

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.В.ДВ.05.02 WEB-программирование

**Направление подготовки (специальность)** 09.03.01 Информатика и вычислительная техника

**Профиль образовательной программы** “Автоматизированные системы обработки информации и управления”

**Форма обучения** очная

## СОДЕРЖАНИЕ

<b>1. Конспект лекций .....</b>	
1.1 Лекция № 1 Обзор Web- технологий.....	
1.2 Лекция № 2 Инструменты создания web - сайтов и приложений.....	
1.3 Лекция № 3 Проектирование Web - сайта.....	
1.4 Лекция № 4 Язык разметки гипертекста – HTML.....	
1.5 Лекция № 5 Структура HTML документа.....	
1.6 Лекция № 6 Форматирование текста. Интерактивные формы HTML.....	
1.7 Лекция № 7 Введение в JavaScript. Синтаксис языка. Объектная модель JavaScript..	
1.8 Лекция № 8 Типы данных. Операторы JavaScript.....	
1.9 Лекция № 9 Обработка событий. Примеры эффективного программирования на JavaScript.....	
<b>2. Методические материалы по выполнению лабораторных работ .....</b>	
2.1 Лабораторная работа № 1 ЛР-1-2 Обзор Web- технологий.....	
2.2 Лабораторная работа № 2 ЛР-3-4 Инструменты создания web - сайтов и приложений.....	
2.3 Лабораторная работа № 3 ЛР-5-6 Проектирование Web - сайта.....	
2.4 Лабораторная работа № 4 ЛР-7-8 Язык разметки гипертекста – HTML.....	
2.5 Лабораторная работа № 5 ЛР-9-10 Структура HTML документа.....	
2.6 Лабораторная работа № 6 ЛР-11-12 Форматирование текста. Интерактивные формы HTML.....	
2.7 Лабораторная работа № 7 ЛР-13-14 Введение в JavaScript. Синтаксис языка. Объектная модель JavaScript.....	
2.8 Лабораторная работа № 8 ЛР-15-16 Типы данных. Операторы JavaScript.....	
2.9 Лабораторная работа № 9 ЛР-17-18 Обработка событий. Примеры эффективного программирования на JavaScript.....	

## **1. КОНСПЕКТ ЛЕКЦИЙ**

### **1.1. Лекция № 1 (2 часа)**

**Тема: «Обзор Web- технологий»**

#### **1.1.1. Вопросы лекции:**

1. Основные понятия и определения.
2. Сдерживающие факторы развития web-технологий.
3. Клиентские Web-технологии.

#### **1.1.2 Краткое содержание вопросов:**

1. Web-программирование - это отдельное направление в программировании, используемое для создания web-приложений. В связи с ускоренными темпами развития Интернет технологий постоянно появляются новые направления в этой области. Поэтому Web-программирование не сформировавшаяся наука, а скорее набор существующих программных технологий (клиентских и серверных), используемых для организации работы пользователя в сети Интернет. Сервер – компьютер, на котором находятся документы и приложения. Клиент – рабочая станция, используемая для просмотра документов и приложений, хранящихся на сервере.

2. Сдерживающие факторы развития web-технологий:

- ограничения, связанные с сетью Интернет (скорость);
- отсутствие централизованного управления Интернет (конкуренция браузеров);
- ограничения, связанные с технологиями программирования.

3 Интерпретатор команд обрабатывает код приложения и выдает браузеру сформированный документ. Преимуществом такой архитектуры являются возможность работы с динамическими документами (появляется интерактивность), а также возможность однажды загрузив приложение работать в нем, не обращаясь к серверу.

### **1.2. Лекция № 2 (2 часа)**

**Тема: «Инструменты создания web - сайтов и приложений»**

#### **1.2.1. Вопросы лекции:**

1. Инструменты создания web - сайтов.

#### **1.2.2 Краткое содержание вопросов:**

Существует множество инструментов для дизайнеров и разработчиков, но не так-то просто найти нечто стоящее. Ознакомьтесь с каждым пунктом и постарайтесь определить, какой инструмент вам больше всего подойдет.

01. Initializr

Initializr – это генератор шаблонов HTML5, с помощью которого вы без труда сможете начать собственный HTML5-проект. Инструмент основан на мощном шаблоне - HTML5 Boilerplate. Вам просто нужно начать, и вы получите полностью редактируемый шаблон, основанный на Boilerplate.

#### 02. Modernizr

Modernizr добавляет классы к элементам «html», которые позволяют вам указать определенные функции браузера в таблицах стилей. Вам даже не придется писать какой-то javascript-код.

#### 03. HTML5 test

HTML5 test – это лишь направляющий документ о совместимости браузеров с новой технологией HTML5. Инструмент не тестирует браузеры на совместимость, и не проверят функционал каждого свойства.

#### 04. What's my IP

Сайт помогает вам определить, какие HTML5/CSS3-функции поддерживаются вашим браузером, а какие нет.

#### 05. SproutCore

SproutCore – это HTML5-приложение для создания привлекательных приложений для любого современного веб-браузера, без применения каких-либо плагинов.

