Министерство образования и науки Украины

Севастопольский национальный технический университет

Кафедра технической кибернетики

ОТЧЕТ

по контрольной работе

# создания интерактивных WEB - страниц для получения информации в сети Интернет с использованием форм, заполняемых пользователем.

Выполнил: студент гр. А-41з

Брусинов С. Э.

Проверил: Альчаков В.В.

Севастополь - 2007

# 1 Цель работы

# Изучить возможности создания интерактивных WEB - страниц для получения информации в сети Интернет с использованием форм, заполняемых пользователем.

**2 Основные положения**

# 2.1 Кратко о WWW и о гипертексте

# INTERNET определяется как глобальная информационная сеть, имеющая доступ практически ко всей совокупности всемирных компьютерных сетей. В INTERNET находят применение различные службы (сервисы), такие как электронная почта, телеконференции, системы поиска и доставки файлов, Всемирная паутина - Word Wide Web, в основе которой лежит идея формирования документа в форме гипертекстовой структуры.

# Всемирная паутина - WWW - это средство поддержки системы гипертекста на INTERNET.

# Гипертекст представляет собой форму организации документа, текст которого может содержать ссылки к отдельным словам, фразам, рисункам и другим элементам текста, что позволяет получить дополнительную информацию об этих элементах текста. С помощью ссылок отдельные страницы текста связываются между собой, образуя гипертекстовый документ. Таким образом, гипертекстовый документ состоит из набора текстовых фрагментов, графики, выделенных элементов документа и системы связи выделенных элементов с другими документами.

**2.2 Тег, контейнер, атрибут**

В основе гипертекстового языка лежит понятие тег (tag - метка, ярлык, этикетка, бирка) определяющим инструкцию по форматированию следующего за тегом текстового (или иного) фрагмента документа.

Каждый тег начинается со знака «меньше (<)» заканчивается знаком «больше (>)». Между этими двумя символами находится ключевое слово, указывающее основную функцию тега. Например, ключевое слово IMG в теге <IMG> означает, что тег вставляет изображение в документ. Разнообразие ключевых слов позволяет создавать огромные количества эффектов форматирования.

Многие теги HTML, особенно те, которые обеспечивают специальное форматирование, являются парными. Первый тег в паре активизирует эффект, а второй тег – завершает его действие. Например, тег <I> включает курсив. В конце текста, выделяемого курсивом, ставится тег </I>. Косая черта ( / ) перед ключевым словом означает прерывание эффекта, управляемого этим словом. Первый тег в паре называется открывающим, второй – закрывающим. Парные теги называются контейнерными, потому что они образуют как бы контейнер для форматируемого текста. Например, в HTML-коде, HTML – теги <I> очень важны </I>, теги <I> и </ I> являются контейнерными для текста “очень важны”.

Другие HTML-теги стоят сами по себе и называются одиночными. Они, как правило, являются «одномоментными» инструкциями форматирования, например, вставкой изображения, горизонтальной линии, перевода строки и т.д.

Помимо ключевого слова, имеющегося у каждого тега, некоторые теги содержат атрибуты, модифицирующие его действие. Подбирая значение атрибутов, можно экспериментировать в поисках оптимальной схемы размещения элементов на странице. Например, тег <HR>, который изображает горизонтальную линию, имеет несколько атрибутов, влияющих на внешний вид линии. Так, используя атрибут WIDTH, можно заставить линию дойти до середины строки (width = 50%), а атрибутом SIZE изменять толщину линии (size = 4, значение по умолчанию =1).

**2.3 Структура любого HTML- документа**

Теги <html> и </html> открывают и соответственно закрывают HTML документ. Теги <head> и </head> отделяют заголовок, где, как правило, записывается служебная информация не читаемая в окне программы просмотра, например, в нашем случае в заголовке документа командами <title> и </title> в верхнем левом углу браузера выводятся слова "пример" . Тело всего документа выделяется тегами <body> и </body>. Между этими тегами, как правило, записывается все то, что будет отображаться при открытии документа. Слово "привет!" отображается при просмотре большими буквами потому, что заключено между тегами <h1> и </h1> эти теги применяются для оформления заголовков в тексте выводимом на экран браузером. В эти тегах Вы можете применять числа от 1 до 6. Основной текст нашего документа "пример HTML - документа" выделен тегами <p> и </p> это теги абзаца. Обычно между этими тегами в программе пишут текст, который затем будет выводиться на экран.

