

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Базовая кафедра Менеджмента и информационных технологий

УТВЕРЖДАЮ:

И.о. проректора по учебной работе

_____ Е.И. Луковникова

«_____» _____ 20 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ИНТЕРНЕТ-ПРОГРАММИРОВАНИЕ

Б1.В.08

НАПРАВЛЕНИЕ ПОДГОТОВКИ

09.03.03 Прикладная информатика

ПРОФИЛЬ ПОДГОТОВКИ

Прикладная информатика в экономике

Программа академического бакалавриата

Квалификация (степень) выпускника: бакалавр

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	4
3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ	4
3.1 Распределение объёма дисциплины по формам обучения.....	4
3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости	5
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	5
4.1 Распределение разделов дисциплины по видам учебных занятий	5
4.2 Содержание дисциплины, структурированное по разделам и темам	6
4.3 Лабораторные работы.....	9
4.4 Практические занятия.....	9
4.5. Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат.....	9
5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ	10
6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ	11
7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	11
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	13
9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	13
9.1. Методические указания для обучающихся по выполнению лабораторных работ	14
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	49
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	49
Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	50
Приложение 2. Аннотация рабочей программы дисциплины	55
Приложение 3. Протокол о дополнениях и изменениях в рабочей программе	56

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к проектному виду профессиональной деятельности выпускника в соответствии с компетенциями и видами деятельности, указанными в учебном плане.

Цель дисциплины

Цель изучения дисциплины Интернет-программирование – изучение и освоение современных Web - технологий для разработки и эксплуатации интернет ресурсов.

Задачи дисциплины

Задачи изучения дисциплины:

- освоение основных концепций и принципов разработки интернет-приложений,
- освоение основных концепций и принципов программирования в интернет
- освоение современных технологий для разработки Web-сайтов,
- приобретение навыков по использованию технологий HTML, CSS, JavaScript в интернет.
- приобретение навыков по программированию в интернет
- приобретение практических навыков публикации и сопровождения Web-сайтов и Web-приложений.

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
1	2	3
ОПК-4	способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Знать: основы организации современных ЭВМ и их общие характеристики, тенденции развития устройств компьютера и компьютерных сетей, принципы организации использования средств вычислительной техники Уметь: применять информационно-коммуникационные технологии при решении стандартных задач профессиональной деятельности Владеть: основными методами решения проблем информационной безопасности компьютерных систем и защиты информации
ПК-8	способностью программировать приложения и создавать программные прототипы решения прикладных задач	знать: <ul style="list-style-type: none">- технологии создания интернет-проектов;- языки: HTML5, CSS3, JavaScript, JQuery;- программное обеспечение для создания интернет-приложений;-предметную область разработки интернет-приложения;- перспективы развития предметной области; уметь: <ul style="list-style-type: none">- разрабатывать интернет-сайт или интернет-приложение по заданной тематике;- проводить оптимизацию структуры сайта и файлов;

		<ul style="list-style-type: none"> - готовить текстовый, графический, видео и аудио контент для сайта; - опубликовать сайт в интернете; - создать онлайн-предприятие для своей будущей профессиональной деятельности; <p>владеть:</p> <ul style="list-style-type: none"> – навыками создания сайтов и веб-приложений по заданной тематике; - навыками написания технического задания, документации пользователя, администратора и разработчика
--	--	--

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.В.ОД.8 Интернет-программирование относится к вариативной части дисциплин и обязательна для изучения.

Дисциплина Интернет-программирование базируется на знаниях, полученных при изучении таких учебных дисциплин, как: Информатика и программирование, Компьютерный практикум.

Основываясь на изучении перечисленных дисциплин, Интернет-программирование представляет основу для изучения дисциплины Информационные системы и технологии, Проектирование информационных систем.

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ

3.1. Распределение объема дисциплины по формам обучения

Форма обучения	Курс	Семестр	Трудоемкость дисциплины в часах						Курсовая работа (проект), контрольная работа, реферат, РГР	Вид промежуточной аттестации (экзамен, зачет)
			Всего часов (с экз.)	Аудиторных часов	Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа		
1	2	3	4	5	6	7	8	9	10	11
Очная	3	5	144	51	17	34	-	57	-	экзамен
Заочная	3		144	16	4	-	12	119	-	экзамен
Заочная (ускоренное обучение)	-	-	-	-	-	-	-	-	-	-
Очно-заочная	-	-	-	-	-	-	-	-	-	-

3.2. Распределение объема дисциплины по видам учебных занятий и трудоемкости

Вид учебных занятий	Трудо- емкость (час.)	в т.ч. в интерактив- ной, актив- ной, иннова- ционной формах, (час.)	Распределение по семестрам, час
			5
1	2	3	4
I. Контактная работа обучающихся с преподавателем (всего)	51	14	51
Лекции (Лк)	17	8	17
Лабораторные работы (ЛР)	34	6	34
II. Самостоятельная работа обучающихся (СР)	57	-	57
Подготовка к лабораторным работам	37	-	37
Подготовка к экзамену в течение семестра	20	-	20
III. Промежуточная аттестация экзамен	36	-	36
Общая трудоемкость дисциплины час.	144		144
зач. ед.	4		4

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной формы обучения:

№ раз- дела и темы	Наименование раздела и тема дисциплины	Трудо- ем- кость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоя- тельная работа обучаю- щихся*
			лекции	лабора- торные работы	
1	2	3	4	5	6
1.	Введение в современные Web-технологии.	16	4	-	12
1.1.	Основные понятия	4	1	-	3
1.2.	Особенности Web-технологий.	4	1	-	3
1.3.	Протоколы интернет.	4	1	-	3
1.4.	Языки Web-программирования	4	1	-	3
2.	Основы Web-технологий.	81	10	34	37
2.1.	Язык разметки гипертекста HTML.	42	6	18	18
2.2.	Каскадные таблицы стилей CSS.	22	2	10	10
2.3.	Основы скриптового языка JavaScript.	17	2	6	9
3.	Введение в Web-дизайн	11	3	-	8

3.1.	Основы веб-дизайна	5	1	-	4
3.2.	Верстка и дизайн Web-страниц.	6	2	-	4
	ИТОГО	108	17	34	57

- для заочной формы обучения:

№ раздела и темы	Наименование раздела и тема дисциплины	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся*
			лекции	практические занятия	
1	2	3	4	5	7
1.	Введение в современные Web-технологии.	11	1	-	10
1.1.	Основные понятия	-	0,25	-	2,5
1.2.	Особенности Web-технологий.	-	0,25	-	2,5
1.3.	Протоколы интернет.	-	0,25	-	2,5
1.4.	Языки Web-программирования	-	0,25	-	2,5
2.	Основы Web-технологий.	94	2	12	80
2.1.	Язык разметки гипертекста HTML.	-	1	6	40
2.2.	Каскадные таблицы стилей CSS.	-	0,5	3	20
2.3.	Основы скриптового языка JavaScript.	-	0,5	3	20
3.	Введение в Web-дизайн	30	1	-	29
3.1	Основы веб-дизайна	-	0,5	-	15
3.2.	Верстка и дизайн Web-страниц.	-	0,5	-	14
	ИТОГО	135	4	12	119

4.2. Содержание дисциплины, структурированное по разделам и темам

№ раздела и темы	Наименование раздела и темы дисциплины	Содержание лекционных занятий (Вид занятия в интерактивной, активной, инновационной формах, (час.)
1	2	3	4
1.	Введение в современные Web-технологии		
1.1.	Основные понятия	История возникновения и развития Интернет. Краткая история развития сети Интернет. Появление сети ARPANET. Объединение сетей. Основные понятия и принципы работы, возможности Интернет. Протоколы Интернет TCP/IP. Понятие виртуального канала, маршрутизация. Основные понятия: всемирная паутина;	-

		интернет; web-технологии; контент; URI (Uniform Resource Identifier - «унифицированный идентификатор ресурса») страницы. Классификация и характеристика Web-сайтов по различным признакам. Распространенные форматы данных в Интернете. Архитектура клиент-сервер. Основы работы Web-сервера. Назначение браузеров, основные понятия. Использование и виды браузеров. Понятие сервера, сайта, домашней страницы, их основное различие.	
1.2.	Особенности Web-технологий.	Web-технологии. особенности; достоинства и недостатки; интеграционная роль. Web-серверы и Web-браузеры. Взаимодействие Web-сервера и Web-браузера. Обзор существующих веб-технологий: JAVA, CGI, SSI, CSS, PHP, ASP, VBScript, Macromedia Flash, DHTML, XHTML, XML; их назначение и недостатки.	-
1.3.	Протоколы интернет.	HTTP (HyperText Transfer Protocol — «протокол передачи гипертекста»). Структура. Методы запросов. Система адресации в Интернет. Понятие хоста, доменная система имен. Электронная почта, протоколы SMTP и POP3. Сквозные протоколы и шлюзы, протоколы UDP и FTP. Доступ к сети Интернет.	-
1.4.	Языки Web-программирования	Языки Web-программирования. Языки сценариев (скриптовые языки), виды языков: клиентские языки; серверные языки.	-
2.	Основы Web-технологий		
2.1.	Язык разметки гипертекста HTML.	Язык разметки гипертекста HTML. Инструментарий для разработки Web-страниц. Современные стандарты языка HTML. Консорциум Всемирной паутины (W3C). Структура HTML кода. Элементы языка HTML и их классификация. Составные элементы HTML-документа. Типы данных HTML. Структура HTML-документа. Язык разметки веб-документов HTML, назначение, основные элементы. Структура файла в формате HTML. Теги тела документа и гипертекстовые ссылки. Использование графики в HTML. Средства описания таблиц и списков в HTML. Элементы HTML – формы. Создание страниц HTML – документов, содержащих фреймы.	Лекция- беседа (4 часа)
2.2.	Каскадные таблицы стилей CSS.	Каскадные таблицы стилей (CSS).Стилевое оформление HTML-документов. Использование каскадных таблиц стилей	Обсуждение ситуаций (2 часа)