#### 06. SwitchToHTML5

Когда вы создаете новые веб-сайты или обновляете старые до новой технологии, то вы можете по-настоящему ощутить важность платформы. Мы предлагаем вам полезный скрипт, написанный на PHP. Вам будет предоставлена полезная платформа HTML5 каждый раз, когда вы хотите начать новый HTML5-проект.

#### 07. When can I use

Еще один инструмент для проверки совместимости браузеров с новыми технологиями (HTML5, CSS3, SVG).

### **1.3. Лекция № 3 (2 часа)**

#### **Тема: «Проектирование Web - сайта»**

##### **1.3.1. Вопросы лекции:**

1. Этапы создания web - сайтов.

##### **1.3.2 Краткое содержание вопросов:**

Создание Web-сайта предполагает наличие нескольких этапов — от придумывании идеи до ее воплощения. Чтобы создать действительно интересный и полезный продукт, необходимо пройти следующие этапы.

**Анализ и проектирование.** Анализ сильных и слабых сторон конкурентов, информационное проектирование Web-сайта, **разработка** его концепции, оценка целевой аудитории.

**Написание контента.** Под контентом (Content) понимается информационное наполнение сайта. Термин применяется для обозначения идейного содержимого Web-сайта.

**Креатив.** Креатив (Creative) — термин для обозначения визуальной составляющей Web-сайта. Сюда входит разработка дизайна, графических элементов, обработка графики и все, что с ней связано.

**Написание кода Web-сайта.** Программирование Web-файла, написание функциональной части.

**Тестирование.** На этом этапе проверяется все: удобство навигации, целостность данных, корректность ссылок и орфография, к тому же не все браузеры одинаково интерпретируют одни и те же теги HTML.

**Публикация.** Размещение Web-сайта в Интернете. Большинство Интернет-провайдеров (**Internet serviceprovider—JSP**) предоставляют услугу по размещению домашних Web-страниц бесплатно (эта услуга называется Web-хостингом —Webhosting), например <http://www.chat.ru/> (до 20 Мб), <http://www.narod.ru/> (до 100 Мб), <http://www.boom.ru/> (до 50 Мб). Для публикации можно размещать Web-сайт у провайдера или **воспользоваться** другими возможностями,

**Раскрутка.** Рекламная компания по узнаванию Web-сайта и повышения его **посещаемости**. Сюда входит **регистрация** Web-сайта в поисковых системах, обмен ссылками, баннерная реклама и др.

**Поддержка.** Необходимо решить, каким образом будут добавляться новые разделы и материалы, что будет происходить со старыми. Возможно, потребуется создание архива новостей, куда будут попадать новости, потерявшие свою актуальность. Еще более важным является регулярное обновление информации на Web-сайте для его постоянной привлекательности.

#### **1.4. Лекция № 4 (2 часа)**

**Тема: «Язык разметки гипертекста – HTML»**

##### **1.4.1. Вопросы лекции:**

1. Элементы Web-страницы.
2. Общая структура простейшего документа html.

##### **1.4.2 Краткое содержание вопросов:**

**Web-страница** представляет собой набор элементов.

**Тег (Tag)** — это стартовый и конечный маркеры элемента. Теги определяют границы действия элементов и отделяют элементы друг от друга. Теги заключаются в угловые скобки, например: **<HTML>**. Конечный тег, если он предусмотрен синтаксисом языка, снабжается косой чертой: **</HTML>**.

**Гиперссылка (Hyperlinks)** — фрагмент текста, который является указателем на другой файл или объект. Гиперссылки позволяют переходить от одного документа к другому.

**Фрейм (Frame)** — область гипертекстового документа со своими полосами прокрутки. Фреймы позволяют разбивать страницы на прямоугольные области, в каждой из которых отображается своя собственная страница. Вы можете разместить один или несколько

фреймов на странице (такая страница называется страницей фреймов или фреймсет (frameset)).

**Форма (Form)** — область гипертекстового документа, которая необходима для организации обратной связи с посетителями сайта.

**Апплет (Applet)** — небольшая прикладная программа, передаваемая на компьютер клиента в виде отдельного файла и запускаемая при просмотре Web-страницы.

**Скрипт (Script)** — программа, включенная в состав Web-страницы для расширения ее возможностей.

**Загрузка (DownLoad)** — копирование документа с Web-сервера на компьютер клиента.

**Размещение (UpLoad)** — копирование документа с компьютера клиента на Web-сервер — используется при создании собственной Web-страницы (т.е. при ее опубликовании).

**Пиксел (Pixel)** — наименьшая цветная точка экрана монитора.

### **Общая структура простейшего документа html**

Общая структура простейшего документа HTML такова:

**<COMMENT>**Комментарий**</COMMENT>**

**<HTML>**

**<HEAD>**

**<TITLE>**Название документа**</TITLE>**

**</HEAD>**

**<BODY>**

Здесь расположен текст самого документа HTML.

**</BODY>**

**</HTML>**

Дадим пояснения указанным тегам документа HTML.