**2.4 Виды списков, которые возможно реализовать в HTML- документе**

HTML поддерживает три различных типов списков, каждый из которых позволяет красиво организовать большое количество информационных элементов. Для задания списка необходима пара тегов на весь список и по паре на каждый элемент списка Если речь идет об упорядоченных списках, то имеются в виду теги <OL> и < /OL>, если же речь идет о неупорядоченных списках (маркированных), то подразумеваются теги <UL> & < /UL>.

Браузер автоматически нумерует элементы упорядоченного списка, начиная с 1.

Третий тип списков – списки описаний (глоссарии) позволяют перечислять термины, за которыми следуют их определения. Термины заключаются в теги <DT> & < /DT>, а описания (определения) в теги <DD> & < /DD>.

Для определения каждого элемента списка используется открывающийся и закрывающийся теги <LI> & < /LI>.

Например:

<OL>

<LI> Numered list items < /LI>

<LI> List items are idented < /LI>

< /OL>

Тег <OL> имеет два атрибута: Type и Start. Атрибут TYPE устанавливатеся равным 1, А, а, i или I, в зависимости от типа нумерации списка – цифрами, большими или малыми буквами, большими или малыми римскими цифрами.

Атрибут START используется, если нумерация должна начинаться не с единицы.

Например:

< OL TYPE = “A” START = 3 > - значит, нумерация списка будет начинаться с буквы С.

Тег <UL> поддерживает атрибут Type, позволяющий вместо маркировочного символа «кружок» (О), (Type = “dise”), устанавливаемого по умолчанию, задавать маркированный символ «квадрат» (Type = “square”) или «окружность» (Type = “circle”).

Рациональное использование тегов <UL> & <OL> позволяет создавать вложенные друг в друга маркированные и нумерованные списки, а также использование атрибута TYPE тега <OL> позволяет создавать вложенные нумерованные списки с разным типом нумерации (буквы и цифры).

**2.5 Использование именованных якорей гиперссылок**

Когда пользователь выбирает гиперссылку, браузер загружает документ, на который указывает гипертекст и выводит на экран его начало. В том случае, если документ очень большой пользователю приходится долго его прокручивать, чтобы перейти к нужному месту. Чтобы избежать этого, HTML предоставляет возможность установить именованные якоря (закладки в документе), а гиперссылки будут непосредственно указывать на эти якоря (т.е. будет осуществляться переход по закладке внутри документа).

Для создания точек привязки (установления закладок) в пределах одного документа используется следующий синтаксис:

< A name = «уникальное\_имя\_закладки»>[тело ссылки]< /A >,

где «уникальное\_имя\_закладки» - это уникальное имя точки привязки, а «тело\_ссылки» - это литерал, отображаемый на экране.

Ниже приведен пример программы, поясняющей механизм расстановки закладок внутри документа.

<HTML>

<HEAD>

<TITLE> Anchors < /TITLE >

< /HEAD>

<BODY>

< A name = “Topic\_1”> Topic.1 </A>

Topic 1 stuff

< A name = “Topic\_2”> Topic.2 </A>

Topic 2 stuff

< A name = “Topic\_3”> Topic.3 </A>

Topic 3 stuff

< /BODY>

< /HTML >

В качестве места, для которого создается закладка, могут выступать заголовки разделов внутри большого по размерам документа. Переход по этим закладкам обеспечивает быстрое перемещение по документу к нужному разделу.

Теги, использующиеся для связывания с точками привязки в пределах одного HTML- документа, идентичны тегам создания гиперссылок на внешние документы. Для создания гиперссылки на новый фрагмент используется соединяющий синтаксис:

< A href = [ URL ] # “имя\_закладки” > имя\_якоря < /A >

, где [URL] – необязательный адрес, «имя\_закладки» - название точки привязки в пределах текущего документа, «имя\_якоря» - это текст или какое-либо другое содержимое, щелчек по которому будет активизировать гиперссылки.

Именованные якоря полезны при создании оглавления, ссылки которого указывают на различные части документа. Наиболее распространенный метод связывания с точкой внутренней привязки предусматривает использование символа «#» и названия точки привязки без указания URL. В этом случае переход по необходимому адресу осуществляется без повторной загрузки документа.

