуживания базы. Представим себе, что в таблице *Клиенты*, где каждый клиент уникален, кто-то удалит запись для одного из клиентов, но не сделает этого в таблице *Заказы*. Получится, что согласно таблице заказы некто, не имеющий ни имени, ни адреса, а только абстрактный код, делал заказы. Узнать по коду, кто же это был на самом деле, будет невозможно – произошло нарушение целостности данных.

В данном случае владелец базы может применить три подхода: либо вообще ничего не делать для защиты целостности данных, либо запретить удаление данных из ключевых полей главных таблиц, либо разрешить его, но при этом адекватно обработать и связанные таблицы. Вручную сделать это чрезвычайно трудно, поэтому и нужны средства автоматизации.

Связь между таблицами позволяет:

- либо исключить возможность удаления или изменения данных в клю-

чевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц;

- либо сделать так, что при удалении (или изменении) данных в ключевом поле главной таблицы автоматически (и абсолютно корректно) произойдет удаление или изменение соответствующих данных в полях связанных таблиц.

Для настройки свойств связи надо в окне *Схема данных* выделить линию, соединяющую поля двух таблиц, щелкнуть на ней правой кнопкой мыши и открыть контекстное меню связи, после чего выбрать в нем пункт *Изменить связь* – откроется диалоговое окно *Изменение связи*. В нем показаны названия связанных таблиц и имена полей, участвующих в связи (здесь же их можно изменить), а также приведены элементы управления для обеспечения условий целостности данных.

Если установлен только флажок *Обеспечение целостности данных*, то удалять данные из ключевого поля главной таблицы нельзя. Если вместе с ним включены флажки *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*, то, соответственно, операции редактирования и удаления данных в ключевом поле главной таблицы разрешены, но сопровождаются автоматическими изменениями в связанной таблице.

Таким образом, смысл создания реляционных связей между таблицами состоит, с одной стороны, в защите данных, а с другой стороны – в автоматизации внесения изменений сразу в несколько таблиц при изменениях в одной таблице.

3. Порядок выполнения работы

Задание 1. Создание базовых таблиц

Руководитель предприятия, выполняющего сборку компьютеров из готовых компонентов, заказал разработку базы данных, основанной на двух таблицах комплектующих. Одна таблица содержит данные, которые отображаются для клиентов при согласовании спецификации изделия, – в ней указаны розничные цены на компоненты. Вторая таблица нужна для анализа результатов деятельности предприятия – в ней содержатся оптовые цены на компоненты и информация о поставщиках (клиенты предприятия не имеют доступа к данным этой таблицы).

1. Запустите программу Microsoft Access 2000.

2. В окне Microsoft Access включите переключатель *Новая база данных* и щелкните на кнопке ОК.

3. В окне *Файл новой базы данных* выберите папку */Мои документы* и дайте файлу имя: *Комплектующие*. Убедитесь, что в качестве типа

файла выбрано *Базы данных Microsoft Access*, и щелкните на кнопке *Создать*. Откроется окно новой базы – *Комплектующие: база данных*.

4. Откройте панель *Таблицы*.

5. Дважды щелкните на значке *Создание таблицы в режиме конструктора* – откроется бланк создания структуры таблицы.

6. Для первой таблицы введите следующие поля:

Имя поля	Тип поля
Компонент	Текстовый
Модель	Текстовый
Основной параметр	Числовой
Цена	Числовой

Обратите внимание на то, что в данном случае поле *Цена* задано не денежным типом, а числовым. *Данные*, относящиеся к денежному типу, имеют размерность, выраженную в рублях. Но стоимость компонентов вычислительной техники выражать в этой единице измерения не принято. Для сравнимости цен разных поставщиков обычно используют «условные единицы». В таких случаях удобно использовать поле числового типа, чтобы не перенастраивать всю СУБД.

7. Щелкните на поле *Цена*. В нижней части бланка задайте свойство *Число десятичных знаков*, равным 2.

8. Для связи с будущей таблицей поставщиков надо задать ключевое поле. Поскольку здесь ни одно поле явно не претендует на «уникальность», используем комбинацию полей *Компонент* и *Модель*. Выделите оба поля в верхней части бланка (при нажатой клавише SHIFT). Щелчком правой кнопки мыши откройте контекстное меню и выберите в нем пункт *Ключевое поле*.

9. Закройте окно *Конструктора*. При закрытии окна дайте таблице имя *Комплектующие*.

10. Повторив действия пунктов 5-9, создайте таблицу *Поставщики*, в которую входят следующие поля.

Имя поля	Тип поля
Компонент	Текстовый
Модель	Текстовый
Цена оптовая	Числовой
Поставщик	Текстовый

Телефон	Текстовый
Адрес	Текстовый
Примечание	Поле МЕМО

Обратите внимание на то, что поле номера телефона является текстовым, несмотря на то, что обычно номера телефонов записывают цифрами. Это связано с тем, что они не имеют числового содержания. Номера телефонов не сравнивают по величине, не вычитают из одного номера другой и т. д. Это типичное текстовое поле. Ключевое поле можно не задавать – для текущей задачи оно не требуется.

11. В окне *Комплектующие: база данных* откройте по очереди созданные таблицы и наполните их экспериментальным содержанием (3-4 записи). Закончив работу, закройте таблицы и завершите работу с программой.

Задание 2. Создание межтабличных связей

1. Запустите программу Microsoft Access 2000.

2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке ОК.

3. В окне *Комплектующие: база данных* откройте панель *Таблицы*. Убедитесь, что на ней присутствуют значки ранее созданных таблиц *Комплектующие* и *Поставщики*.