**<COMMENT>**— комментарий к документу. Не является обязательным.

**<HTML>**— идентификатор всего блокаHTML-команд.

**<HEAD>**— идентификатор заголовка документаHTML.

**<TITLE>**— идентификатор заголовка окна просмотра.

**<BODY>**— идентификаторHTML-команддокумента для просмотра.

Три основных тега **<HTML>**, **<HEAD>**и**<BODY>**передают броузеру основную информацию для идентификации и**организации** документа.

Все указанные теги — парные, т.е. каждый из них заканчивается конечным тегом с косой чертой. Все команды можно писать как с маленькой, так и с большой буквы.

**<META>**— непарный тег применяется для указания подробной информации о документе.

**<P>**— идентификатор конца абзаца, после которого добавляется пустая строка. В конце нужен тег**</P>**.

**<BR>**— идентификатор перевода строки.

**<HR>**— идентификатор изображения горизонтальной линии.

Эти теги одиночные, т.е. непарные, они не требуют тегов с косой чертой.

**<PRE>**— установка равношириного шрифта. В конце нужен тег**</PRE>**.

### **Парные теги форматирования заголовков и подзаголовков документа**

**<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H5>**, **<H6>**.При этом заголовки будут выведены большими буквами, причем размер букв у тега**<H1>**будет самый большой, у **<H2>** — меньше, у**<H3>**— еще меньше и т.д.

### **Парные теги форматирования символов текста**

**<B>**— идентификатор полужирного шрифта.

**<strong>**— идентификатор выделенного шрифта.**<i>**— идентификатор курсива.**<U>**— идентификатор подчеркивания.**<s>**— идентификатор перечеркивания.

**<tt>**— идентификатор равношириного шрифта (телетайпного или курьера).

**<big>**— задает увеличенный размер шрифта.**<small>**— задает уменьшенный размер шрифта.**<center>**— задает центрирование текста. **<sub>**— задает нижний индекс.

**ТЕГИ ФОРМАТИРОВАНИЯ АБЗАЦЕВ** **<p align=left>**— выравнивание текста в абзаце по левому краю.**<p align=right>**— выравнивание текста в абзаце по правому краю.

**<p align=center>**— выравнивание текста в абзаце по центру.**<p align=justify>**— полное выравнивание по обоим краям экрана.**align** — атрибут выравнивания.

### Теги списков

Теги списков являются способом наглядного отображения структурированной информации. Упорядоченные списки служат для отображения последовательных операций или алгоритмов. Браузер автоматически генерирует номера для каждого пункта в списке.

**<OL>**— идентификатор упорядоченного списка. В конце**</OL>**.

Неупорядоченные списки служат для составления перечней, когда порядок следования пунктов несущественен. Перед каждым элементом будет стоять маркер.

**<UL>**— идентификатор неупорядоченного списка. В конце**</UL>**.

Отдельные элементы в упорядоченном и неупорядоченном списках помечают одиночным тегом **<LI>**, а элементы в списках определений **<DL>** тегами **<DT>** для термина и **<DD>** для значения термина.

**<LI>**— идентификатор элемента в упорядоченном и неупорядоченном списке.

Конечный тег **</LI>** может быть опущен.

Проектирование WEB – САЙТОВ.

Создание WEB- сайтов средствами языка HTML/ Теги списков определений. Теги цвета. Теги вставки изображений. Тег перехода к другому файлу. Тег перехода к другой странице. Тег вставки адреса Электронной почты. Тег вставки документа. Теги для создания бегущей строки. Список цветов символов HTML. Теги подключения звука. Теги фреймов. Создание таблиц в HTML.

### Теги списков определений

Списки определений обеспечивают специальное форматирование, как в словарях, для терминов и связанных с ними описаний.

**<DL>**— идентификатор списка определений. В конце**</DL>**.

**<DT>**— идентификатор термина в списке определений.

**<DD>**— идентификатор значений термина в списке определений. **<DL>**

**<DT>**название термина 1

**<DD>**определение термина 1

**<DD>**другое определение термина 1

**</DL>**

### Теги цвета



**<BODY bgcolor=«teal» text=«aqua» link=«red»>**— задает цвет текста(*text*), цвет фона(*bgcolor*) и цвет гипертекстовой ссылки(*link*).

**<BODY background=«back.jpg»>**— задает фоновый рисунок (обои) наWeb-странице в файлеback.jpg.

**<BODYbackground=«\windows\tartan.bmp»>**— фоновый рисунок (обои) взят из файлаtartan.bmp, причем указан путь этого файла.

**<font color=«yellow» size=5>**— задает цвет символов текста(*color*) и их размер(*size*)- В конце нужен тег**</font>**.

#### еги линий

**<hr color=«lime»>**— задает цвет горизонтальной линии.

**<hr color=«red» size=3 width=220 align=center>**— атрибут**size=N** задает толщину линии в пикселях; атрибут**width-M** задает длину линии в пикселях; атрибут**align=center**(или **left**, или **right**) задает смещение линии (ее размещение в центре, или смещение влево, или вправо).