Пример определения точек привязки и установления гипертекстовых ссылок на эти точки приведен ниже:

<HTML>

<HEAD>

<TITLE> Anchors < /TITLE >

< /HEAD>

<BODY>

Read About < A href = “#Topic\_1” > Topic\_1 < /A >

Read About < A href = “#Topic\_2” > Topic\_2 < /A >

Read About < A href = “#Topic\_3” > Topic\_3 < /A >

<HR>

< A name = “Topic\_1”> <HR> Topic 1 < /HR> </A>

Topic 1 stuff

< A name = “Topic\_2”> <HR> Topic 2 < /HR> </A>

Topic 2 stuff

< A name = “Topic\_3”> <HR> Topic 3 < /HR> </A>

Topic 3 stuff

< /BODY>

< /HTML >

**2.6 Графические объекты для гиперссылок в HTML- документе**

Чтобы разместить изображение на странице необходимо воспользоваться тегом <IMG>. При работе с этим тегом вставки изображения используется следующий синтаксис:

< IMG SRC = “[URL] имя\_файла”>.

Значение атрибута SRC задает «имя\_файла» с изображением.

Тег <IMG> включает несколько полезных атрибутов, характеристика которых дана ниже:

Атрибут ALT. С помощью этого атрибута указывается строка текста, используемая браузером вместо изображения, которое по каким-либо причинам невозможно загрузить.

Атрибут ALIGN. Определяет выравнивание текста относительно изображения. Если этот атрибут принимает одно из значений TOP, MIDDLE или BOTTOM, то текст, следующий за изображением, выравнивается по верхнему краю, по центру или по нижнему краю изображения. Если атрибут ALIGN установить в значение LEFT или RIGHT, то изображение будет плавать у левого или правого поля соответственно, а текст будет его обтекать.

Атрибут HEIGHT и WIGTH позволяют определять размер изображения в пикселах. В качестве примера, поясняющего действие атрибутов HEIGTH & WIGTH на изображение, может быть приведен следущий синтаксис тега <IMG>:

< IMG SRC = “URL” wigth=пикселы heigth=пикселы >.

В том случае, когда графическое изображение используется как альтернатива гипертексту, синтаксис установки графического якоря такой же, как при установке текстового. Между тегами <A href = “URL”> и < /A> вместо текста должен быть помещен тег <IMG>, в соответсвии со следующим снтаксисом:

< A href = “URL” > < IMG = “файл\_изображения” > < /A>.

Примечание. В том случае, если графический объект интерпретируется как якорь гиперссылки, тогда вызываемый HTML- документ, само изображение, а также вызывающий документ должны находиться в одной папке (на локальном компьютере). В этом случае синтаксис гиперссылки имеет следующий вид:

< A HREF = “bs.htm” > < IMG SRC = “km.jpg”> < /A >

**2.7 Фреймы их использование**

Фреймы позволяют разделить Web-страницы на независимые области произвольного или фиксированного размера. Выполняя определенные действия в одних фреймах, можно изменять содержимое других фреймов – это позволяет рассматривать фреймы в качестве инструмента навигации. Перечень гиперссылок на документы на узле помещается в одном фрейме, а другой фрейм выделяется собственно под содержимое.

Первый шаг в создании фреймовой схемы расположения состоит в разбивке окна на нужные области. Это осуществляется путем реализации отдельного HTML – документа, который целиком используется для определения фрейма. В этом документе вместо тегов <BODY> & < /BODY> используются теги <FRAMESET> и < /FRAMESET>, предназначенные для определения фреймов. Атрибуты тега <FRAMESET> определяют, как экран разбивается на части.

Тег <FRAMESET> может иметь один их двух атрибутов:

ROWS – для разбивки экрана на две и более строки;

COLS – для разбивки экрана на столбцы.

Каждому атрибуту присваивается список значений сообщающий браузеру размер строк и столбцов (в % от общей ширины окна, в пикселях или символом (\*), который означает, что браузер должен использовать все оставшееся место). В качестве примера может быть приведен следующий синтаксис:

<FRAMESET ROWS = “25%, 40%, 35%” >

. . .

< /FRAMESET>

или

<FRAMESET COLS = “55, 120, 20, \* ”>

. . .

< /FRAMESET>

Первый пример определяет ширину фреймов в процентах, второй – в пикселях.