		<p>CSS для оформления Web-страниц. Принцип разделения формы и содержания. Технология CSS (каскадные таблицы стилей), назначение, применение. Селекторы, принцип наследования и переопределение. Селекторы, принцип наследования и переопределение. Классы и идентификаторы в CSS. Свойства элементов CSS: шрифта, текста, цвета и фона, границ, списков и т.п. Способы подключения CSS-файла к веб-странице.</p>	
2.3.	<p>Основы скриптового языка JavaScript.</p>	<p>Технология JavaScript, назначение и применение. Понятие объектной модели применительно к JavaScript. Массивы, функции, объекты в JavaScript. Операторы языка JavaScript. Размещение сценариев. Структура сценариев на JavaScript. Типы и структуры данных в JavaScript. Переменные и литералы. Операторы. Конструкции языка JavaScript. Функции пользователя. Проверка условий. Циклы. Объекты языка JavaScript (Array, String, Date, Math, Number, Function). Создание пользовательских объектов. Знакомство с объектной моделью документа. Объект navigator, screen, location, history, event. Объекты элементов управления формами. Передача, обработка и проверка данных форм. Динамическое изменение элементов формы. Программирование свойств окна браузера. Управление окнами и фреймами. Работа с фреймами. Всплывающие окна. Открытие новых окон браузера. Управление вспомогательными окнами.</p>	<p>Лекция- беседа (2 часа)</p>
3.	Введение в Web-дизайн		
3.1.	<p>Основы веб-дизайна</p>	<p>Цели, с которыми создаются сайты. Основные элементы Web-страниц. Классификация Web-сайтов по признакам их компоновки. Принципы компоновки Web-страницы. Средства разработки Web-сайта. Основные постулаты веб-дизайна. Элементы веб-страницы. Алгоритм по разработке веб-сайта. Веб-хостинг, понятие и разновидности. Понятие о баннере и разновидности и о службах баннерного обмена.</p>	-
3.2.	<p>Верстка и дизайн Web-страниц.</p>	<p>Элементы веб-страницы. Алгоритм по разработке веб-сайта. Понятие макетов Web-страниц их виды и особенности. Веб-хостинг, понятие и разновидности. Понятие о баннере и разновидности и о службах баннерного обмена.</p>	-

4.3. Лабораторные работы

<i>№ п/п</i>	<i>Номер раздела дисциплины</i>	<i>Наименование лабораторной работы</i>	<i>Объем (час.)</i>	<i>Вид занятия в интерактивной, активной, инновационной формах, (час.)</i>
1	1.	Приемы работы в браузерах. Поиск информации в Internet	2	Работа в малых группах (2 часа)
2	2.	Создание Web-сайта	26	Работа в малых группах (2 часа)
3	3.	Принципы Web-дизайна	6	Работа в малых группах (2 часа)
ИТОГО			34	6

4.4. Практические занятия

<i>№ п/п</i>	<i>Номер раздела дисциплины</i>	<i>Наименование практической работы</i>	<i>Объем (час.)</i>	<i>Вид занятия в интерактивной, активной, инновационной формах, (час.)</i>
1	1.	Приемы работы в браузерах. Поиск информации в Internet	2	Работа в малых группах (2 часа)
2	2.	Создание Web-сайта	8	-
3	3.	Принципы Web-дизайна	2	-
ИТОГО			12	2

4.5. Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат

Учебным планом не предусмотрено.

5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

<i>№, наименование разделов дисциплины</i>	<i>Компетенции</i>	<i>Кол-во часов</i>	<i>Компетенции</i>		Σ <i>комп.</i>	<i>t_{ср} час</i>	<i>Вид учебных занятий</i>	<i>Оценка результатов</i>
			<i>ОПК</i>	<i>ПК</i>				
			<i>4</i>	<i>8</i>				
<i>1</i>		<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
1. Введение в современные Web-технологии		28	+	+	2	14	Лк., ЛР	защита ЛР, экзамен
2. Основы Web-технологий		93	+	+	2	46,5	Лк., ЛР	защита ЛР, экзамен
3. Введение в Web-дизайн		23	+	+	2	11,5	Лк., ЛР	защита ЛР, экзамен
<i>всего часов</i>		144	72	72	2	72		

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

а) Подготовка к лекционным и лабораторным занятиям

1. Гладких, Б.А. Информатика от абака до интернета. Введение в специальность: учебное пособие / Б.А. Гладких. – Томск: Издательство «НТЛ», 2005. – 484 с. – ISBN 5-89503-259-1; То же [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=201174> (09.11.2017).

2. Грошев, А.С. Информатика: лабораторный практикум / А.С. Грошев; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования Северный (Арктический) федеральный университет им. М.В. Ломоносова. - Архангельск: ИД САФУ, 2014. - 155 с.: ил. - Библиогр. в кн. - ISBN 978-5-261-00924-5; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=312295> (09.11.2017).

3. Прохоренок Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 912 с.

б) Самоподготовка и самопроверка

1. Сборник тестовых заданий по дисциплине «Информатика и ИКТ». [Электронный ресурс]. - URL: sfrsgpa.ru/wp-content/uploads/2016/11/85-fedorova.pdf. 28.03.2018.

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№	Наименование издания (автор, заглавие, выходные данные)	Вид занятия	Количество экземпляров в библиотеке, шт.	Обеспеченность, (экз./ чел.)
1	2	3	4	5
Основная литература				
1.	Савельев, А.О. HTML5. Основы клиентской разработки / А.О. Савельев, А.А. Алексеев. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 272 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429150 (10.05.2018).	Лк, ПЗ, ЛР	ЭР	1
2.	Крахоткина, Е.В. Технологии разработки Internet-приложений: учебное пособие / Е.В. Крахоткина; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2016. - 124 с. : ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=459070	Лк, ПЗ, ЛР	ЭР	1
3.	Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 265 с.: ил.; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428953	Лк, ПЗ, ЛР	ЭР	1

	(09.11.2017).			
4.	Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с.: ил.; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429078 (09.11.2017).	Лк, ПЗ, ЛР	ЭР	1
5.	Информационные технологии. HTML и XHTML: учебное пособие / А.И. Костюк, С.М. Гушанский, М.Ю. Поленов, Б.В. Катаев; Министерство образования и науки Российской Федерации, Южный федеральный университет, Инженерно-технологическая академия. - Таганрог: Издательство Южного федерального университета, 2015. - 131 с.: ил. - Библиогр. в кн. - ISBN 978-5-9275-1329-1; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=461923 (03.04.2018).	Лк, ПЗ, ЛР	ЭР	1
Дополнительная литература				
6.	Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с.: ил.; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429078 (09.11.2017).	Лк, ПЗ, ЛР	ЭР	1
7.	Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 265 с.: ил.; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428953 (09.11.2017).	Лк, ПЗ, ЛР	ЭР	1
8.	Введение в программные системы и их разработку / С.В. Назаров, С.Н. Белоусова, И.А. Бессонова и др. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 650 с.: схем., табл., ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429819 (09.11.2017).	Лк, ПЗ, ЛР	ЭР	1
9.	Информационные технологии. HTML и XHTML: учебное пособие / А.И. Костюк, С.М. Гушанский, М.Ю. Поленов, Б.В. Катаев; Министерство образования и науки Российской Федерации, Южный федеральный университет, Инженерно-технологическая академия. - Таганрог: Издательство Южного федерального университета, 2015. - 131 с.: ил. - Библиогр. в кн. - ISBN 978-5-9275-1329-1; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=461923	Лк, ПЗ, ЛР	ЭР	1

	(03.04.2018).			
10.	Информационные технологии : лабораторный практикум / Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет» ; авт.-сост. С.В. Говорова, М.А. Лапина. - Ставрополь : СКФУ, 2016. - 168 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=459048 (10.05.2018).	Лк, ПЗ, ЛР	ЭР	1

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ
http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=
2. Электронная библиотека БрГУ <http://ecat.brstu.ru/catalog>
3. Электронно-библиотечная система «Университетская библиотека online»
<http://biblioclub.ru>
4. Электронно-библиотечная система «Издательство «Лань» <http://e.lanbook.com>
5. Информационная система "Единое окно доступа к образовательным ресурсам"
<http://window.edu.ru>
6. Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru>
7. Университетская информационная система РОССИЯ (УИС РОССИЯ)
<https://uisrussia.msu.ru/>
8. Национальная электронная библиотека НЭБ <http://xn--90ax2c.xn--p1ai/how-to-search>
/договор №101/НЭБ/2318 от 03.07.2017
9. http://mathserfer.com/problast.php?tema=vect_act
10. http://libedu.ru/l_b/minorskii_v_p/_sbornik_zadach_po_vysshei_matematike.html
11. <http://www.exponenta.ru/educat/news/kuleshov/index.asp>
12. <http://www.allmath.ru/>

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Обучающийся должен разработать собственный режим равномерного освоения дисциплины. Подготовка студента к предстоящей лекции включает в себя ряд важных познавательных-практических этапов:

- чтение записей, сделанных в процессе слушания и конспектирования предыдущей лекции, вынесение на поля всего, что требуется при дальнейшей работе с конспектом и учебником;
- техническое оформление записей (подчеркивание, выделение главного, выводов, доказательств);
- выполнение практических заданий преподавателя;
- знакомство с материалом предстоящей лекции по учебнику и дополнительной литературе.