#### Теги для вставки изображений

Графика на страницах — это средство выражения мысли, подчеркивания идеи.

Изображения можно сохранять в нескольких форматах. Тип файлов **GIF**— графический формат обмена, стал первым типом файлов, которые поддерживались в**WWW**. Затем был разработан формат**JPEG**— объединенная группа экспертов фотографии. Это форма эффективнее для представления больших изображений.

Приобрел популярность формат **PNG**— персональная сетевая графика, который заменяетGIF.

При публикации сайта рисунок должен находиться в папке, где хранятся файлы страниц, а **URL-адрес**должен быть реальным.

**JPEG**использует специальную технику компрессии изображений, что является преимуществом для рисунков и фотографий.

**<IMG SRC=«lycos.gif»>**— вставка графического изображения в виде анимационного файла**lycos.gif**. Можно также использовать и графические файлы формата**\*.jpg** (или **\*.jpeg**), **\*.bmp**.

**<IMG SRC=«echomsk.gif» width=«403» height=«263»>**— если при вставке изображения использованы атрибуты **width** (ширина в пикселях) и **height** (высота в пикселях), то при загрузке изображения браузер покажет сначала рамку, где должно быть изображение, а затем уже само изображение, поскольку оно загружается дольше. Таким образом, резервируется место на экране под изображение.

### **1.5. Лекция № 5 (2 часа)**

**Тема: «Структура HTML документа»**

#### **1.5.1. Вопросы лекции:**

1. Основные понятия и определения.
2. . Инструменты и технологии программирования.
3. Структура HTML-документа.

#### **1.5.2 Краткое содержание вопросов:**

1. HTML (HyperTextMarkupLanguage) – язык создания гипертекстовых документов. HTML не является языком программирования, т.к. предназначен только для представления данных (команды языка указывают клиенту, в каком виде должна быть отображена та или иная часть документа). В связи с этим язык HTML имеет следующие ограничения: •не содержит средств обработки информации. •не предоставляет никакой возможности сохранять данные в процессе работы приложения. •имеет ограниченные возможности для взаимодействия с пользователем Любая HTML страница состоит из набора объектов: текста, графики, ссылок, списков и т.д. Для представления этих объектов в языке HTML используются специальные операторы - тэги (от англtag - ярлык).

2. HTML документы представляют собой обычные текстовые документы и могут быть созданы в любом текстовом редакторе, например в Блокноте. Существуют специализированные WYSIWYG (WhatYouSeeIsWhatYouGet) HTML-редакторы, например, AdobeDreamWaver или MS Frontpage. HTML документы принято сохранять в файлы с расширением \*.htm или \*.html. Просмотр HTML документов осуществляется при помощи специальных программ-просмотрщиков (web-браузеров), которые интерпретируют HTML код для отображения на экране. 2.3. Структура HTML

3. HTML со времени своего рождения (1989 г.) постоянно развивался, непрерывно претерпевая изменения и дополнения стандартов. В данном курсе рассматривается последняя версия HTML 4.0. Согласно спецификации HTML 4.0 структура HTML-документа.

### **1.6. Лекция № 6 (2 часа)**

**Тема: «Форматирование текста. Интерактивные формы HTML»**

#### **1.6.1. Вопросы лекции:**

1. Escape-последовательности.
2. Организация списков.

3. Комментарии.
4. Фреймы.
5. Создание HTML форм
6. Элементы управления

#### **1.6.2 Краткое содержание вопросов:**

1. В случае, если кодировка или конфигурация оборудования и программного обеспечения не позволяет вводить определенные символы, например, ©, ®, £ или §, можно использовать ссылки на символы, называемые Escape-последовательностями. Escape-последовательности - это независимый от кодировки механизм ввода любых символов. Escape-последовательности в HTML могут принимать две формы:

-Числовые ссылки на символы, например, © (символ авторского права ©).

-Буквенные обозначения символов, например, ©

2. HTML поддерживает 3 вида списков: нумерованные, пронумерованные и списки-определения. Нумерованные списки создаются при помощи тегов

3. Комментарии в HTML имеют следующий синтаксис: Информация в комментариях не имеет специального значения и никак не влияет на отображение документа.

4. Фреймы делят окно браузера на несколько областей, каждая из которых функционирует независимо от других и может отображать отдельный HTML-документ

5. HTML форма представляет собой раздел HTML документа, содержащего элементы управления: текстовые поля, флажки, кнопки с зависимой фиксацией, меню и т.д. HTML форма задается тегом

6. Элементы управления Пользователи взаимодействуют с формами с помощью управляющих элементов.

#### **1.7. Лекция № 7 (2 часа)**

**Тема: «Введение в JavaScript. Синтаксис языка. Объектная модель JavaScript»**

##### **1.7.1. Вопросы лекции:**

1. Объектная модель браузера
2. Объект window
3. Размещение JavaScript на HTML-странице

#### **1.7.2 Краткое содержание вопросов:**

1. JavaScript является языком клиентским скриптов. Коды программ, написанные на JavaScript, передаются в клиентский браузер и исполняются им. Поэтому важным вопросом изучения языка JavaScript является понимание объектной модели браузера. Объектная модель браузера представляет собой строгую иерархическую структуру, позволяющую обращаться к любой части браузера или загруженных страниц с помощью языка JavaScript.