Атрибут <FRAMEBORDER> тега <FRAMESET> позволяет устанавливать обрамление у фреймов. Он может принимать одно из двух значений:

1 - соответствует наличию обрамления

0 - отсутствию обрамления.

В случае, если необходимо конструировать достаточно сложную схему разбиения экрана на фреймы осуществляют вложение тегов <FRAMESET> & < /FRAMESET> друг в друга.

Например, схеме организации фреймов, когда экран разбивается сначала на два фрейма по вертикали, а затем первый фрейм разбивается на два по горизонтали (см. Рис.1а), соответствует следующий HTML –код тега FRAMESET:

<FRAMESET COLS = “55%, 45%”>

<FRAMESET ROWS = “75%; 25%”>

. . . данные, размещаемые в верхней части первого столбца . . .

. . . данные, размещаемые в нижней части первого столбца . . .

< /FRAMESET>

. . . данные, размещаемые во втором столбце . . .

< /FRAMESET>

а) б)

Рис. 1 Схемы разбиения фреймов.

Схеме разбивки окна на фреймы, изображенной на рис.1б, соответствует следующий фрагмент HTML –программы:

<FRAMESET ROWS = “50%, 50%”>

. . . данные, размещаемые в верхней части . . .

<FRAMESET COLS = “50%, 50%”>

. . . данные, размещаемые в первом столбце нижней части . . .

. . . данные, размещаемые во втором столбце нижней части . . .

< /FRAMESET>

< FRAMESET>

Используя вложенные теги <FRAMESET>, можно добиваться построения схемы с фреймами любой степени сложности.

Размещение содержимого в фреймах управляет тег <FRAME>. Этот тег имеет обязательный атрибут SRC, который сообщает браузеру URL –документа, загружаемого в фрейм. Среди шести атрибутов (кроме SRC) тега <FRAME> необходимо выделить два наиболее важных:

Атрибут NAME, с помощью которого осуществляется присвоение фрейму имени. Уникальные имена фреймов важны при нацеливании гиперссылок (атрибут TARGET тега <A>) на конкретные фреймы.

Атрибут SCROLLING = YES/NO/AUTO, который управляет появлением горизонтальной и вертикальной полос прокрутки (в случае, если он равен YES). Если атрибут SCROLLING = AUTO (это значение устанавливается по умолчанию), то браузер сам выводит линейки прокрутки, если они необходимы.

Нацеливание гиперссылок на фреймы позволяет постоянно иметь в одном фрейме их неменяющийся список, а вызываемые документы загружать в другие фреймы. Первоначально, во фреймы, предназначенные для вывода содержимого документов, вызываемых по гиперссылкам, можно загружать любые HTML –документы (не несущие никакой принципиальной информации), а затем содержимое документов, вызываемых по гиперссылкам.

Простейший пример формирования фреймовой схемы и определения ее содержимого (однако являющийся законченным HTML -документом) имеет следующий синтаксис:

<HTML>

<HEAD>

<TITLE> ФРЕЙМЫ < /TITLE>

< /HEAD>

<FRAMESET COLS = “50%, 50%”>

<FRAME SRC = “Logo1.htm” NAME = “NUMBER 1”>

<FRAME SRC = “Logo2.htm” NAME = “NUMBER 2”>

< /FRAMESET>

< /HTML>

**2.8 Размещение и реализация изображения в виде карты**

Как мы уже знаем, в HTML-документ можно включать изображения, которые считываются программой просмотра из файлов в одном из графических форматов. Ранее (раздел 2.5) было сказано, что всё изображение целиком можно сделать якорем гиперссылки. Однако, подчас возникает вполне естественное желание сделать различные части одного изображения гиперссылками на различные документы. Представим себе, например, документ с изображением электронной схемы прибора, более подробные сведения об элементах которого пользователь может немедленно получать, щелкая мышью на соответствующих участках чертежа. HTML предлагает для этого следующий способ: добавив к флагу <IMG ...> специальный атрибут ISMAP и/или USEMAP, вы можете "сказать" программе просмотра, что данное изображение является картой (Maps, Clickable Images, Imagemaps). Реакция на щелчок мыши на изображении-карте будет различной в зависимости от координат курсора мыши в момент щелчка.