Активная работа на лекции, ее конспектирование, продуманная, целенаправленная, систематическая, а главное - добросовестная и глубоко осознанная последующая работа над конспектом - важное условие успешного обучения студентов.

9.1. Методические указания для обучающихся по выполнению лабораторных работ/практических работ

Лабораторные работы в дисциплине предусмотрены для разделов **1. Введение в современные Web-технологии; 2. Основы Web-технологий; 3. Введение в Web-дизайн.** Курс Интернет-программирование предназначен для студентов дневной формы обучения, нацелен на системное изучение различных видов компьютерных технологий и возможности их применения для создания веб-документов. Важное место в курсе отводится приобретению практического навыка работы с наиболее современными интернет-технологиями.

Одной из особенностей профессиональной компетенции современного специалиста является владение им информационной технологией работы в различных программных средах, относящихся к классу интернет технологий, что создает базу для изучения последующих дисциплин. Веб-технологии служат программным инструментарием решения функциональных задач.

Практические занятия по дисциплине Интернет-программирование в дисциплине предусмотрены для разделов **1. Введение в современные Web-технологии; 2. Основы Web-технологий; 3. Введение в Web-дизайн** позволяет студенту более глубоко разобраться в теоретическом материале, определить сферы его практического применения и предназначен для студентов заочной формы обучения. Основная цель практического занятия – развитие самостоятельности студента. Подготовка к практическим занятиям состоит в добросовестном анализе теоретического материала, составлении кратких справочников, словариков, схем, алгоритмов. Кроме того, все домашние задания к практическому занятию должны быть выполнены, либо подготовлены вопросы преподавателю, раскрывающие трудности в освоении учебного материала.

Данный практикум предназначен для студентов дневной и заочной формы обучения. Основная задача практикума – сориентировать студента в разрозненной информации по интернет технологиям и показать ему целостную картину данного направления.

Структура, содержание и оформление лабораторных работ

Лабораторная работа – небольшой научный отчет, обобщающий проведенную студентом работу, которую представляют для защиты преподавателю. К лабораторным работам предъявляется ряд требований, основным из которых является полное, исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения заданий и профессиональной подготовке студентов.

В отчет по лабораторной / практической работе должны быть включены следующие пункты: цель работы; краткие теоретические сведения; описание методики эксперимента; экспериментальные результаты; анализ результатов работы.

Цель работы должна отражать тему лабораторной работы, а также конкретные задачи, поставленные студенту на период выполнения работы. По объему цель работы в зависимости от сложности и многозадачности работы составляет от трех до восьми строк.

Краткие теоретические сведения содержат краткое теоретическое описание изучаемого в работе явления или процесса, а также необходимые расчетные формулы. Материал подраздела не должен копировать содержание методического пособия или учебника по данной теме, он ограничивается изложением основных понятий и законов, расчетных формул, таблиц, требуемых для дальнейшей обработки полученных экспериментальных результатов. Объем литературного обзора не должен превышать 1/3 части всего отчета.

Описание методики эксперимента включает методику проведения эксперимента, процесс получения данных и способ их обработки. Если используются стандартные пакеты компьютерных программ для обработки экспериментальных результатов, то необходимо обосновать возможность и целесообразность их применения, а также подробности обработки данных с их помощью.

Экспериментальные результаты – непосредственно результаты, полученные в ходе проведения лабораторных работ: экспериментально или в результате компьютерного моделирования. Это определенные значения величин, графики, таблицы, диаграммы.

Анализ результатов работы. Раздел отчета должен содержать подробный анализ полученных результатов. Следует сравнить полученные результаты с известными литературными данными, обсудить их соответствие существующим теоретическим моделям. Если обнаружено несоответствие полученных результатов и теоретических расчетов или литературных данных, необходимо обсудить возможные причины этих несоответствий.

Лабораторная работа № 1.

Приемы работы в браузерах. Поиск информации в Internet.

Цель работы:

1. Изучение структуры IP-адреса;
2. Ознакомление с наиболее популярными утилитами для диагностики сетевой конфигурации и сетевых соединений;
3. Ознакомление с основами протокола HTTP.

Задание:

1. Изучить теоретические сведения

Порядок выполнения:

- 1) прочитайте лекцию, теоретические сведения и рассмотрите приведенные в лекции образцы решения заданий;
- 2) выполнять задания лабораторного занятия аналогично.

Форма отчетности:

Подготовить отчет, содержащий следующие пункты:

- Титульный лист;
- Содержание;
- Краткая теория;
- Результаты работы (скриншоты выполненных заданий);
- Выводы.

Выполнить лабораторное задание и сохранить созданные файлы в папке с именем ЛР
Веб фамилия группа.

Задания для самостоятельной работы:

Задание 1

Просмотреть сетевые настройки

1) С помощью утилиты *ipconfig* (запускается в командной строке командой *ipconfig*) определите IP-адрес и маску подсети для своего компьютера.

2) Определите класс подсети, в которой находится ваш компьютер без использования маски подсети и по маске подсети.

3) Определите адрес подсети, в которой находится ваш компьютер, с использованием функции “Логическое И” над IP-адресом и маской подсети. Следует иметь в виду, что операция “Логическое И” должна производиться с двоичным представлением операндов.

Задание 2

Проверить доступность хостов

С помощью утилиты *ping* (запускается в командной строке командой *ping*) проверьте доступность хостов, минимальное, среднее и максимальное время приема-передачи ICMP пакетов до них. Можно рассмотреть хосты, например в следующей последовательности:

- 1) Сервер вашего непосредственного провайдера или сервера вашей подсети;
- 2) Какой-либо сервер вашего региона;
- 3) Веб-сервер Интернет-Университета Информационных Технологий: *www.intuit.ru*;
- 4) Веб-сервер Университета в Кембридже: *www.cam.ac.uk*;
- 5) Веб-сервер Университета в Калифорнии: *www.ucla.edu*;
- 6) Веб-сервер Университета в Токио: *www.u-tokyo.ac.jp*;
- 7) Веб-сервер компании Майкрософт: *www.microsoft.com*.

Обратите внимание, что в последнем случае ICMP-пакеты блокируются веб-сервером.

Задание 3

Определить маршруты следования

С помощью утилиты *tracert* (запускается в командной строке командой *tracert*) определите маршруты следования и время прохождения пакетов до хостов, приведенных в задании 2.

Задание 4

Просмотреть активные текущие сетевые подключения

1) С помощью утилиты *netstat* (запускается в командной строке командой *netstat*) посмотрите активные текущие сетевые подключения и их состояние на вашем компьютере.

2) Запустите несколько экземпляров веб-браузера, загрузив в них веб-страницы с разных веб-серверов. Посмотрите с помощью *netstat*, какие новые сетевые подключения появились в списке.

3) Закрывайте браузеры и с помощью *netstat* проверяйте изменение списка сетевых подключений.

Задание 5

Ознакомиться с протоколом HTTP с помощью утилиты telnet.

1) Запустите сеанс *telnet* (запускается в командной строке командой *telnet*). При этом появится подсказка *Microsoft Telnet>*. С полным списком команд можно ознакомиться с помощью команды *help*.

2) Разрешите режим отображения вводимых с клавиатуры символов с помощью команды *set localecho*.

3) В соответствии с протоколом HTTP необходимо установить соединение с веб-сервером. Для этого с помощью команды *open* устанавливается соединение, например: *open www.yandex.ru 80*.

4) Сформируйте клиентский запрос. Как минимум он должен содержать строку состояния, например:

```
GET HTTP://WWW.YANDEX.RU/INDEX.HTML HTTP/1.0
```

Если поля запроса отсутствуют, то ввод заканчивается двумя нажатиями клавиши <ENTER> для вставки пустой строки после заголовка.

Следует обратить внимание на то, что при вводе нельзя допускать ошибок, поскольку при попытке их исправить с помощью клавиши <BACKSPACE>, ее нажатие интерпретируется как часть запроса.

5) Изучите полученный ответ сервера. Обратите внимание на код ответа в строке состояния ответа веб-сервера в строке состояния и поля заголовка ответа.

Если ответ сервера очень большой (в первую очередь из-за размера документа в теле ответа), то содержимое ответа сервера в окне интерпретатора командной строки обрезается с начала. В этом случае рекомендуется для просмотра заголовка вместо метода *GET* использовать метод *HEAD*.

Задание 6

Ознакомиться со структурой IP адреса

IP-адрес состоит из двух частей: номера сети и номера узла в сети. Самой распространенной является запись IP-адреса в виде четырех чисел, разделенных точками, каждое из которых представляет значение байта в десятичной форме, например: 213.180.204.11. Запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла.

Для разделения этих частей обычно используется 2 подхода:

- С помощью маски (RFC 950, RFC 1518), представляющей собой число в паре с IP-адресом. С помощью операции «логическое И» над этими двумя числами выделяется номер сети.

- С помощью классов адресов (RFC 791).

Вводится пять классов адресов: A, B, C, D, E.

A, B, C – используются для адресации сетей, D и E – имеют специальное назначение. Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса.

Таблица 1. Распределение адресов в IP сетях.