2. Объект window – главный объект в иерархии объектной модели браузера. Объект window обращается к активному окну.

3. Код скрипта JavaScript может объявляться в HTML-коде следующими способами: 1. Непосредственное размещение в любой части HTML-кода при помощи тега SCRIPT

Размещение в отдельном файле с объявлением в заголовке HTML-документа при помощи тега SCRIPT

Контекстное размещение в обработчике событий.

## **1.8. Лекция № 8 (2 часа)**

### **Тема: «Типы данных. Операторы JavaScript»**

#### **1.8.1. Вопросы лекции:**

1. Правила написания программ JavaScript
2. Типы данных
3. Массивы JavaScript

#### **1.8.2 Краткое содержание вопросов:**

1. Текст программы на JavaScript состоит из последовательности операторов. Именно в такой последовательности операторы и исполняются интерпретатором (браузером). Особенности записи программ на языке JavaScript:

- Синтаксис языка JavaScript близка очень близок языку C++;
- Один оператор в JavaScript может быть разбит на несколько строк, или, наоборот, в одной строке может быть несколько операторов;
- В программе отдельные операторы записываются через точку с запятой (допускается не ставить ; если оператор является в строке последним);
- JavaScript не имеет жестких требований к форматированию текста программы, таким образом возможно использование символов перевода строки и отступов для придания тексту программы лучшей читабельности;

-Однострочные комментарии в JavaScript записываются после символов `//`, многострочные – между символами `/* */`.

2. Типы данных Переменные JavaScript могут иметь один из следующих типов:

- Числовой (целые числа или числа с плавающей точкой)
- Булевый
- Строковый
- Нулевой тип (определяется ключевым словом `null`).

Тип данных при объявлении переменной не указывается. Тип присваивается переменной только тогда, когда ей присваивается какое-либо значение. Переменные в программе объявляются при помощи ключевого слова `var` (допускается опускать)

3. Тип массив введен в JavaScript для возможности манипулирования разными объектами: это список всех гипертекстовых ссылок страницы, список всех картинок на странице, список, список всех элементов формы и т.п. Пользователь может создать и свой собственный массив, используя конструктор `Array()`. Размерность массива может динамически изменяться. Можно сначала определить массив, а потом присвоить одному из его элементов значение. Как только это значение будет присвоено, изменится и размерность массива.

### **1.9. Лекция № 9 (2 часа)**

**Тема: «Обработка событий. Примеры эффективного программирования на JavaScript.»**

#### **1.9.1. Вопросы лекции:**

1. JAVA APPLET.

#### **1.9.2 Краткое содержание вопросов:**

Отдельной позицией в развитии технологии WEB программирования является использование - языка программирования Java и виртуальной Java-машины, встроенной в браузеры. Возможности Java технологии используются, прежде всего, для преодоления негативных особенностей HTTP протокола (в основном для поддержки постоянных TCP-соединений).

Наибольшей мощностью в реализации клиентского программного обеспечения обладают апплеты – программы, написанные на языке JAVA компании Sun Microsystems. Java - объектно-ориентированный язык. Наиболее интересным свойством является то, что программа на Java компилируется в псевдокод, который выполняется на виртуальной машине (естественно, реализация такой машины своя для каждой платформы). Java легко и естественно интегрируется с WWW. Создан Java-подобный упрощенный язык Javascript

для пользователей, которые не являются программистами. Интерпретатор этого языка встроен в браузер, <http://java.sun.com/>. В широком смысле Java - это технология программирования, изначально рассчитанная на интеграцию с Web-сервисом, то есть на использование в сетевой среде независимо от платформы, при этом используются два важнейших постулата. Возможно несколько вариантов использования языка Java.

JAVA APPLET – технология, позволяющая встраивать программы, написанные на языке JAVA в HTML страницы. Совместное использование HTML и Java-апплетов позволяет создать отличное Web решение. В этом случае HTML применяется для отображения меню и выполнения простых интерфейсных функций, а с помощью Java-апплетов обеспечиваются более сложные компоненты интерфейса программы, а также согласованное отображение диаграмм, графиков и таблиц. В таких приложениях чаще всего реализованы функции агрегирования и детализации, а также вращения (pivoting) данных. За счет широких графических возможностей Java по сравнению с HTML, можно представлять данные в виде диаграмм и изменять их в интерактивном режиме.

Web-архитектуры быстро вытесняют традиционные клиент/серверные приложения для целого ряда категорий программного обеспечения. Это направление стремительно развивается за счет появления различных Web инструментов на базе HTML- и Java-технологий от известных поставщиков и быстро растущих новых компаний. Происходит переход от инструментальных средств исследования данных, ориентированных на специалистов-аналитиков, к готовым аналитическим приложениям, доступным для более широкого круга пользователей.