4. Разыщите на панели инструментов кнопку *Схема данных*. Если есть сложности, найдите команду строки меню: *Сервис / Схема данных*. Воспользуйтесь любым из этих средств, чтобы открыть окно *Схема данных*. Одновременно с открытием этого окна открывается диалоговое окно *Добавление таблицы*, на вкладке *Таблицы* которого можно выбрать таблицы, между которыми создаются связи.

5. Щелчком на кнопке *Добавить* выберите таблицы *Комплектующие* и *Поставщики* – в окне *Схема данных* откроются списки полей этих таблиц.

6. При нажатой клавише SHIFT выделите в таблице *Комплектующие* два поля – *Компонент* и *Модель*.

7. Перетащите эти поля на список полей таблицы *Поставщики*. При отпускании кнопки мыши автоматически откроется диалоговое окно *Изменение связей*.

8. На правой панели окна *Изменение связей* выберите поля *Компонент* и *Модель* таблицы *Поставщики*, включаемые в связь. Не устанавливайте флажок *Обеспечение целостности данных* – в данном упраж-

нении это не требуется, но может препятствовать постановке учебных опытов с таблицами.

9. Закройте диалоговое окно *Изменение связей* и в окне *Схема данных* рассмотрите образовавшуюся связь. Убедитесь в том, что линию связи можно выделить щелчком левой кнопки мыши, а щелчком правой кнопки мыши открывается контекстное меню, позволяющее разорвать связь или отредактировать ее.

10. Закройте окно *Схема данных* и программу Microsoft Access.

Задание 3. Создание запроса на выборку

Нужно создать запрос на выборку жестких дисков, с емкостью не менее 8 Гбайт при цене менее 150 условных единиц. Результирующая таблица должна содержать адрес поставщика и номер его телефона.

1. Запустите программу Microsoft Access 2000.

2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплекующие* и щелкните на кнопке ОК.

3. В окне *Комплекующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса в режиме Конструктора* – откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно *Добавление таблицы*.

4. В окне *Добавление таблицы* выберите таблицу *Поставщики* и щелкните на кнопке *Добавить*. Закройте окно *Добавление таблицы*.

5. В списке полей таблицы *Поставщики* выберите поля, включаемые в результирующую таблицу: *Компонент*, *Модель*, *Цена оптовая*, *Поставщик*, *Телефон*. Выбор производите двойными щелчками на именах полей.

6. Задайте условие отбора для поля *Компонент*. В соответствующую строку введите: *Жесткий диск*. Из таблицы будут выбираться не все изделия, а только жесткие диски.

7. Задайте условие отбора для поля *Цена оптовая*. В соответствующую строку введите: *<150*. Из таблицы будут выбираться только изделия, имеющие цену менее 150 условных единиц.

8. Необходимо задать условие отбора по основному потребительскому параметру – емкости жесткого диска. Однако в таблице *Поставщики* такого поля нет. С другой стороны, в ней есть поле *Модель*, которое однозначно определяет параметры изделия. Благодаря тому, что по полю *Модель* установлена связь с таблицей *Комплекующие*, мы получаем возможность ввести в запрос поле *Основной параметр*, взяв его из другой таблицы.

Добавьте список полей таблицы *Комплектующие* в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка и в открывшемся контекстном меню выберите пункт *Добавить таблицу* – откроется окно *Добавление таблицы*. Выберите в нем таблицу *Комплектующие*.

9. Двойным щелчком на поле *Основной параметр* в списке полей таблицы *Комплектующие* введите это поле в бланк запроса по образцу.

10. В строке *Условие отбора* столбца *Основной параметр* введите условие > 8 (емкость диска более восьми гигабайт).

11. Закройте бланк запроса по образцу. При закрытии запроса введите его имя – *Выбор комплектующих*.

12. В окне *Комплектующие: база данных* откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержимое зависит от того, что было введено в таблицы *Комплектующие* и *Поставщики* при их наполнении в задании 1. Если ни одно изделие не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.

13. По окончании исследований закройте все открытые объекты и завершите работу с программой Microsoft Access.

Задание 4. Создание запросов «с параметром»

Выше мы рассмотрели, как действует условие отбора, но должны отметить его существенный недостаток. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий изделия, имеющие цену менее 150 условных единиц, то пользователь базы уже не в состоянии отобрать изделия, цена которых менее 140 условных единиц, поскольку у него нет соответствующего запроса.

Специальный тип запросов, называемый запросами «с параметром», позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этим приемом обеспечивается гибкость работы с базой данных.

Создайте простой запрос, для отбора процессоров, предельную цену которых пользователь может задать сам при запуске запроса.

1. Запустите программу Microsoft Access 2000.

2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплекующие* и щелкните на кнопке ОК.

3. В окне *Комплекующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса в режиме Конструктора* – откроется бланк запроса по образцу.

4. Согласно упражнению 3, создайте запрос на выборку, основанный на таблице *Поставщики*, в который войдут следующие поля:

- Компонент;
- Модель;
- Цена оптовая;
- Поставщик;
- Телефон;

5. В строке *Условие отбора* поля *Компонент* введите: *Процессор*.

6. Строку *Условие отбора* для поля *Цена оптовая* надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение. Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 единиц, мы бы написали: >100 , Если бы нам были нужны процессоры дешевле 80 единиц, мы бы написали <80 . Но если мы хотим дать пользователю возможность выбора, мы должны написать: $< [Введите максимальную цену]$.