Старый стандарт HTML 2.0 определял только атрибут ISMAP, который предусматривает обработку "щелчка на карте" на WWW-сервере при помощи CGI-запросов, аналогичных применяемым для обработки заполняемых форм. При таком способе обработки координатных запросов читатель WWW-документа должен щелкнуть на интересующей его точке и дождаться отклика сервера, прежде чем он получит возможность хотя бы узнать, соответствует данная точка какой-нибудь гиперссылке, или нет.

Альтернативу описанному выше механизму обработки "щелчков на карте" сервером составляет предложенный в версии HTML 3.0 механизм обработки клиентом, то есть, самой программой просмотра. Карты такого рода определяются при помощи атрибута USEMAP. Описание областей изображения, ассоциированных с различными гиперссылками, помещается в HTML-файл, как показано в нижеследующем примере. Представим себе несложную радиосхему, состоящую из трех диодов и двух резисторов. Картинку с радиосхемой мы поместили в файл mapex.gif, а описание функций каждого из элементов схемы - в файлы diode1.html, diode2.html, diode3.html, res1.html и res2.html (содержание этих файлов здесь не приводится). Мы хотим, чтобы читатель, щелкая на изображении того или иного элемента, загружал соответствующий документ с описанием данного элемента

**3 Ход выполнения работы**

**Задание: вар №2;**

предметная область: социологический опрос;

2 3 4

1

схема фрейма:

элементы управления: кнопка (submit), текстовое окно (textarea)

**3.1Создание загрузочного файла**

Сначала был создан загрузочный файл, который является главной страничкой сервера и имеет имя index.html. В этом файле заданы размеры, количество и атрибуты фреймов.

<HTML>

<TITLE>Социологический опрос</TITLE> <FRAMESET rows="30%,\*" >

<FRAMESET cols="33%,33%,\*" >

<FRAME src="frame/2.html" name="2\_page" scrolling="auto" noresize>

<FRAME src="frame/3.html" name="3\_page" scrolling="no" noresize>

<FRAME src="frame/4.html" name="4\_page" scrolling="auto" noresize>

</FRAMESET>

<FRAME src="frame/1.html" name="1\_page" scrolling="auto" >

**3.2 Создание файлов для каждого фрейма**

Затем уже будем создавать файлы для каждого из фреймов. Исходя и варианта задания, у нас их четыре. (1.html, 2.html,3.html,4.html) и два вспомогательных файла (2.html, 4.html) на которые ссылаются соответственно файлы 2.html и 4.html.

Приведем их содержание ниже:

Файл 1.html. В нём записан текст формы

<html>

<head>

<title>Социологический опрос</title>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">

</head>

<body bgcolor="#cffed8">

<FORM method="post" enctype="text/plain" id="soc\_que" action="mailto:brusinov-s-e@rambler.ru">

<TABLE id="tabl" border="0" bordercolor="#FFFFFF" width="100%" cellspacing="0" cellpadding="5">

<TR>

<TD align="center" colspan="3"> <h2>Заполните, пожалуйста, анкету:</h2></TD>

</TR>

<TR>

<TD width="40%" >Фамилия:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="40" maxlength="100" name="fam" ></TD>

</TR>

<TR>

<TD width="40%" >Имя:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="20" maxlength="100" name="nam" ></TD>

</TR>

<TR>

<TD width="40%" >Отчество:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="40" maxlength="100" name="lname" ></TD>

</TR>

<TR>

<TD width="40%">Пол:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%">

<INPUT type="radio" name="m\_f" value="male" checked onClick="document.bgColor='#3399FF'">Мужской

<INPUT type="radio" name="m\_f" value="female" onClick="document.bgColor='#FFCCFF'">Женский

</TD>

</TR>

<TR>

<TD width="40%">Возраст:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%">

<SELECT name="age">

<OPTION selected value="10">Менее 20

<OPTION selected value="20">20-30

<OPTION selected value="30">30-50

<OPTION selected value="50">Более 50

</SELECT>

</TD>

</TR>

<TR>

<TD colspan="3" bgcolor="#FFFFFF" height="30"></TD>

</TR>

<TR>

<TD width="40%" >Страна:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="20" maxlength="100" name="country" ></TD>

</TR>

<TR>

<TD width="40%" >Город:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="20" maxlength="100" name="town" ></TD>