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0 (0 - не используется)	126.0.0.0 (127 - зарезервирован)	2^{24} (3 байта)
B	10	128.0.0.0	191.255.0.0	2^{16} (2 байта)
C	110	192.0.0.0	223.255.255.0	2^8 (1 байт)
D	1110	224.0.0.0	239.255.255.255	групповые адреса
E	11110	240.0.0.0	247.255.255.255	зарезервировано

В рамках IP протокола существуют ограничения при назначении IP-адресов:

- номера сетей и номера узлов не могут состоять из двоичных нулей или единиц;
- если IP-адрес состоит только из двоичных нулей, то он называется неопределенным адресом и обозначает адрес того узла, который сгенерировал этот пакет;
- если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет; такой адрес может быть использован только в качестве адреса отправителя;
- если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета; такой адрес называется *ограниченным широковещательным*, поскольку пакет не сможет выйти за границы сети;
- если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет рассылается всем узлам сети, номер которой указан в адресе назначения; такой тип адреса называется *широковещательным*;
- если первый октет адреса равен 127, то такой адрес называется внутренним адресом стека протоколов; он используется для тестирования программ, организации клиентской и серверной частей приложений, установленных на одном компьютере;
- групповые адреса, относящиеся к классу D, предназначены для экономичного распространения в Интернете, большой корпоративной сети аудио- или видеопрограмм.

Стандартным классам сетей можно поставить в соответствие следующие значения маски: класс А – 255.0.0.0; класс В – 255.255.0.0; класс С – 255.255.255.0;

Исходные данные	<i>IP адрес</i>	62.76.167.21
	<i>Маска сети</i>	255.255.255.0
Логическая операция	И	
Результат	<i>Адрес сети</i>	62.76.167.0
	<i>Номер компьютера</i>	21

Пример

Для определения сетевых настроек компьютера и сетевого оборудования, диагностики и получения другой информации, относящейся к интернет-протоколам, широко используются специальные утилиты.

Задание 7

Ознакомиться с утилитой *ipconfig*.

Ipconfig - это утилита командной строки для вывода деталей текущего соединения компьютера с сетью и контроля над клиентским сервисом DHCP. DHCP (Dynamic Host Configuration Protocol) - это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

Синтаксис команды: ***ipconfig*/ключи**

Команда *ipconfig/all* - отображает полную информацию по всем сетевым адаптерам.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

Z:\>ipconfig/all

Настройка протокола IP для Windows

Имя компьютера . . . . . : CLR385N16
Основной DNS-суффикс . . . . . : cs.vsu.ru
Тип узла . . . . . : неизвестный
IP-маршрутизация включена . . . . . : нет
DNS-прокси включен . . . . . : нет
Порядок просмотра суффиксов DNS . . . . . : cs.vsu.ru
vsu.ru

Подключение по локальной сети 3 - Ethernet адаптер:

DNS-суффикс этого подключения . . . :
Описание . . . . . : VMware Virtual Ethernet Adapter for
VMnet8
Физический адрес . . . . . : 00-50-56-C0-00-08
DHCP включен . . . . . : нет
IP-адрес . . . . . : 192.168.111.1
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . :

Подключение по локальной сети 2 - Ethernet адаптер:

DNS-суффикс этого подключения . . . :
Описание . . . . . : VMware Virtual Ethernet Adapter for
VMnet1
Физический адрес . . . . . : 00-50-56-C0-00-01
DHCP включен . . . . . : нет
IP-адрес . . . . . : 192.168.61.1
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . :

Подключение по локальной сети - Ethernet адаптер:

DNS-суффикс этого подключения . . . : cs.vsu.ru
Описание . . . . . : Intel(R) PRO/100 VE Network Connecti
on #2
Физический адрес . . . . . : 00-14-85-17-CD-6C
DHCP включен . . . . . : да
Автонастройка включена . . . . . : да
IP-адрес . . . . . : 10.16.1.146
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . : 10.16.1.2
DNS-сервер . . . . . : 62.76.220.205
DNS-серверы . . . . . : 62.76.220.205
62.76.220.204
Аренда получена . . . . . : 12 ноября 2008 г. 8:08:31
Аренда истекает . . . . . : 12 ноября 2008 г. 23:08:31

Z:\>_

```

Пример вывода для Windows

Задание 8

Ознакомиться с утилитой *ping*

Ping (Packet InterNet Grouper) - это системная программа, предназначенная для проверки соединений в сетях на основе TCP/IP. Она отправляет *Echo-Request* запросы протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT, Round Trip Time) позволяет определять двусторонние задержки (RTT) по маршруту и частоту потери пакетов. Что позволяет косвенно определять загруженность каналов передачи данных и промежуточных устройств. Полное отсутствие ICMP-ответов может также означать, что удаленный узел (или какой-

либо из промежуточных маршрутизаторов) блокирует ICMP Echo-Reply или игнорирует ICMP Echo-Request.

Синтаксис: **ping** –параметры конечное_имя

Конечное имя – это доменное имя или IP-адрес хоста

```
C:\>ping www.mail.ru

Обмен пакетами с www.mail.ru [194.67.57.26] по 32 байт:

Ответ от 194.67.57.26: число байт=32 время=11мс TTL=118
Ответ от 194.67.57.26: число байт=32 время=11мс TTL=118
Ответ от 194.67.57.26: число байт=32 время=11мс TTL=118
Ответ от 194.67.57.26: число байт=32 время=11мс TTL=118

Статистика Ping для 194.67.57.26:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь),
Приблизительное время приема-передачи в мс:
    Минимальное = 11мсек, Максимальное = 11 мсек, Среднее = 11 мсек
```

Пример:

Задание 9

Ознакомиться с утилитой *tracert*

Tracert (сокращенно *tracert*) — это служебная программа, предназначенная для определения маршрутов следования пакетов в сетях *TCP/IP*. Работа *tracert* основана на протоколе *ICMP*.

Tracert выполняет отправку пакетов указанному узлу сети, отображая при этом сведения о всех промежуточных маршрутизаторах, через которые прошли пакеты на пути к целевому узлу. В случае проблем при доставке пакетов до какого-либо узла программа *tracert* позволяет определить, на каком именно участке сети возникли неполадки.

Синтаксис: **tracert** –параметры конечное_имя

Конечное имя – это доменное имя или IP-адрес хоста

```
Z:\>tracert www.mail.ru

Трассировка маршрута к www.mail.ru [194.67.57.26]
с максимальным числом прыжков 30:

 1  <1 мс  <1 мс  <1 мс  c1swr1.cs.vsu.ru [10.16.1.2]
 2  <1 мс  <1 мс  <1 мс  c2swr1-e1000-4-1-ve100.vsu.ru [62.76.168.42]
 3  <1 мс  <1 мс  <1 мс  c2r1-ve5.vsu.ru [62.76.168.50]
 4  27 ms  22 ms  27 ms  RBNet-USU.vsu.ru [62.76.168.1]
 5  28 ms  *      30 ms  cisco13.Moscow.gldn.net [193.232.244.43]
 6  *      *      *      Превышен интервал ожидания для запроса.
 7  36 ms  45 ms  29 ms  cat07.Moscow.gldn.net [194.186.157.82]
 8  24 ms  46 ms  29 ms  cat01.Moscow.gldn.net [194.186.158.110]
 9  44 ms  27 ms  23 ms  mailru-KK12-1-gw.Moscow.gldn.net [195.239.8.10]

10  36 ms  47 ms  49 ms  mail.ru [194.67.57.26]

Трассировка завершена.

Z:\>_
```

Пример

Задание 10

Ознакомиться с утилитой *netstat*

Netstat – служебная программа, отображающая статистику протокола и текущих сетевых подключений *TCP/IP*.

```

C:\WINDOWS\system32\cmd.exe
Z:\>netstat

Активные подключения

Имя      Локальный адрес      Внешний адрес      Состояние
TCP      C1R385N16:1691      csfs.cs.vsu.ru:microsoft-ds ESTABLISHED
TCP      C1R385N16:1713      www2.cs.vsu.ru:microsoft-ds ESTABLISHED
TCP      C1R385N16:1715      fs.cs.vsu.ru:microsoft-ds ESTABLISHED
TCP      C1R385N16:1856      cache.vsu.ru:3128    CLOSE_WAIT
TCP      C1R385N16:1857      cache.vsu.ru:3128    CLOSE_WAIT
TCP      C1R385N16:1878      cache.vsu.ru:3128    ESTABLISHED
TCP      C1R385N16:1879      cache.vsu.ru:3128    ESTABLISHED
TCP      C1R385N16:1880      cache.vsu.ru:3128    ESTABLISHED
TCP      C1R385N16:1881      cache.vsu.ru:3128    ESTABLISHED

Z:\>

```

Пример

Задание 11

Ознакомиться с утилитой *telnet*

Telnet - сетевой протокол для реализации текстового интерфейса по сети. Название «*telnet*» имеет также утилита, реализующая клиентскую часть протокола. Исторически *telnet* служил для удалённого доступа к интерфейсу командной строки операционных систем. Протокол *telnet* может использоваться для выполнения отладки других протоколов на основе транспорта TCP.

Утилита *telnet* поддерживает следующие команды:

- *Close* – закрытие текущего подключения.
- *Display* – отображение параметров операции.
- *Open* – подключение к сайту.
- *Quit* – выход из *telnet*.
- *Set* – установление параметров.
- *Send* – отправление строки на сервер.
- *Status* – вывод сведений о текущем состоянии.
- *Unset* – сброс параметров.

Используя утилиту *telnet* можно, например, вручную отправить запрос клиента и получить ответ сервера по протоколу HTTP.

Для этого выполним следующую последовательность действий:

1. Запуск утилиты *telnet*
2. Установление соединения с веб-сервером с помощью команды: *open имя_хоста 80*
3. Формирование запроса клиента
4. Получение ответа сервера

```

Telnet localhost
Добро пожаловать в программу-клиент Microsoft Telnet
Символ переключения режима: 'CTRL+I'
Microsoft Telnet> open localhost 80
Подключение к localhost..._

```

1) Устанавливаем соединение: *open localhost 80*