7. Закройте запрос. При закрытии сохраните его под именем *Выбор комплектующих*.

8. В окне *Комплекующие: база данных* откройте панель *Запросы* и запустите запрос *Выбор комплектующих* – на экране появится диалоговое окно *Введите значение параметра*.

9. Введите какое-либо число и щелкните на кнопке ОК. В зависимости оттого, что реально содержится в таблице *Поставщики*, по результатам запроса будет сформирована результирующая таблица.

10. Закройте все объекты базы данных. Закройте программу Microsoft Access.

Задание 5. Создание итогового запроса

Если полностью заполнить данными таблицу *Комплекующие*, введя параметры всех компонентов, входящих в сборочную спецификацию персонального компьютера, то можно узнать, во что обходится себестоимость комплектующих узлов. Запросы, выполняющие вычисления по всем записям для какого-либо числового поля, называются *ито-*

говыми запросами. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле, может также исполняться иная *итоговая функция*. Итоговые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу,

Предположим, что малое предприятие собирает компьютеры трех классов; «Элитный», «Деловой» и «Экономичный». Несмотря на то, что архитектура у всех компьютеров близка, их компоненты заметно отличаются по цене и техническим параметрам. Соответственно, имеются значительные отличия в цене этих трех моделей, что важно для захвата разных секторов рынка. Наша задача – подготовить итоговый отчет, с помощью которого можно определять цену каждой из модели компьютеров и динамично ее отслеживать при изменении входящих компонентов или их поставщиков.

1. Запустите программу Microsoft Access 2000.

2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке ОК.

3. В окне *Комплектующие: база данных* откройте панель *Таблицы*. Выберите таблицу *Комплектующие*.

4. Щелчком на значке *Конструктор* откройте таблицу в режиме проектирования – нам это необходимо для создания дополнительного поля *Класс*, в котором будут храниться данные о том, для какого класса изделий предназначены компоненты.

5. В начало структуры таблицы вставьте новое поле. Для этого выделите первое поле (*Компонент*) и нажмите клавишу INSERT.

6. Введите имя нового поля – *Класс* и определите его тип – *Текстовый*.

7. Закройте окно *Конструктора*. При закрытии подтвердите необходимость изменить структуру таблицы.

8. Откройте таблицу *Комплектующие* и наполните ее содержанием, введя для каждого класса данные по следующим изделиям:

- Материнская плата;
- Процессор;
- Оперативная память;
- Жесткий диск;
- Корпус;
- Дисковод CD-ROM;

- Дисковод гибких дисков;
- Видеоадаптер;
- Звуковая карта;
- Клавиатура;
- Мышь.

Цены на эти изделия для каждого класса проставьте произвольно. Прочие поля таблицы можно не заполнять – в формировании итогового запроса они участвовать не будут.

9. Закройте таблицу *Комплектующие*.

10. Откройте панель *Запросы* щелчком на одноименной кнопке окна *Комплектующие: база данных*.

11. Выполните двойной щелчок на значке *Создание запроса в режиме конструктора*. В открывшемся диалоговом окне *Добавление таблицы* выберите таблицу *Комплектующие*, на основе которой будет разрабатываться итоговый запрос. Закройте окно *Добавление таблицы*.

12. В бланк запроса по образцу введите следующие поля таблицы *Комплектующие: Класс, Компонент, Цена*.

13. Для поля *Класс* включите сортировку по возрастанию. Включите также сортировку по полю *Цена*, но на этот раз – по убыванию.

14. На панели инструментов Microsoft Access щелкните на кнопке *Групповые операции* или воспользуйтесь строкой меню (*Вид / Групповые операции*). Эта команда необходима для создания в нижней части бланка строки *Групповые операции*. Именно на ее базе и создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение *Группировка*.

15. Для поля, по которому производится группировка записей (в нашем случае – *Класс*), оставьте в строке *Групповые операции* значение *Группировка*. Для остальных полей щелкните в этой строке – появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле.

16. Для поля *Цена* выберите итоговую функцию *Sum* для определения стоимости изделия как суммы стоимостей комплектующих.

17. Для поля *Компонент* выберите итоговую функцию *Count*, определяющую общее количество записей, вошедших в группу. В нашем случае это количество узлов, из которых собран компьютер.

18. Закройте бланк запроса по образцу и дайте ему имя: *Расчет стоимости изделия*. Запустите запрос и убедитесь, что он правильно работает,

19. Закройте все объекты базы данных. Завершите работу с программой Microsoft Access.

4. Контрольные вопросы

1. В каких *объектах* хранятся данные базы?
2. Чем отличаются *поля* и *записи* таблицы?
3. Какие *типы полей* таблиц вы знаете?
4. Какое поле можно считать *уникальным*?
5. Каким должно быть поле *первичного ключа*?
6. В каком диалоговом окне создают связи между полями таблиц базы данных?
7. Для чего предназначены *запросы*?
8. Запросы баз данных обращены к таблицам. Как называются таблицы, на базе которых основан запрос?
9. Как называется таблица, образуемая результате работы запроса?
10. Как называется категория запросов, предназначенная для выбора данных из таблиц базы?
11. Как называется бланк, заполнением которого создается структура запроса?
12. Как называется поле, данные в котором образуются в результате расчета с участием данных, содержащихся в других полях?
13. Как называется категория запросов, выполняющих *итоговые вычисления*?
14. Какие *итоговые функции* вы знаете?
15. Как с помощью запроса провести сортировку записей в таблице по какому-либо полю?
16. Для чего предназначены *формы*?
17. Какие методы автоматического создания форм вы знаете?
18. Из чего состоят формы?
19. Какие *разделы* форм вы знаете?
20. Какие *элементы управления* форм вы знаете?
21. При создании элемента управления формы рядом с ним автоматически образуется элемент управления, называемый *присоединенной надписью*. Как оторвать эту надпись от элемента управления, если разработчик не хочет, чтобы она располагалась слева, а желает поместить ее ниже элемента управления?
22. Какой командой можно выровнять элементы управления формы, расположенные неаккуратно?
23. Для чего предназначены *отчеты*?
24. Из чего состоят отчеты?
25. Какие *разделы* отчетов вы знаете?
26. Что общего и в чем различие между разделами отчетов и разделами форм?