Преимущества: Для использования данной технологии на стороне клиента понадобится только веб-браузер с установленной с поддержкой JAVA APPLETs. Таковыми на сегодняшний день являются большинство браузеров. Applet может получать данные с сервера в виде графических примитивов в бинарном формате данных, хранить уже полученные данные, а также в фоновом режиме получать данные, которые наиболее вероятно будут востребованы, что, безусловно, сократит нагрузку на сеть и сервер. Applet может сам преобразовывать данные их векторной формы в растровую, что тоже снизит нагрузку на сервер. Applet - полноценная программа, следовательно, он позволит реализовать интерфейс пользователя любой сложности.

Недостатки: Недостатками JAVA APPLET является необходимость при каждом новом обращении к html странице скачивать весь applet целиком. Также могут возникнуть трудности с версией JAVA virtual machine (JVM) установленной в браузере.

Вывод: Данная технология может быть применена в случае, если: клиент располагает веб-браузером с поддержкой нужной версии JVM, запросы производятся

интенсивно и требуется реализовать полноценный интерфейс пользователя, объем получаемых данных многократно превосходит объем JAVA APPLET.

Использование виртуальной Java-машины (JVM), на которой должны выполняться (интерпретироваться) Java-программы. Исходные Java-тексты транслируются в коды этой машины. При появлении новой аппаратно-программной платформы в портировании будет нуждаться только Java-машина; все программы, написанные на Java, пойдут без изменений.

Определено, что при редактировании внешних связей Java-программы и при работе браузера прозрачным для пользователя образом может осуществляться поиск необходимых объектов не только на локальной машине, но и на других компьютерах, доступных по сети. Найденные объекты загружаются, а их методы выполняются затем на машине пользователя.

В компилируемой среде трудно абстрагироваться от аппаратных особенностей компьютера, как трудно (хотя и можно) реализовать прозрачную динамическую загрузку по сети. С другой стороны, прием объектов извне требует повышенной осторожности при работе с ними, а, значит, и со всеми Java-программами. Принимать необходимые меры безопасности проще всего в интерпретируемой, а не компилируемой среде. Вообще, мобильность, динамизм и безопасность - спутники интерпретатора, а не компилятора. Java-среда идеальное средство разработки интерактивных клиентских приложений. Стандартный реляционный доступ к данным очень важен для программ на Java, потому что Java-апплеты по природе своей не являются монолитными, самодостаточными программами. Будучи модульными, апплеты должны получать информацию из хранилищ данных, обрабатывать ее и записывать обратно для последующей обработки другими апплетами. Монолитные программы могут себе позволить иметь собственные схемы обработки данных, но Java-апплеты, пересекающие границы операционных систем и компьютерных сетей, нуждаются в опубликовании открытых схем доступа к данным.

## **2. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

### **2.1 Лабораторная работа № 1,2 (4 часа).**

**Тема:** «Обзор Web- технологий»

#### **2.1.1 Задание для работы:**

1. Интернет;
2. Технология “клиент-сервер”.

#### **2.1.2 Краткое описание проводимого занятия:**

В связи с ускоренными темпами развития Интернет технологий постоянно появляются новые направления в области Web - технологий. Поэтому Web-программирование не сформировавшаяся наука, а скорее набор существующих программных технологий (клиентских и серверных), используемых для организации работы пользователя в сети Интернет. Сервер – компьютер, на котором находятся документы и приложения. Клиент – рабочая станция, используемая для просмотра документов и приложений, хранящихся на сервере.

### **2.2 Лабораторная работа № 3,4 (4 часа).**

**Тема:** «Инструменты создания web - сайтов и приложений»

#### **2.2.1 Задание для работы:**

1. Примеры использования DHTML.

#### **2.2.2 Краткое описание проводимого занятия:**

DHTML (динамический HTML) – это набор средств, которые позволяют создавать более интерактивные Web-страницы без увеличения загрузки сервера. Другими словами, определенные действия посетителя ведут к изменениям внешнего вида и содержания страницы без обращения к серверу.

DHTML построен на объектной модели документа (DocumentObjectModel, DOM), которая расширяет традиционный статический HTML-документ. DOM обеспечивает динамический доступ к содержимому документа, его структуре и стилям. В DOM каждый элемент Web-страницы является объектом, который можно изменять. DOM не определяет новых тэгов и атрибутов, а просто обеспечивает возможность программного управления всеми тэгами, атрибутами и каскадными листами стилей

### **2.3 Лабораторная работа № 5,6 (4 часа).**

**Тема:** «Проектирование Web - сайта»

#### **2.3.1 Задание для работы:**

1. Способы создания Web-страниц.

#### **2.3.2 Краткое описание проводимого занятия:**

**Web-сайт** —набор**связанных** между собой, близких по смыслуWeb-страниц и файлов. На каждомWeb-сайте существует однаWeb-страница называемая домашней (homepage) или главной страницей. Все посетителиWeb-сайта сначала попадают на нее, а потом с помощью гиперссылок попадают на другие страницыWeb-сайта,

Web– страница (документHTML) представляет собой текстовый файл на языкеHTMLформата ".htmlили \*.html, размещенный вWWW.Web-страница кроме текста может содержать графику, анимации, видеоклипы, музыку, а также гипертекстовые ссылки, при помощикоторых можно переходить к другимWeb-страницам и просматривать их.