```

Telnet localhost
GET HTTP://LOCALHOST/PERLCALC.HTML HTTP/1.0ft Telnet
Символ переключения режима: 'CTRL+I'
Microsoft Telnet> open localhost 80
Подключение к localhost...

```

2) Формируем строку состояния запроса клиента: ***GET HTTP://LOCALHOST/PERLCALC.HTML HTTP/1.0*** <ENTER><ENTER>

```
Telnet localhost
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
X-Powered-By: ASP.NET
Date: Mon, 17 Nov 2008 09:55:11 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Sun, 16 Nov 2008 16:36:38 GMT
ETag: "b66a667f948c91:9d1"
Content-Length: 426
<html>
<body>
  <form action='http://localhost/Scripts/test.pl'>
    <p>Operand1: <input type='text' name='A'></p>
    <p>Operand2: <input type='text' name='B'></p>
    <p>Operation:<br>
    <select name='op'>
      <option value='+'>+</option>
      <option value='- '>-</option>
      <option value='* '>*</option>
      <option value='/'>/'</option>
    </select></p>
    <input type='submit' value='Calculate!'>
  </form>
</body>
</html>
Подключение к узлу утеряно.
Нажмите любую клавишу...
```

3) Получаем ответ сервера: ответ веб-сервера *localhost* содержит строку состояния (с кодом успешного завершения 200), поля заголовка (*Server*, *Date*, *Content-type* и др.) и тело, содержащее HTML код запрошенного клиентом документа *http://localhost/perlcalc.html*.

Пример

Задание 12

Определить IP-адрес

Определить по заданному доменному имени хоста его IP адрес.

Задание 13

Определить IP-адрес

Определить IP адрес хоста www.mail.ru.

Задание 14

Определить веб-сервер

С помощью утилиты *telnet* определить, какой веб-сервер установлен на хосте www.rbc.ru.

Задание 15

Определить географическую локализацию хоста

Определить маршрут прохождения ICMP пакетов до хоста www.ttt.com. Определите примерную географическую локализацию хоста.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Изучить лекционный материал и материал, представленный в методических указаниях и в рекомендуемых источниках.

Основная литература

1. Савельев, А.О. HTML5. Основы клиентской разработки / А.О. Савельев, А.А. Алексеев. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 272 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429150> (10.05.2018).

2. Крахоткина, Е.В. Технологии разработки Internet-приложений: учебное пособие / Е.В.

Краюткина; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2016. - 124 с. : ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=459070>

3. Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 265 с.: ил.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428953> (09.11.2017).

4. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с.: ил.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429078> (09.11.2017).

5. Информационные технологии. HTML и XHTML: учебное пособие / А.И. Костюк, С.М. Гушанский, М.Ю. Поленов, Б.В. Катаев; Министерство образования и науки Российской Федерации, Южный федеральный университет, Инженерно-технологическая академия. - Таганрог: Издательство Южного федерального университета, 2015. - 131 с.: ил. - Библиогр. в кн. - ISBN 978-5-9275-1329-1; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=461923> (03.04.2018).

6. Лыткина, Е.А. Основы языка HTML : учебное пособие / Е.А. Лыткина, А.Г. Глотова ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования Северный (Арктический) федеральный университет им. М.В. Ломоносова. - Архангельск : САФУ, 2014. - 104 с. : табл., ил. - Библиогр. в кн. - ISBN 978-5-261-01010-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=436328> (10.05.2018).

Дополнительная литература

7. Информационные технологии: лабораторный практикум / Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет»; авт.-сост. С.В. Говорова, М.А. Лапина. - Ставрополь: СКФУ, 2016. - 168 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=459048> (09.11.2017).

8. Информационные Web-технологии / Ю. Громов, О.Г. Иванова, Н.Г. Шахов, В.Г. Однолько ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2014. - 96 с. : ил. - Библиогр. в кн. - ISBN 978-5-8265-1365-1 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=277935> (10.05.2018).

9. Малашкевич, В.Б. Интернет-программирование : лабораторный практикум / В.Б. Малашкевич ; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с. : ил. - Библиогр.: с. 82. - ISBN 978-5-8158-1854-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=476400> (10.05.2018).

10. Технологии защиты информации в компьютерных сетях / Н.А. Руденков, А.В. Пролетарский, Е.В. Смирнова, А.М. Суоров. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 369 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428820> (10.05.2018).

11. Применение каскадных таблиц стилей (CSS) : курс / П. Храмцов, С. Брик, А. Русак, А. Сурин. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 82 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 978-5-94774-648-8 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429258> (10.05.2018).

Контрольные вопросы для самопроверки

1. Какую роль играют протоколы в сети Интернет?
2. Каким образом организована передача данных в Интернет?

3. За счет чего в сети Интернет достигается надежность передачи данных?
4. Назовите основные организации, курирующие развитие архитектуры и протоколов Интернет.
5. Где регистрируются доменные имена в Российском сегменте Интернет?
6. Кто управляет Интернет и в каком смысле?
7. Кто регистрирует доменные имена в Интернет?
8. Что такое RFC?
9. Какие задачи решает консорциум W3C?
10. Что такое IP-адрес?
11. Что такое доменное имя?
12. Опишите структуру доменных имен.
13. Что такое DNS? Опишите работу DNS-сервера.
14. Что такое прокси-сервер?
15. Какие задачи решает прокси-сервер?
16. Назовите основные протоколы Интернет и их назначение.
17. Какую принципиальную проблему решило создание FTP?
18. Какую принципиальную проблему решило создание Telnet?
19. Для чего используется программа Telnet?
20. Какие вы знаете почтовые протоколы в Интернет?

Лабораторная работа № 2.

Создание Web-сайта

Цель работы:

- 1) освоить технологию HTML для разработки веб-сайта;
- 2) освоить технологию CSS для разработки веб-сайта, научиться выбирать оптимальный вариант действия кода CSS на HTML версию сайта;
- 3) освоить технологию JS для разработки веб-сайта, научиться выбирать оптимальный вариант действия кода JS на HTML версию сайта;
- 4) освоить приемы работ по созданию веб-сайта на примере разработки сайта «Компьютер».

Задание:

1. Изучить теоретические сведения

Порядок выполнения:

- 1) прочитать лекцию, теоретические сведения и рассмотреть приведенные в лекции образцы решения заданий;
- 2) выполнять задания лабораторного занятия аналогично.

Форма отчетности:

Подготовить отчет, содержащий следующие пункты:

- Титульный лист;
- Содержание;
- Краткая теория;
- Результаты работы (скриншоты выполненных заданий);
- Выводы.

Выполнить лабораторное задание и сохранить созданные файлы в папке с именем ЛР Веб фамилия группа.

Задания для самостоятельной работы:

Задание 1

Создать структуру Web-страницы

Для создания титульной Web-страницы сайта используем текстовый редактор Блокнот. Открыть окно текстового редактора Блокнот. Вид Web-страницы задается тэгами, которые заключаются в угловые скобки. Тэги могут быть одиночными или парными, для которых обязательно наличие открывающего и закрывающего тэгов (такая пара тэгов называется контейнером). Закрывающий тэг содержит прямой слэш (/) перед обозначением. Тэги могут записываться как прописными, так и строчными буквами. HTML-код страницы помещается внутрь контейнера `<HTML></HTML>`. Без этих тэгов браузер не в состоянии определить формат документа и правильно его интерпретировать. Web-страница разделяется на две логические части: заголовок и содержание.

Заголовок Web-страницы заключается в контейнер `<HEAD></HEAD>` и содержит название документа и справочную информацию о странице (например, тип кодировки), которая используется браузером для ее правильного отображения. Название Web-страницы содержится в контейнере `<TITLE></TITLE>` и отображается при ее просмотре в строке заголовка браузера. Назовем Web-страницу «Компьютер»:

```
<HEAD>
<TITLE>Компьютер</TITLE>
</HEAD>
```

Основное содержание страницы помещается в контейнер `<BODY></BODY>` и может включать текст, таблицы, бегущие строки, ссылки на графические изображения и звуковые файлы и т.д. Поместим для начала на страницу текст «Все о компьютере»: `<BODY>Все о компьютере</BODY>`.

Созданную Web-страницу необходимо сохранить в виде файла. Принято называть титульный файл сайта, т.е. тот который первый загружается в браузер, именем `index.htm`. В качестве расширения файла Web-страницы можно также использовать `html`. Рекомендуется создать для размещения сайта специальную папку и сохранять все файлы разрабатываемого сайта в этой папке.

Необходимо различать имя файла `index.htm`, т.е. имя под которым Web-страница хранится в файловой системе, и собственно имя Web-страницы «Компьютер», которая высвечивается в верхней строке окна браузера и, в первую очередь анализируется поисковыми системами. Имя Web-страниц должно в максимальной степени соответствовать ее содержанию. В окне приложения Блокнот ввести HTML-код Web-страницы. Сохранить файл под именем `index.htm` в папку сайта. Загрузить этот файл в окно браузера для просмотра.

```
<HTML> <HEAD> <TITLE>Компьютер</TITLE></HEAD><BODY>Все о компьютере
</BODY> </HTML>
```

Задание 2

Форматировать текст

Пока страница выглядит не слишком привлекательно. Мелкий шрифт, черный текст на белом фоне почти не обращают на себя внимание. С помощью HTML-тэгов можно задать различные параметры форматирования текста. Размер шрифта для имеющихся в тексте заголовков задается тэгами от `<H1>` (самый крупный) до `<H6>` (самый мелкий). Заголовок страницы целесообразно выделить самым крупным шрифтом:

```
<H1>Все о компьютере</H1>
```

Некоторые тэги имеют атрибуты, которые могут принимать определенные значения. Так, заголовок по умолчанию выровнен по левому краю страницы, однако заголовок принято размещать по центру. Задать тип выравнивания заголовка для тэга заголовка позволяет атрибут `ALIGN`, которому требуется присвоить определенное значение. Выравнивание по правой границе окна реализуется с помощью `ALIGN="right"`, а по центру — `ALIGN="center"`.