27. Какие способы автоматической генерации отчетов вы знаете?
28. Что следует указать программе Access для автоматической генерации отчета? (На базе чего выполняется отчет?)
29. С какими еще объектами кроме таблиц, запросов, форм и отчетов работает СУБД Access ? Для чего они предназначены?

ЛАБОРАТОРНАЯ РАБОТА №4

Глобальная сеть Internet и создание Web-документов

1. Цель работы.

Целью работы является изучение возможностей сети Internet, ознакомление с программой просмотра гипертекста Internet Explorer и освоения основных приемов создания Web- документов.

2. Основные теоретические сведения.

Глобальная компьютерная сеть Internet объединяет десятки тысяч серверов, предоставляющих пользователям следующие ресурсы:

- E – MAIL - электронную почту;
- FTP - файловые архивы;
- TELNET - терминальный доступ к удаленному серверу;
- WWW - гипертекстовая информация;
- NEWS - телеконференции;
- TALK, IRC (Internet Relay Chat) - общение в реальном времени;
- MUD (MultiUser Dungeons) – многопользовательские игры.

Сегодня ключевой службой Internet является мультимедийная служба World Wide Web (WWW или просто Web), объединяющая гипертекстовые документы, подготовленные на языке HTML (Hyper Text Markup Language). HTML – документы выглядят наподобие книжных страниц или журнальных полос и могут содержать текстовую, графическую, аудио- и видео информацию. Доступ к Web – страницам основан на протоколе передачи гипертекста (HyperText Transfer Protocol, HTTP). HTTP – страницы размещаются на HTTP – серверах, которые также называются Web – серверами. Механизм гиперсвязей позволяет переходить от одной страницы к другой в поисках информации. При этом страницы могут быть расположены как на одном сервере, так и нескольких. Краткий список наиболее важных терминов, знание которых необходимо для работы в Internet.

WWW – система взаимосвязанных документов, помещенных в Internet. Эти документы называются страницами.

Средство просмотра WWW – любое приложение, при помощи которого можно просматривать страницы Web (Internet Explorer, Netscape Navigator).

Начальная страница. Это самая первая страница, которая появляется при запуске программы запроса.

URL (Universal Resource Locator). Универсальный локатор ресур-

сов – это адрес, который в Internet определяет страницу Web или какой-то другой ресурс. Например, URL для Web-сервера компании Microsoft выглядит так:

<http://www.microsoft.com>.

Связь – это любой выделенный фрагмент страницы Web, при помощи которого можно перейти к другой странице. Для этого по связи нужно щелкнуть мышью. Связь можно отличить по виду указателя мыши, он принимает форму «руки», если его поместить над выделенным фрагментом.

HTML – язык разметки гипертекста – специальный язык описания документов, при помощи которого создаются все документы Web.

HTTP – протокол передачи данных, при помощи которого по Internet пересылаются документы HTML.

FTP – протокол передачи файлов, регламентирующий процесс пересылки файлов с одного компьютера на другой с сети Internet.

2.1. MS Internet Explorer – средство просмотра гипертекста.

Программа MSIE служит для просмотра гипертекстовых документов формата HTML. Для запуска программы достаточно щелкнуть два раза по значку Internet, расположенному на Рабочем столе. А также можно запустить из Главного меню, выбрав команду *Программы – Приложения – Internet – Internet Explorer*.

В окне программы MSIE выделен ряд областей:

- панель инструментов – обеспечивает быстрый доступ к наиболее часто используемым командам. Все команды панели инструментов также могут быть вызваны и через меню.
- В панели адреса – изображается адрес Internet для активной страницы Web. Эта панель используется и для перемещений по сети. Для этого нужно в панели адреса ввести адрес нужной страницы.
- Значок с изображением земного шара служит индикатором занятости программы MSIE – земной шарик вращается когда MSIE ожидает поступления данных с удаленного компьютера.
- В строке состояния изображается информация о состоянии MSIE в данный момент. Для получения подробной информации нужно расположить указатель мыши над значком в правой части строки состояния.
- В области документа изображается страница Web, просматриваемая в данный момент. Выделенные фрагменты страницы – это связи. Щелчок мышью по связи загружает ту страницу, на которую эта связь указывает.

Программа MSIE позволяет перемещаться по системе Web не-

сколькими способами:

- перейти к определенной странице, щелкнув по связи;
- перейти к определенной странице, указав адрес Internet (URL);
- вернуться к ранее просмотренным страницам при помощи команд *Вперед* и *Назад*;
- при помощи команд меню *Файл*;
- вернуться к излюбленным страницам, сохранив указатели на эти страницы.

MSIE позволяет распечатать любую страницу Web командой *Печать* из меню *Файл*. При этом на экране появляется общее диалоговое окно *Печать*.