</TR>

<TR>

<TD width="40%" >Дом:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="20" maxlength="100" name="house" ></TD>

</TR>

<TR>

<TD colspan="3" bgcolor="#FFFFFF" height="30"></TD>

</TR>

<TR>

<TD width="40%" >Место работы:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="40" maxlength="100" name="work" ></TD>

</TR>

<TR>

<TD width="40%" >Ваше социальное положение(должность):</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%"><INPUT type="text" size="40" maxlength="100" name="post" ></TD>

</TR>

<TR>

<TD width="40%">Ваша заработная плата:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%">

<SELECT name="wages">

<OPTION selected value="100">Менее 200

<OPTION selected value="200">200-500

<OPTION selected value="500">500-1500

<OPTION selected value="1500">Более 1500

</SELECT>

</TD>

</TR>

<TR>

<TD colspan="3" bgcolor="#FFFFFF" height="30"></TD>

</TR>

<TR>

<TD width="40%" >Ваши увлечения:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%">

<INPUT name="hobby1" type="checkbox" value="Спорт">Спорт

<INPUT name="hobby2" type="checkbox" value="Туризм">Туризм

<INPUT name="hobby3" type="checkbox" value="Искуство">Искуство

<INPUT name="hobby4" type="checkbox" value="Музыка">Музыка

<INPUT name="hobby5" type="checkbox" value="Техника">Техника

</TD>

</TR>

<TR>

<TD colspan="3" bgcolor="#FFFFFF" height="30"></TD>

</TR>

<TR>

<TD width="40%">Ваши комментарии:</TD>

<TD width="5%" bgcolor="#FFFFFF" ></TD>

<TD width="50%">

<textarea name="IsItBad" cols="67"> Запишите, устраивает ли вас ваше социальное положение,если нет, то почему?

</textarea>

</TD>

</TR>

<TR bgcolor="#FFFFFF">

<TD width="40%" > <INPUT type="reset" value="Очистить анкету" title="нажмите сюда чтобы очистить поля формы" onClick="document.bgColor='#3399FF'"></TD>

<TD width="5%" ></TD>

<TD width="50%" align="right"> <INPUT type="submit" value="Отправить анкету" title="нажмите сюда чтобы отправить данные" ></TD>

</TR>

</TABLE>

</FORM>

</body>

</html>

Файл 2.html:

<html>  
<head>  
</head>  
<BODY BGCOLOR="#CFFED8">  
   
 <A HREF="2.1.html" TARGET="2\_page">"Чем обусловлена необходимость в опросе"</A>  
</BODY>  
</html>

Файл 2.1.html.

</head>  
<body>  
<BODY BGCOLOR="#CFFED8">  
<FONT COLOR="#FF0000">  
<kbd>  
 <p>Каждый из нас в той или иной степени   
соприкасается …………(текст пропущен)…стране и в   
других странах) . </p>  
</kbd>  
</FONT>   
</body>  
</html>

Файл 3.html:

<html>  
<head>  
 <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">  
</head>  
<BODY BGCOLOR="#CFFED8">  
<kbd><FONT COLOR="red"><MARQUEE style="font-size: 24pt; color: #00000">Помогите провести социологический опрос. Просто заполните анкету, приведенную ниже.Опрос поможет дать оценку социальной обстановке в стране. Предоставленные Вами данные абсолютно конфиденциальны и разглашению не подлежат. Благодарим за сотрудничество.</MARQUEE>  
 </FONT></kbd>   
</BODY>  
</html>

Файл 4.html: <html>  
<html>  
<head>  
 <title>Содержание</title>  
 <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">  
</head>  
<BODY BGCOLOR="#CFFED8">  
   
 <A HREF="4.1.html" TARGET="4\_page">"Каким образом происходит опрос"</A>  
  
</BODY>  
</html>

Файл 4.1.html:

<html>  
<head>  
 <title>Как происходит опрос</title>  
</head>  
<body>  
<BODY BGCOLOR="#CFFED8">  
<FONT COLOR="#FF0000">  
<kbd>  
Всё очень……(текст пропущен)……… картотеку.  
</p>  
</kbd>  
</FONT>   
</body>  
</

**4 Выводы**

Входе выполнения контрольной работы были изучены возможности интерактивных WEB страниц, была разработана активная форма для передачи данных в сеть Internet в соответствии с вариантом задания.