```
<H1 ALIGN="center">Все о компьютере</H1>
```


С помощью тэга FONT и его атрибутов можно задать параметры форматирования шрифта любого фрагмента текста. Атрибут FACE позволяет задать гарнитуру шрифта (например, FACE="Arial"), атрибут SIZE—размер шрифта (например, SIZE=4), атрибут COLOR—цвет шрифта (например, COLOR="blue"). Значение атрибута COLOR можно задать либо названием цвета (например, "red", "green", "blue" и т.д.), либо его шестнадцатеричным значением. Шестнадцатеричное представление цвета использует RGB формат "RRGGBB", где две первые шестнадцатеричные цифры задают интенсивность красного (red), две следующие — интенсивность зеленого (green) и две последние — интенсивность синего (blue) цветов. Минимальная интенсивность цвета задается шестнадцатеричным числом 00, а максимальная —FF. Легко догадаться, что синему цвету будет соответствовать значение "#0000FF".

Таким образом, задать синий цвет заголовка можно с помощью тэга FONT с атрибутом COLOR:

```
FONT COLOR="blue">
<H1 ALIGN="center">Все о компьютере</H1>
</FONT>
```

Заголовок целесообразно отделить от остального содержания страницы горизонтальной линией с помощью одиночного тэга <HR>.

Разделение текста на абзацы производится с помощью контейнера <P></P>. При просмотре в браузере абзацы отделяются друг от друга интервалами. Для каждого абзаца можно задать определенный тип выравнивания. На титульной странице обычно размещается текст, кратко описывающий содержание сайта. Поместим на титульную страницу текст, разбитый на абзацы с различным выравниванием:

```
<P ALIGN="left">
```

На этом сайте вы сможете получить различную информацию о компьютере, его программном обеспечении и ценах на компьютерные комплектующие

```
</P>
```

```
<P ALIGN="right">
```

Терминологический словарь познакомит вас с компьютерными терминами, а также вы сможете заполнить анкету

```
</P>
```

Таким образом, если вставить в страницу определенную последовательность тэгов, то мы получим Web-страницу, содержащую отцентрированный заголовок синего цвета, выводимый крупным шрифтом и отделенный горизонтальной линией от остального текста.

В окне приложения Блокнот в контейнер <BODY> вставьте последовательность тэгов и просмотрите результат в браузере.

```
<FONT COLOR="blue">
```

```
<H1 ALIGN="center">
```

```
Все о компьютере</H1>
```

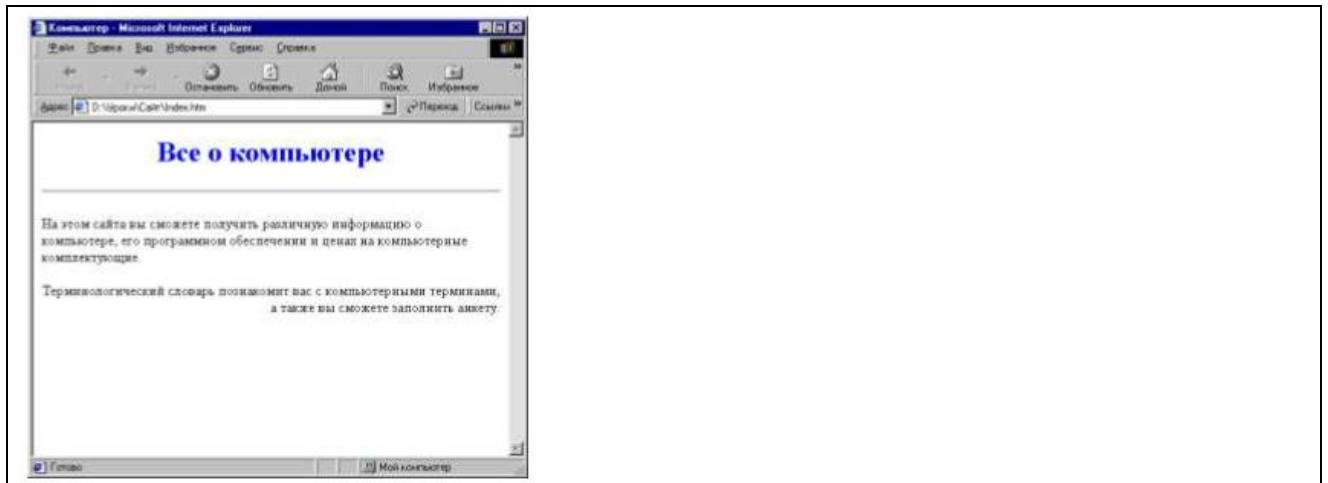
```
</FONT>
```

```
<HR>
```

```
<P ALIGN="left">На этом сайта вы сможете получить различную информацию о компьютере, его программном обеспечении и ценах на компьютерные комплектующие. </P>
```

```
<P ALIGN="right">
```

```
Терминологический словарь познакомит вас с компьютерными терминами, а также вы сможете заполнить анкету.</P>
```



Задание 3 Вставить изображения

На Web-страницы обычно помещают изображения, чтобы сделать их визуально более привлекательными. На Web-страницах могут размещаться графические файлы трех форматов — GIF, JPG и BMP. Если рисунок сохранен в другом формате, то его необходимо предварительно преобразовать в один из выше указанных форматов с помощью графического редактора. Для этих целей можно использовать редактор PhotoEditor, который входит в пакет MicrosoftOffice.

На титульной странице создаваемого сайта уместно разместить изображение того объекта, которому посвящен сайт (в нашем случае — компьютера). Многочисленные фотографии компьютеров можно найти, например, на сайтах производителей и продавцов компьютерной техники.

Для вставки изображения используется тэг с атрибутом SRC, который указывает на место хранения файла на локальном компьютере или в Интернете. Если графический файл находится на локальном компьютере в том же каталоге, что и файл Web-страницы, то в качестве значения атрибута SRC достаточно указать только имя файла:

```
<IMG SRC="computer.gif">
```

Если файл находится в другом каталоге на данном локальном компьютере, то значением атрибута должно быть полное имя файла, включая путь к нему в иерархической файловой системе. Например:

```
<IMG SRC="C:\computer\computer.gif">
```

Если файл находится на удаленном сервере в Интернете, то должен быть указан URL-адрес этого файла. Например:

```
<IMG SRC="http://www.server.ru/computer.gif">
```

Иллюстрации на Web-страницах стали неотъемлемым элементом дизайна. Однако пользователи иногда, в целях экономии времени, отключают в браузере загрузку графических изображений и читают только тексты. Поэтому, чтобы не терялся смысл и функциональность страницы, вместо рисунка должен выводиться поясняющий текст.

Поясняющий текст выводится с помощью атрибута ALT, значением которого является текст, поясняющий, что должен был бы увидеть пользователь на рисунке:

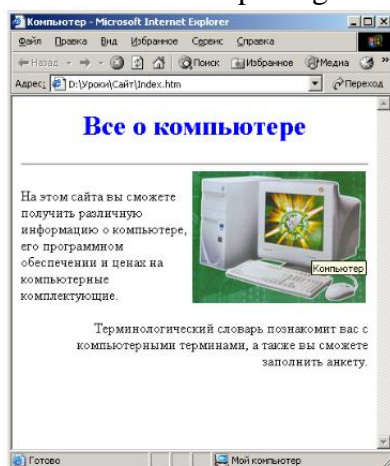
```
<IMG SRC="computer.gif" ALT="Компьютер">
```

Расположить рисунок относительно текста различным образом позволяет атрибут ALIGN тэга , который может принимать пять различных значений:

- TOP (верх),
- MIDDLE (середина),
- BOTTON (низ),
- LEFT (слева),
- RIGHT(справа).

Для того чтобы рисунок располагался по правому краю текста, тэг вставки изображения должен принять следующий вид:

В окне приложения Блокнот в контейнер <BODY> вставьте перед абзацем и текста тэга вставки изображения, просмотрите результат в браузере.



Задание 4

Вставить гиперссылки

Первая титульная страница должна обеспечивать посетителю Web-сайта возможность начать «путешествие» по сайту. Для этого на титульную страницу должны быть помещены гиперссылки на другие страницы сайта.

Для создания гиперсвязей между титульной страницей и другими страницами сайта необходимо, прежде всего, создать заготовки Web-страниц. Такие «пустые» страницы должны иметь заголовок, но могут пока не иметь содержания. Все создаваемые страницы необходимо сохранить в виде файлов в папке сайта.

Каждая страница будет содержать следующий HTML-код:

```
<HTML>
<HEAD>
<TITLE>Заголовок страницы</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Создать пустые страницы «Программы», «Словарь», «Комплекующие» и «Анкета».

Сохранить их в файлах с именами software.htm, glossary.htm, hardware.htm и anketa.htm в каталоге сайта.

Гиперссылка состоит из двух частей: указателя ссылки и адресной части ссылки.

Указатель ссылки — это то, что мы видим на Web-странице (текст или рисунок), обычно выделенный синим цветом и подчеркиванием. Активизация гиперссылки вызывает переход на другую страницу.