MSIE позволяет записать любую страницу Web на диск в виде файла. При помощи команды *Сохранить как* из меню *Файл* можно сохранить страницу Web в одном из двух форматов – текстовом или HTML.

MSIE позволяет перекачивать файлы с некоторых серверов Web, позволяющих это сделать. Обычно MSIE отображает такие файлы в виде в виде связей. После щелчка на соответствующей связи появляется окно *Сохранение документа*, где нужно указать место записи файла. Перекачивание файла можно прервать щелчком на кнопке *Останов*.

MSIE способна работать с ярлыками страниц Web. Эти ярлыки могут указывать на информацию, которая находится в любом месте сети. Создав ярлык его можно поместить на Рабочий стол, отправить друзьям в сообщении электронной почты или внедрить в документ какого-либо приложения. Для создания ярлыка текущей страницы Web нужно вызвать команду *Создать ярлык* из меню *Файл*. После создания ярлыка его можно переместить, скопировать или переименовать. Двойной щелчок по ярлыку вызовет переход к той странице, на которую он указывает.

MSIE позволяет перетаскивать графические изображения и фрагменты текста из области документа и помещать их на *Рабочий стол* или в документы других приложений:

- Убедитесь, что MSIE не развернуто во весь экран, и что на экране видна часть *Рабочего стола*.
- Расположите указатель мыши над рисунком и нажмите левую кнопку мыши.
- Переместите указатель мыши на поверхность *Рабочего стола*. При этом вид указателя должен измениться.
- Отпустите кнопку мыши. На *Рабочем столе* должен появиться значок, соответствующий скопированному изображению.

MSIE может превратить рисунок страницы Web в обои Windows. Нужно щелкнуть правой кнопкой мыши на этом рисунке. Появится меню объекта, из которого нужно выбрать команду *Установить обои Windows*.

Многие страницы Internet содержат огромные объемы текста, который просто невозможно прочитать целиком. При помощи команды *Найти* меню *Правка* можно найти нужную фразу или фрагмент текста в пределах страницы.

Некоторые составные части Internet строятся по иной, чем Web технологии: FTP и GOPHER. MSIE обеспечивает для работы с этими системами точно такой же интерфейс как и при работе с Web- серверами. Для подключения к FTP – серверу его адрес вводится точно также как и адреса страниц Web – т.е. в панели адреса или с помощью команды *Открыть* в меню *Файл*. Например, <ftp://ftp.microsoft.com>.

Если для подключения к FTP-серверу нужно указать пользовательские имя и пароль, то используется следующий формат запроса: <ftp://имя.пароль@ftp.microsoft.com>.

2.2. Создание Web-страницы.

Все страницы в Web созданы с помощью языка HTML. HTML – документы представляют собой ASCII – файлы, доступные для просмотра и редактирования в любом редакторе текстов. В отличие от обычного текстового файла, в HTML-документах присутствуют специальные команды – теги, которые указывают правила форматирования документа. Эти команды начинаются с имени тега в угловых скобках, а заканчиваются именем тега с косой чертой также в угловых скобках.

Некоторые основные теги:

<HTML/ - начало документа,

</HTML/ - конец документа,

<TITLE/ - титульный заголовок страницы,

<BODY/ - часть документа, отображаемая на экране,

<H1/,<H2/,<H3/,<H4/ - размер шрифта,

<P/ - абзац,

<BR/ - начало строки,

<HR/ - горизонтальная разделительная линия.

Пример простейшего HTML – документа:

<HTML/

<HEAD/

<TITLE/ Простой HTML- документ </TITLE/

```
</HEAD/
<BODY/
<H1/ HTML - это совсем просто </H1/
Добро пожаловать в WWW <BR/ и мир HTML – документов
</BODY/
</HTML/
```

Символы верхнего и нижнего регистров в тегах не различаются.

Основная мощь языка HTML заключается в возможности связи отдельных частей текста и иллюстраций с другими документами. Гипертекстовые ссылки выделяются в тексте документа специальным цветом и, активизированные мышью, дают возможность перемещаться по документам, или по частям одного документа.

Для указания на ссылку в языке HTML используется тег <A/. Чтобы включить ссылку в документ, необходимо:

- поместить в текст тег <A/ после которого следует пробел;
- указать имя файла, содержащего документ, на который происходит ссылка в форме: HREF=имя файла и закрывающую скобку /.
- указать на завершение тега </A/

Пример простой ссылки:

```
<A HREF="CHAPTER2.HTM"/Глава 2</A/
```

Для включения в состав документа графических изображений используется тег

```
<IMG SRC="имя файла"/
```

Для создания HTML – документов в состав MSIE входит специальный редактор Microsoft Front Page Express. Кроме этого, Web – страницы можно создать с помощью приложений Office 97.

Например, текстовый редактор Word 97 предоставляет два способа создания Web – страниц: с помощью шаблона, либо преобразование в Web – страницу существующего Web – документа. Способ, которым каждое приложение Office 97 обрабатывает HTML, заключается в использовании специального модуля, который перекодирует стандарт данного приложения в формат HTML. Необходимо убедиться, что эти модули установлены: в меню *СЕРВИС* нужно выбрать пункт *Шаблоны и надстройки*. В списке надстроек нужно проверить наличие элементов HTML.DOT и HTML.WLL.

1. Находясь в Word, в меню *Файл* выбрать пункт *Создать*.

2. В появившемся окне выбрать вкладку Web – страницы и сделать двойной щелчок по шаблону *МАСТЕР WEB – страниц*. Появится диалоговое окно этого мастера.
3. Выбрать пункт *Простая* и щелкнуть на кнопке *Далее*. Появится список стилей, из которых выбрать *Элегантный стиль* и нажать кнопку *Готово*.
4. Появится окно документа с открытым шаблоном.