Один из способов создания Web-страниц (или документовHTML) состоит в использовании текстового редактора **Блокнот** (NotePad), встроенного в Windows. В этом редакторе создается файл Web-страницы, который сохраняется с расширением \*.htm с последующим просмотром результатов с помощью браузера. После сохранения файла и выхода из **Блокнота** для просмотра отредактированной страницы надо выполнить команду **Файл — Открыть** в программе **Internet Explorer** и указать путь к созданной Web-странице с помощью кнопки **Обзор**.

## 2.4 Лабораторная работа № 7,8 (4 часа).

**Тема:** «Язык разметки гипертекста – HTML»

### 2.4.1 Задание для работы:

1. Теги HTML.

### 2.4.2 Краткое описание проводимого занятия:

Тег	Описание
<!--...-->	Используется для добавления комментариев.
<!DOCTYPE>	Объявляет тип документа и предоставляет основную информацию для браузера — его язык и версия.
<a>	Создаёт гипертекстовые ссылки.
<abbr>	Определяет текст как аббревиатуру или акроним. Поясняющий текст задаётся с помощью атрибута title.
<address>	Задаёт контактные данные автора/владельца документа или статьи. Отображается в браузере курсивом.
<area>	Представляет собой гиперссылку с текстом, соответствующей определенной области на карте-изображении или активную область внутри карты-изображения. Всегда вложен внутри тега <map>.
<article>	Раздел контента, который образует независимую часть документа или сайта, например, статья в журнале, запись в блоге, комментарий.
<aside>	Представляет контент страницы, который имеет косвенное отношение к основному контенту страницы/сайта.
<audio>	Загружает звуковой контент на веб-страницу.
<b>	Задаёт полужирное начертание отрывка текста, не придавая акцент или важность выделенному.
<base>	Задаёт базовый адрес (URL), относительно которого вычисляются все относительные адреса. Это поможет избежать проблем при переносе страницы в другое место, так как все ссылки будут работать, как и прежде.
<bdi>	Изолирует отрывок текста, написанный на языке, в котором чтение текста происходит справа налево, от остального текста.
<bdo>	Отображает текст в направлении, указанном в атрибуте dir, переопределяя текущее направление написания текста.
<blockquote>	Выделяет текст как цитату, применяется для описания больших цитат.
<body>	Представляет тело документа (содержимое, не относящееся к метаданным документа).
 	Перенос текста на новую строку.
<button>	Создаёт интерактивную кнопку. Внутри тега можно поместить содержимое — текст или изображение.
<canvas>	Холст-контейнер для динамического отображения изображений, таких как простые изображения, диаграммы, графики и т.п. Для рисования используется скриптовый язык JavaScript.
<caption>	Добавляет подпись к таблице. Вставляется сразу после тега <table>.

`<cite>` Используется для указания источника цитирования. Отображается курсивом.

`<code>` Представляет фрагмент программного кода, отображается шрифтом семейства monospace.

`<col>` Выбирает для форматирования один или несколько столбцов таблицы, не содержащих информацию одного типа.

`<colgroup>` Создает структурную группу столбцов, выделяющую множество логически однородных ячеек.

`<datalist>` Элемент-контейнер для выпадающего списка элемента `<input>`. Варианты значений помещаются в элементы `<option>`.

`<dd>` Используется для описания термина из тега `<dt>`.

`<del>` Помечает текст как удаленный, перечёркивая его.

`<details>` Создаёт интерактивный виджет, который пользователь может открыть или закрыть. Представляет собой контейнер для контента, видимый заголовок виджета помещается в тег `<summary>`.

`<dfn>` Определяет слово как термин, выделяя его курсивом. Текст, идущий следом, должен содержать расшифровку этого термина.

`<div>` Тег-контейнер для разделов html-документа. Используется для группировки блочных элементов с целью форматирования стилями.

`<dl>` Тег-контейнер, внутри которого находятся термин и его описание.

`<dt>` Используется для задания термина.

`<em>` Выделяет важные фрагменты текста, отображая их курсивом.

`<embed>` Тег-контейнер для встраивания внешнего интерактивного контента или плагина.

## 2.5 Лабораторная работа № 9,10 (4 часа).

**Тема:** «Структура HTML документа»

### 2.5.1 Задание для работы:

1. Структура документа;
2. Структура и параметры тегов.

### 2.5.2 Краткое описание проводимого занятия:

1. HTML-документ заключается в теги `<html>` и `</html>`. Между этими тегами располагаются два раздела: раздел заголовка (элемент **head**) и раздел тела документа (элемент **body** для простого документа либо элемент **frameset**, задающий набор кадров). Все указанные элементы имеют начальный и конечный тег.
2. Секция заголовка содержит описание параметров, используемых при отображении документа, но не отражающихся непосредственно в окне браузера. Секция тела документа содержит текст, предназначенный для отображения браузером и элементы, указывающие на способ форматирования текста, определяющие графическое оформление документа, задающие параметры гиперссылок и так далее.