Адресная часть гиперссылки представляет собой URL-адрес документа, на который указывает ссылка. URL-адрес может быть абсолютным и относительным. Абсолютный URL-адрес документа полностью определяет компьютер, каталог и файл, в котором располагается документ.

Абсолютный адрес документа, находящегося на локальном компьютере, будет включать в себя путь к файлу и имя файла, например:

C:/Web-сайт/filename.htm

Абсолютный адрес документа, находящегося на удаленном компьютере в Интернете, будет включать имя сервера Интернета, путь к файлу и имя файла, например:

http://www.host.ru/Web-caйм/filename.htm

Относительный URL-адрес указывает на местоположение документа относительно того, в котором находится указатель ссылки. При разработке сайта рекомендуется входящие в

него Web-страницы связывать относительными ссылками. Это позволит не изменять адресную часть ссылок при перемещении Web-сайта в другой каталог локального компьютера или при его публикации в Интернете.

Гиперссылка на Web-странице существует в форме указателя ссылки, щелчок по которому обеспечивает переход на Web-страницу, указанную в адресной части ссылки.

Прежде всего, необходимо разместить на титульной странице тексты гиперссылок на каждую страницу сайта. Для представления гиперссылок удобнее всего выбрать названия страниц, на которые осуществляется переход.

Разместим гиперссылки под введенным текстом, в новом абзаце, в одну строку, разделенные несколькими пробелами. Такое размещение гиперссылок часто называют панелью навигации.

Панель навигации будет представлять собой абзац, выровненный по центру, в котором указатели гиперссылок разделены пробелами ():

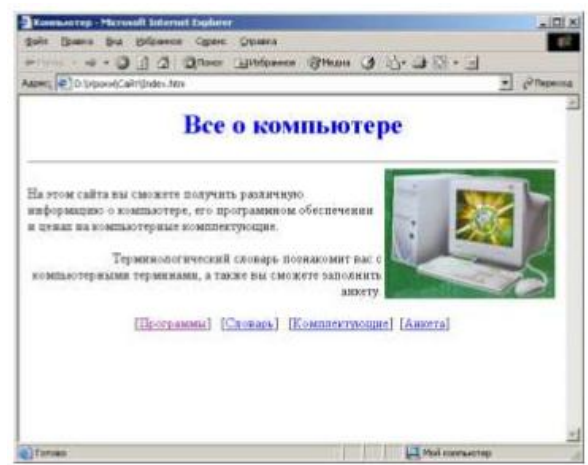
```
<P ALIGN="center"> [Программы] &nbsp;  [Словарь] &nbsp;  [Комплекующие] &nbsp;  [Анкета] </P>
```

Теперь для каждой гиперссылки определим адрес перехода. Для этого используется контейнер гиперссылки <A> с атрибутом HREF, значением которого является URL – адрес документа на локальном компьютере или в Интернете. Контейнер должен содержать указатель гиперссылки:

```
<A HREF="URL">Указатель гиперссылки</A>
```

Вставьте в титульную страницу код, создающий панель навигации:

```
<P ALIGN="center">[<A HREF="software.htm">Программы</A>] &nbsp;  [<A HREF="glossary.htm">Словарь</A>] &nbsp;  [<A HREF="hardware.htm">Комплекующие</A>]&nbsp;  [<A HREF="anketa.htm">Анкета</A>]</P>
```



Панель навигации на титульной странице создана, теперь активизация текста гиперссылок будет приводить к переходу на другие страницы сайта.

Задание 5

Создать списки на Web-странице

Часто при размещении текста на Web-страницах удобно использовать списки, в различных вариантах:

- нумерованные списки, когда элементы списка идентифицируются с помощью чисел;
- маркированные списки (в HTML их принято называть нумерованными), когда элементы списка идентифицируются с помощью специальных символов (маркеров);
- списки определений, позволяющие составлять перечни определений в так называемой словарной форме.

Возможно создание и вложенных списков, причем вкладываемый список может по своему типу отличаться от основного.

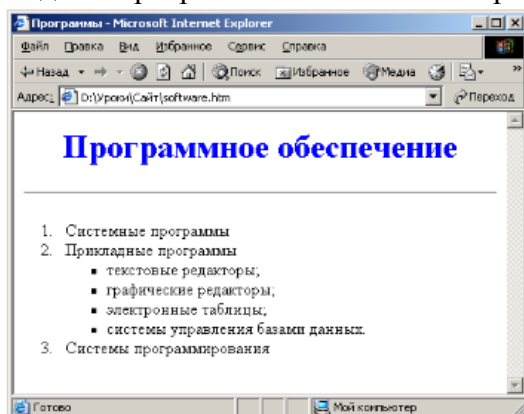
На странице «Программы» разместим информацию об основных типах программного обеспечения компьютера в форме вложенного списка.

Сначала создадим основной нумерованный список основных категорий программного обеспечения. Список располагается внутри контейнера

``, а каждый элемент списка определяется тэгом ``.

С помощью атрибута `TYPE` тэга `` можно задать тип нумерации: арабские цифры (по умолчанию), " I " (римские цифры), "a" (прописные буквы) и др.

Откройте в Блокноте файл `software.htm`, введите заголовок Программное обеспечение и добавьте следующий HTML-код, задающий список: ` Системные программы Прикладные программы Системы программирования Программирования `



Создадим вложенный нумерованный список, для одного из элементов основного списка. Список располагается внутри контейнера ``, а каждый элемент списка определяется также тэгом ``. С помощью атрибута `TYPE` тэга `` можно задать вид маркера списка: "disc" (диск), "square" (квадрат) или "circle" (окружность).

Добавьте HTML-код, задающий вложенный список для элемента `` Прикладные программы:

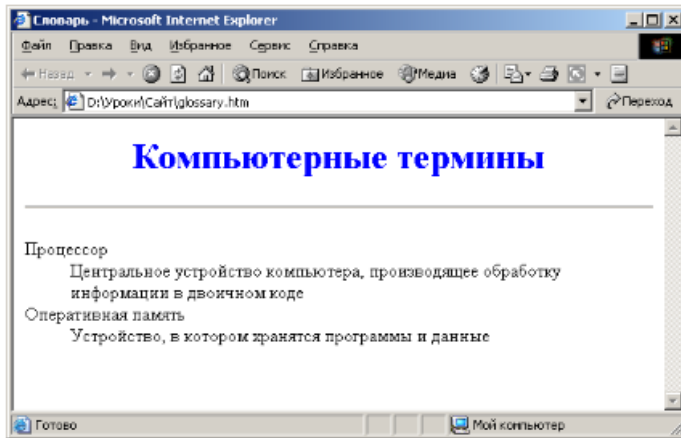
```
<OL>
<LI>СИСТЕМНЫЕ ПРОГРАММЫ
<LI>ПРИКЛАДНЫЕ ПРОГРАММЫ
<UL TYPE="SQUARE">
<LI>ТЕКСТОВЫЕ РЕДАКТОРЫ;
<LI>ГРАФИЧЕСКИЕ РЕДАКТОРЫ;
<LI>ЭЛЕКТРОННЫЕ ТАБЛИЦЫ;
<LI>СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ.
</UL>
<LI>СИСТЕМЫ ПРОГРАММИРОВАНИЯ
</OL>
```

Страницу «словарь» мы представим в виде словаря компьютерных терминов.

Для этого используем контейнер списка определений `<DL></DL>`. Внутри него текст оформляется в виде термина, который выделяется непарным тэгом `<DT>` и определения, которое следует за тэгом `<DD>`.

Откройте в Блокноте файл `glossary.htm`, введите заголовок **Компьютерные термины** и добавьте следующий HTML-код, задающий список определений:

```
<DL>
<DT>Процессор
<DD>Центральное устройство компьютера, производящее обработку информации в двоичном коде
<DT>Оперативная память
<DD>Устройство, в котором хранятся программы и данные
</DL>
```



Задание 6

Обработать события с использованием JavaScript

Написанный на JavaScript сценарий может реагировать на действия пользователя и другие внешние события. Каждое из событий связано с тем или иным объектом: формой, гипертекстовой ссылкой или даже с окном, содержащим текущий документ. В качестве примеров внешних событий, на которые могут реагировать объекты JavaScript, можно привести следующие.

- окончание загрузки документа в окно (или окончание загрузки документов во все фреймы окна). Это событие связано с объектом *window*;
- щелчок мышью на объекте. Это событие может быть связано с интерактивным элементом формы или с гипертекстовой ссылкой;
- получение объектом фокуса ввода. Это событие может быть связано с объектами типа *Text*, *Password* и с другими интерактивными элементами;
- передача на сервер данных, введенных пользователем с помощью интерактивных элементов. Связывается с формой.

Обработка события производится с помощью специально предназначенного для этого фрагмента кода, называемого *обработчиком события*. Для каждого события JavaScript предоставляет свой обработчик. Однако при построении сценария можно создавать собственный обработчик события и использовать его вместо обработчика, заданного по умолчанию.

Имя обработчика определяет, какое событие он должен обрабатывать. Так, для того чтобы сценарий нужным образом отреагировал на щелчок мышью, используется обработчик с именем *onClick*, для обработки события, заключающегося в получении фокуса ввода, - обработчик *onFocus*.

Для того чтобы указать интерпретатору JavaScript на то, что обработкой события должен заниматься обработчик, необходимо включить в HTML-дескриптор следующее выражение: *имя_обработчика="команды_обработчика"*

Это выражение включается в тэг, описывающий объект, с которым связано событие.