3. Порядок выполнения работы.

Задание 1. Создание простейшей Web-страницы

1. Запустите текстовый редактор *Блокнот*.
2. Введите следующий документ:

```
<HTML/
<HEAD/
<TITLE/Заголовок документа</TITLE/
</HEAD/
<BODY/
Содержание
документа
</BODY/
</HTML/
```

3. Сохраните этот документ под именем first.htm.

Перед сохранением убедитесь, что сброшен флажок *Не показывать расширения для зарегистрированных типов файлов* (Пуск / Настройка / Свойства папки / Вид). В противном случае редактор *Блокнот* может автоматически добавить в конец имени расширение .TXT.

4. Запустите программу Internet Explorer (Пуск / Программы / Internet Explorer),

5. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл first.htm.

6. Посмотрите, как отображается этот файл – простейший корректный документ HTML. Где отображается содержимое элемента TITLE? Где отображается содержимое элемента BODY?

7. Как отображаются слова «Содержание» и «документа», введенные в двух отдельных строчках? Почему? Проверьте, что происходит при уменьшении ширины окна.

Задание 2. Изучение приемов форматирования абзацев

1. Откройте документ first.htm в программе Блокнот.
2. Удалите весь текст, находящийся между тегами <BODY/ и </BODY/. Текст, который будет вводиться в последующих пунктах это-

го упражнения, необходимо поместить после тега <BODY/, а его конкретное содержание может быть любым.

3. Введите заголовок первого уровня, заключив его между тегами <H1/ и </H1/.

4. Введите заголовок второго уровня, заключив его между тегами <H2/ и </H2/.

5. Введите отдельный абзац текста, начав его с тега <P/. Пробелы и символы перевода строки можно использовать внутри абзаца произвольно.

6. Введите тег горизонтальной линейки <HR/.

7. Введите еще один абзац текста, начав его с тега <P/.

8. Сохраните этот документ под именем paragraph.htm.

9. Запустите обозреватель Internet Explorer (*Пуск / Программы / Internet Explorer*).

10. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл paragraph, htm.

11. Посмотрите, как отображается этот файл. Установите соответствие между элементами кода HTML и фрагментами документа, отображаемыми на экране.

Задание 3. Создание гиперссылок

1. Откройте документ first.htm в программе *Блокнот*.

2. Удалите весь текст, находящийся между тегами <BODY/ и </BODY/. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY/.

3. Введите фразу: Текст до ссылки.

4. Введите тег: <A HREF="first.htm"/.

5. Введите фразу: Ссылка.

6. Введите закрывающий тег </A/.

7. Введите фразу: Текст после ссылки.

8. Сохраните документ под именем link.htm.

9. Запустите обозреватель Internet Explorer (*Пуск / Программы / Internet Explorer*).

10. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл link.htm.

11. Убедитесь в том, что текст между тегами <A/ и </A/ выделен как ссылка (цветом и подчеркиванием).

12. Щелкните на ссылке и убедитесь, что при этом загружается документ, на который указывает ссылка.

13. Щелкните на кнопке *Назад* на панели инструментов, чтобы

вернуться к предыдущей странице. Убедитесь, что ссылка теперь считается «просмотренной» и отображается другим цветом.

Задание 4. Создание изображения и использование его на Web-странице

1. Откройте программу Paint (*Пуск / Программы / Стандартные / Paint*). Задайте размеры нового рисунка, например 50x50 точек (*Рисунок / Атрибуты*).

2. Выберите красный цвет переднего плана и зеленый цвет фона. Залейте рисунок фоновым цветом.

3. Инструментом *Кисть* нанесите произвольный красный рисунок на зеленый фон.

4. Сохраните рисунок под именем pic1.gif (в формате GIF).

5. Дайте команду *Рисунок / Атрибуты*. Установите флажок *Использовать прозрачный цвет фона*. Щелкните на кнопке *Выбор цвета* и выберите зеленый цвет, уже использованный на рисунке в качестве фонового.

6. Сохраните рисунок еще раз под именем pic2.gif (в формате GIF) и закройте программу Paint.

7. Если это задание выполняется не сразу после предыдущего, откройте документ first.htm в программе *Блокнот*.

8. Удалите весь текст, находящийся между тегами <BODY/ и </BODY/. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY/.

9. Введите произвольный текст (протяженностью 4-5 строк) и установите текстовый курсор в его начало.

10. Введите тег <IMG SRC="pic1.gif" ALIGN="BOTTOM"/.

11. Сохраните документ под именем picture.htm.

12. Запустите обозреватель Internet Explorer.

13. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл picture.htm. Посмотрите на получившийся документ, обращая особое внимание на изображение.

14. Вернитесь в программу *Блокнот* и измените значение атрибута: ALIGN="TOP". Сохраните файл под тем же именем.

15. Вернитесь в программу Internet Explorer и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

16. Вернитесь в программу *Блокнот* и измените значение атрибута: ALIGN="LEFT". Сохраните файл под тем же именем.

17. Вернитесь в программу Internet Explorer и щелкните на кнопке

Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

18. Вернитесь в программу *Блокнот* и добавьте в тег <IMG/ атрибуты: HSPACE=40 VSPACE=20. Сохраните файл под тем же именем.

19. Вернитесь в программу Internet Explorer и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

20. Вернитесь в программу *Блокнот* и измените имя рисунка: SRC="pic2.gif". Сохраните файл под тем же именем.