## 2.6 Лабораторная работа № 11,12 (4 часа).

**Тема:** «Форматирование текста. Интерактивные формы HTML»

### 2.6.1 Задание для работы:

1. Форматирование текста: представление текстовой информации;
2. Escape-последовательности;
3. Формы HTML: организация формы;
4. Основные параметры формы.

## 2.6.2 Краткое описание проводимого занятия:

1. Все компьютерные файлы можно разделить на две категории: текстовые файлы (*plaintext*) и двоичные (бинарные) файлы. Текстовые файлы содержат обычные символы, такие как буквы, цифры, знаки препинания и некоторое количество управляющих символов, таких как знаки табуляции и перевода строки. Символы обычно кодируются с помощью *ASCII-кодировки* или её расширений для различных языков. *Кодировка* – совокупность всех уникальных символов, которые способна распознать операционная система.

2. Сочетания символов, состоящих из косой черты (\), за которой следует буква или набор цифр, называются *escape-последовательностями*. Для представления знака новой строки, одиночной кавычки или некоторых других символов в символьной константе, необходимо использовать *escape-последовательности*. *Escape-последовательность* рассматривается как один символ и, следовательно, является допустимой символьной константой. *Escape-последовательности* обычно используются для указания действий, например возврата каретки или табуляции, на терминалах и принтерах. Они также используются для предоставления буквенных представлений непечатаемых символов, а также символов, которые обычно имеют специальное значение, например двойных кавычек ("). В следующей таблице перечислены *escape-последовательности ANSI* и представляемые ими значения. Обратите внимание, что вопросительный знак, перед которым стоит обратная косая черта (\?), обозначает литерал вопросительного знака в таких случаях, когда последовательность символов может быть ошибочно интерпретирована как триграф. Дополнительные сведения см. в разделе Триграфы.

Форма (англ. form) в HTML — раздел документа, позволяющий пользователю вводить информацию для последующей обработки системой. Синтаксически форма в HTML задаётся с помощью элемента `<form>` и в дополнение к разметке обычных элементов содержит разметку для элементов управления (англ. controls), надписей.

Элементы управления служат для взаимодействия пользователя с формой. Атрибут `name` определяет имя элемента управления с областью видимости внутри данной формы.

С каждым элементом формы связано начальное и текущее значение. За некоторыми исключениями (`<textarea>`, `<object>`), начальное значение может быть задано атрибутом `value`. Значения, соответствующие элементам, могут изменяться при взаимодействии пользователя или скриптов (например, на Javascript) с формой. При очистке (англ. reset) формы элементы приобретают начальные значения. Данные всех действующих (англ. successful) элементов формы отправляются (англ. submit) на обработку в виде пар имя-значения.

## 2.7 Лабораторная работа № 13,14 (4 часа).

**Тема:** «Введение в JavaScript. Синтаксис языка. Объектная модель JavaScript»

### 2.7.1 Задание для работы:

1. Объектная модель: модель DOM. Синтаксис языка JavaScript.

### 2.7.2. Краткое описание проводимого занятия:

Большинство действий в javascript выполняется с HTML-страницей. В javascript страница представлена в виде объектной модели DOM (DocumentObjectModel).

Любые действия со страницей требуют вызова соответствующего метода DOM. Понимание, как работать с документом в модели DOM - краеугольный камень в javascript-программировании.

- Введение. DOM в примерах.

Основным инструментом работы и динамических изменений на странице является DOM (DocumentObjectModel) - объектная модель, используемая для XML/HTML-документов.

Согласно DOM-модели, документ является иерархией. Каждый HTML-тег образует отдельный элемент-узел, каждый фрагмент текста - текстовый элемент, и т.п.

## **2.8 Лабораторная работа № 15,16 (4 часа).**

**Тема:** «Типы данных. Операторы JavaScript»

### **2.8.1 Задание для работы:**

1. Типы данных, локальные и глобальные переменные, массивы, арифметические и логические операции, циклы и управляющие структуры.

### **2.8.2.Краткое описание проводимого занятия:**

В VBA, как и в других языках программирования высокого уровня, для хранения значений используют переменные и константы. Переменная – именованная область памяти, используемая для хранения значения, которое можно изменить при выполнении программы. Переменные подразделяются на простые и индексированные (переменные с индексом). Индексированными переменными являются элементы массивов. Все подробности о массивах будут рассмотрены в параграфе. Константа – именованная область памяти, используемая для хранения фиксированного значения, которое невозможно изменить при выполнении программы.

## **2.9 Лабораторная работа № 17,18 (4 часа).**

**Тема:** «Обработка событий. Примеры эффективного программирования на JavaScript»

### **2.9.1 Задание для работы:**

1. Примеры эффективного программирования на JS.

### **2.9.2.Краткое описание проводимого занятия:**

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам (встраивается в HTML)

Основные архитектурные черты JavaScript: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.