21. Вернитесь в программу Internet Explorer и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов. В чем различие между двумя созданными рисунками?

Задание 5. Приемы форматирования текста

1. Если это задание выполняется не сразу после предыдущего, откройте документ first.htm в программе Блокнот.

2. Удалите весь текст, находящийся между тегами <BODY/ и </BODY/. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY/, а его конкретное содержание может быть любым.

3. Введите тег <BASEFONT SIZE="5" COLOR="BROWN"/. Он задает вывод текста по умолчанию увеличенным шрифтом и коричневым цветом.

4. Введите произвольный абзац текста, который будет выводиться шрифтом, заданным по умолчанию. Начните этот абзац с тега <P/.

5. Введите теги: <P/ <FONT SIZE="-2" FACE="ARIAL" COLOR="GREEN"/.

6. Введите очередной абзац текста, закончив его тегом </FONT/.

7. В следующем абзаце используйте по своему усмотрению парные теги: <B/ (полужирный шрифт), <I/ (курсив), <U/ (подчеркивание), <S/ (вычеркивание), <SUB/ (нижний индекс), <SUP/ (верхний индекс).

8. В следующем абзаце используйте по своему усмотрению парные теги: <EM/ (выделение), <STRONG/ (сильное выделение), <CODE/ (текст программы), <KBD/ (клавиатурный ввод), <SAMP/ (пример вывода), <VAR/ (компьютерная переменная).

9. Сохраните полученный документ под именем format.htm.

10. Запустите обозреватель Internet Explorer (*Пуск / Программы / Internet Explorer*).

11. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и

откройте файл `format.htm`.

12. Изучите, как использованные элементы HTML влияют на способ отображения текста.

13. Вернитесь в программу *Блокнот* и измените документ так, чтобы элементы, задающие форматирование, были вложены друг в друга. Сохраните документ под тем же именем.

14. Вернитесь в программу Internet Explorer и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы.

Задание 6. Приемы создания списков

1. Если это задание выполняется не сразу после предыдущего, откройте документ `first.htm` в программе *Блокнот*.

2. Удалите весь текст, находящийся между тегами `<BODY/` и `</BODY/`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY/`, а его конкретное содержание может быть любым.

3. Вставьте в документ тег `<OL TYPE="I"/`, который начинает упорядоченный (нумерованный) список.

4. Вставьте в документ элементы списка, предваряя каждый из них тегом `<LI/`.

5. Завершите список при помощи тега `</OL/`.

6. Сохраните полученный документ под именем `list.htm`.

7. Запустите обозреватель Internet Explorer (*Пуск / Программы / Internet Explorer*).

8. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл `list.htm`.

9. Изучите, как упорядоченный список отображается в программе Internet Explorer, обращая особое внимание на способ нумерации, заданный при помощи атрибута `TYPE=`.

10. Вернитесь в программу *Блокнот* и установите текстовый курсор после окончания введенного списка.

11. Вставьте в документ тег `<UL TYPE="SQUARE"/`, который начинает неупорядоченный (маркированный) список.

12. Вставьте в документ элементы списка, предваряя каждый из них тегом `<LI/`.

13. Завершите список при помощи тега `</UL/`. Сохраните документ под тем же именем.

14. Вернитесь в программу Internet Explorer и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид

страницы, обратив внимание на способ маркировки, заданный при помощи атрибута TYPE=.

15. Вернитесь в программу Блокнот и установите текстовый курсор после окончания введенного списка.

16. Вставьте в документ тег <DL/, который начинает список определений.

17. Вставьте в список определяемые слова, предваряя соответствующие абзацы тегом <DT/.

18. Вставьте в список соответствующие определения, предваряя их тегом <DD/.

19. Завершите список при помощи тега </DL/. Сохраните документ под тем же именем.

20. Вернитесь в программу Internet Explorer и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как выглядит при отображении Web-страницы список определений.

Задание 7. Создание таблиц

1. Если это задание выполняется не сразу после предыдущего, откройте документ first.htm в программе Блокнот.

2. Удалите весь текст, находящийся между тегами <BODY/ и </BODY/. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY/. В данном упражнении используется список номеров телефонов.

3. Введите тег <TABLE BORDER="10" WIDTH="100%"/.

4. Введите строку: <CAPTION ALIGN="TOP"/Список телефонов</CAPTION/.

5. Первая строка таблицы должна содержать заголовки столбцов. Определите ее следующим образом:

```
<TR BGCOLOR="YELLOW"  
ALIGN="CENTER"/  
<TH/Фамилия<TH/Номер телефона
```

6. Определите последующие строки таблицы, предваряя каждую из их тегом <TR/ и помещая содержимое каждой ячейки после тега <TD/.

7. Последнюю строку таблицы задайте следующим образом:

```
<TR/<TD ALIGN="CENTER" COLSPAN="2"/На первом этаже здания  
имеется бесплатный телефон-автомат.
```

8. Завершите таблицу тегом </TABLE/.

9. Сохраните документ под именем table.htm.

10. Запустите обозреватель Internet Explorer (*Пуск / Программы /*

Internet Explorer).

11. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл `table.htm`.

12. Изучите, как созданная таблица отображается в программе *Internet Explorer*, обращая особое внимание на влияние заданных атрибутов.

13. Измените ширину окна обозревателя и установите, как при этом изменяется внешний вид таблицы.

Задание 8. Создание описания фреймов

1. Запустите текстовый редактор *Блокнот*.

2. Введите следующий документ:

```
<HTML/  
<HEAD/  
<TITLE/Описание фреймов</TITLE/  
</HEAD/  
<FRAMESET  
ROWS="60%,*"/  
<FRAME SRC="table.htm"/  
<FRAMESET COLS="35%,65%"  
NORESIZE/  
<FRAME SRC="flrst.htm"/  
<FRAME SRC="links.htm"/  
</FRAMESET/  
</HTML/
```

3. Сохраните этот документ под именем `frames.htm`.

4. Запустите обозреватель *Internet Explorer*.

5. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл `frames.htm`.

6. Изучите представление нескольких созданных ранее документов, в отдельных фреймах.

7. Посмотрите, что происходит при изменении ширины окна обозревателя.

8. Проверьте, можно ли изменить положение границ фреймов методом перетаскивания при помощи мыши.

9. Щелкните на ссылке, имеющейся в одном из фреймов, и посмотрите, как будет отображен новый документ.

10. Щелкните на кнопке *Назад* на панели инструментов и убедитесь, что возврат к предыдущему документу не нарушает структуру фреймов.

11. Вернитесь в программу *Блокнот* и измените структуру и параметры фреймов по своему усмотрению. Сохраните документ под тем же именем.

12. Вернитесь в программу Internet Explorer и щелкните на кнопке *Обновить* на панели инструментов. Убедитесь, что измененный вид Web-страницы соответствует замыслу. Если это не так, вернитесь в программу *Блокнот*, найдите и исправьте ошибки.

Задание 9. Создание Web-документа с помощью редактора Frontpage Express

1. Запустите программу Frontpage Express (*Пуск / Программы / Стандартные / Средства Интернета / Frontpage Express*).

2. Введите в программе Frontpage Express произвольный текст документа.

3. С помощью панели инструментов форматирования отформатируйте текст по собственному усмотрению.

4. Для создания таблицы щелкните на кнопке *Вставить таблицу* на стандартной панели инструментов.

5. Для добавления иллюстраций используйте кнопку *Вставить изображение* на стандартной панели инструментов. Иллюстрации возьмите из папки C:/Windows.

6. Дайте команду *Файл / Сохранить*, щелкните на кнопке *Как файл* и задайте имя файла wysiwyg.htm. Подтвердите сохранение изображений, требующих преобразования формата.

7. Запустите обозреватель Internet Explorer (*Пуск / Программы / Internet Explorer*).

8. Дайте команду *Файл / Открыть*. Щелкните на кнопке *Обзор* и откройте файл wysiwyg.htm.

9. Убедитесь, что созданный документ правильно отображается обозревателем. Обратите внимание на наличие отличий вида документа при отображении в обозревателе и в программе Frontpage Express.

10. Измените ширину окна обозревателя и посмотрите, как при этом меняется вид документа.

11. Вернитесь в программу Frontpage Express и дайте команду *Вид / HTML*

12. Изучите автоматически сгенерированный код HTML, определите, как с помощью тегов HTML реализованы использованные команды форматирования.

4. Контрольные вопросы.

1. Какие ресурсы доступны в Internet?
2. Как найти нужную информацию в Internet?
3. Как скопировать файл по ftp?
4. Какие возможности предоставляет программа Internet Explorer?

Список литературы

1. Автоматизированные видеоинформационные системы. Андреев А.Л. СПбНИУ ИТМО, 2011, <http://e.lanbook.com/books/>
2. Автоматизированные библиотечно-информационные системы. Меркулова А.Ш. 2011. <http://e.lanbook.com/books/>
3. Пакеты прикладных программ : учеб. /- 3-е изд. Э.В. Фуфаев, Л.И. Фуфаева М : Академия, 2008.
4. Компьютерное моделирование в системе Mathcad : учеб.пособие для вузов /. В.А. Охорзин. М. : Финансы и статистика, 2006.
5. Базы данных : учеб.пособие для вузов /. – 3-е изд., стереотип. А.В. Кузин, С.В. Левонисова. М. :Академия, 2008.
6. Основы сетей передачи данных: курс лекций : учеб.пособие /. - Изд. 2-е, испр. В.Г. Олифер, Н.А.Олифер. М.: нтернет-Университет Информационных технологий.
7. Сети и телекоммуникации. Учеб. пособие для вузов, С.А. Пескова, А.В. Кузин, А.Н. Волков. М. Академия, 2008.
8. Компьютерные технологии при проектировании и эксплуатации технологического оборудования : учеб. пособие / - 2-е изд., испр. и доп. Г.В. Алексеев и др. СПб. : ГИОРД, - 296 с. : ил.
9. Информационные технологии в экономике и управлении. Учебник. Под ред. Трофимова, 2014.
10. Информатика и информационные технологии. Гаврилов А.П. Изд-во «Юрайт».
- 11.Базы данных. Фуфаев Э.В., Фуфаев Д.Э. 2008. <http://www.arhibook.ru/>.
- 12.Компьютерные сети. Том 1. Системы передачи данных. Смелянский, 2011Р.Л. <http://www.arhibook.ru/>
- 13.Excel 2010 на примерах. Петраков А. М., Клейменов С. А., Мельников В. П., 2008. <http://www.arhibook.ru/>
- 14.Информационные технологии: учебник. Под ред. В. В. Трофимова, 2011. <http://www.arhibook.ru/>
- 15.Современные информационные технологии. Максимов Н. В., Партыка Т. Л, Попов И. И. 2008. <http://www.arhibook.ru/>
- 16.Информационные системы и технологии управления. Титоренко Г.А.(ред.), 2011. <http://www.arhibook.ru/>.