Например, если необходимо обработать событие, заключающееся в получении фокуса полем ввода, дескриптор, описывающий этот интерактивный элемент, должен иметь примерно следующий вид: `<input type="text" name="Inform" onFocus="handleFocus();">`

Имя обработчика является одним из атрибутов HTML-дескриптора, а команды, предназначенные для обработки события, выступают в роли значения этого атрибута. В данном случае обработка события производится в теле функции *handleFocus()*. В принципе, обработчиком может быть не только функция, но и любая последовательность команд JavaScript в виде составного оператора.

Следующий пример демонстрирует обработку события, связанного с наведением курсора мыши на гиперссылку:

```
<a href = "http://www.myhp.edu" onmouseover="alert('An onMouseOver event'); return false">
```

```

</a>
```

Ниже приводится полный текст HTML документа с JavaScript сценарием, в котором обрабатывается событие нажатия кнопки мыши, и определяется, какая именно из них была нажата:

Пример 1.

```
<html>
  <head>
    <script language = "javascript">
      function whichButton(event)
      {
        if (event.button == 2)
          {
            alert("Вы щелкнули правой кнопкой мыши!");
          }
        else
          {
            alert("Вы щелкнули левой кнопкой мыши!");
          }
      }
    </script>
  </head>

  <body onmousedown="whichButton(event)">
    <p>Щелкните любой кнопкой мыши в любом месте документа</p>
  </body>
</html>
```

Для того чтобы обработать какое-либо стандартное событие в браузере, необходимо добавить в подходящий HTML тэг атрибут, соответствующий этому событию, указав в качестве значения атрибута имя JavaScript функции. Список атрибутов, которые определены для HTML тэгов приводится ниже:

Атрибут	Описание	Номер версии браузера			W3C
		IE	F	O	
onabort	Прерванная загрузка изображения	4	1	9	Да
onblur	утрата фокуса элементом	3	1	9	Да
onchange	Изменение содержимого в поле ввода	3	1	9	Да
onclick	Щелчок мыши на объекте	3	1	9	Да
ondblclick	Двойной щелчок мыши на объекте	4	1	9	Да
onerror	Ошибка при загрузке изображения или документа	4	1	9	Да
onfocus	Получение фокуса элементом	3	1	9	Да
onkeydown	Нажатие клавиши	3	1	Нет	Да
onkeypress	Клавиша нажата	3	1	9	Да
onkeyup	Отжатие клавиши	3	1	9	Да
onload	Завершение загрузки страницы или изображения	3	1	9	Да
onmousedown	Нажатие кнопки мыши	4	1	9	Да

onmousemove	Перемещение курсора мыши	3	1	9	Да
onmouseout	Смещение курсора мыши с объекта	4	1	9	Да
onmouseover	Наведение курсора мыши на объект	3	1	9	Да
onmouseup	Отжатие кнопки мыши	4	1	9	Да
onreset	Кнопка "Reset" нажата	4	1	9	Да
onresize	Изменение размера окна	4	1	9	Да
onselect	Выделение текста	3	1	9	Да
onsubmit	Кнопка "Submit" нажата	3	1	9	Да
onunload	Уход с веб-страницы	3	1	9	Да
IE: Internet Explorer, F: Firefox, O: Opera, W3C: стандарт					

Задание 7

Использовать регулярные выражения для поиска

Регулярные выражения — система поиска текстовых фрагментов в электронных документах, основанная на специальной системе записи образцов для поиска.

Образец, задающий правило поиска, называется «шаблоном». Применение регулярных выражений принципиально преобразило технологии электронной обработки текстов.

С помощью регулярных выражений можно задавать структуру искомого шаблона и его позицию внутри строки (например, в начале или в конце строки, на границе или не на границе слова).

При описании структуры шаблона используются:

- гибкая система квантификаторов (операторов повторения);
- операторы описания наборов символов и их типа (числовые, нечисловые, специальные).

Для того, чтобы задать положение искомого фрагмента внутри строки, можно использовать один из следующих операторов:

Представление	Позиция
^	Начало строки
\$	Конец строки
\b	Граница слова
\B	Не граница слова
(?=шаблон)	Искомая строка следует после указанной строкой (с просмотром вперед)
(?!шаблон)	Искомая строка не следует после указанной строки (с просмотром вперед)
(?<шаблон)	Искомая строка следует после указанной строкой (с просмотром назад)
(?<!шаблон)	Искомая строка не следует после указанной строки (с просмотром назад)

Язык регулярных выражений предоставляет набор квантификаторов, позволяющих указать число повторений шаблона:

Представление	Число повторений
{n}	Ровно n
{m, n}	От m до n включительно
{m,}	Не менее m
{,n}	Не более n

Имеются и более простые квантификаторы:

Представление	Число повторений	Эквивалент
*	Ноль или более	{0,}

+	Одно или более	{1,}
?	Ноль или одно	{0,1}

Для задания внутри шаблона группы символом можно использовать следующие операторы:

Оператор	Описание
[xyz]	Любой символ из указанного множества
[^xyz]	Любой символ не входящий в указанное множество
[x-z]	Любой символ из указанного диапазона
[^x-z]	Любой символ не входящий в указанный диапазон
. (точка)	Любой символ кроме символов разрыва или переноса строки
\w	Любой буквенно-цифровой символ, включая символ подчеркивания
\W	Любой не буквенный символ
\d	Любая цифра
\D	Любой нецифровой символ
\s	Любой не отображаемый символ
\S	Любой символ (кроме неотображаемых символов)

Для группировки отдельных частей шаблона можно использовать следующие операторы:

Оператор	Описание
()	Поиск группы символов внутри скобок и сохранение найденного соответствия
(?:)	Поиск группы символов внутри скобок без сохранения найденного соответствия
	Комбинирование частей в одно выражения с последующим поиском любой из частей в отдельности. Аналогично оператору «ИЛИ».

Если шаблон поиска включает специальные (как правило неотображаемые) символы, для их описания можно использовать следующие обозначения:

Обозначение	Описание
\0	Символ с нулевым кодом
\n	Символ новой строки
\r	Символ начала строки
\t	Символ табуляции
\v	Символ вертикальной табуляции
\xxx	Символ, имеющий заданный восьмеричный ASCII код xxx
\xdd	Символ, имеющий заданный шестнадцатиричный ASCII код dd
\uxxxx	Символ, имеющий ASCII код выраженный ЮНИКОДОМ xxxx

Квантификаторам в регулярных выражениях соответствует максимально длинная строка из возможных (т.е. квантификаторы являются «жадными»). Это может приводить к некоторым проблемам. Например, шаблон (<.*>) описывающий на первый взгляд теги HTML на самом деле будет выделять более крупные фрагменты в документе.

Например, строка вида

<p><i>Регулярные выражения</i> - удобный инструмент для поиска в строках </p>

формально соответствует указанному выше шаблону

Для решения данной проблемы можно использовать два подхода.

- В регулярном выражении учитываются символы, не соответствующие желаемому образцу (например, <[^>]*> для вышеописанного случая).

- Определение квантификатора как *нежадного* (ленивого) - большинство реализаций позволяют это сделать, добавив после него знак вопроса.

Например, по шаблону (<.*?>) будут найдены все теги из рассмотренной строки.

Таким образом, получаются следующие «нежадные» модификации квантификаторов:

Квантификатор	Описание
*?	«не жадный» эквивалент *
+?	«не жадный» эквивалент +
{n,}?	«не жадный» эквивалент {n,}

Следует, однако, иметь в виду, что использование «ленивых» квантификаторов может привести к ситуации, когда выражению соответствует слишком короткая, в частности, пустая строка.

Задание 8

Использовать регулярные выражения в JavaScript

При поиске по тексту можно использовать шаблон, описывающий подстроку. В JavaScript такой шаблон может быть описан с помощью объекта *RegExp*. В простейшем случае такой шаблон описывает отдельный символ, однако имеет смысл его использовать для регулярных выражений.

Следующий ниже код описывает *RegExp* объект с именем *pattn*, содержащий регулярное выражение, описывающее целое десятичное число:

```
var pattn = new RegExp("/[0-9]+/");
```

Объект *RegExp* имеет 3 встроенных метода: *test()*, *exec()* и *compile()*. Метод *test()* выполняет поиск по шаблону:

```
var pattn = new RegExp("[0-9]+");  
document.write(pattn.test("38 попугаев"));
```

Результат:

```
true
```

Метод *exec()* выполняет поиск подстроки по шаблону и возвращает найденные соответствия; если соответствий нет, возвращается значение *null*:

```
var pattn=new RegExp("[0-9]+");  
document.write(pattn.exec("38 попугаев"));
```

Результат:

```
38
```

Если необходимо найти все соответствия, то при вызове конструктора *RegExp* следует указать дополнительный параметр "g", указывающий на необходимость глобального поиска.

Пример 2

```
var pattn = new RegExp("[0-9]+", "g");  
do  
{  
    result = pattn.exec("1 попугай, 2 попугая,..., 38 попугаев");  
    document.write(" " + result);  
}  
while (result != null)
```

Результат:

```
1 2 38 null
```

Метод *compile()* применяется для изменения ранее созданного шаблона.

Пример 3

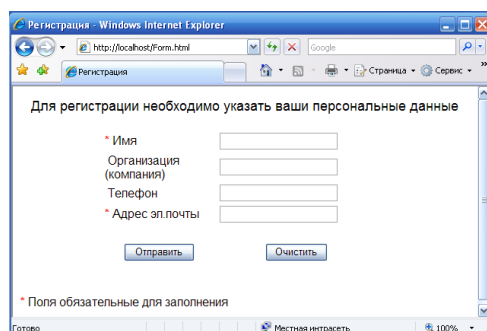
```
var pattn = new RegExp("[0-5]+");  
document.write(pattn.exec("38 попугаев"));  
  
pattn.compile("[6-9]+");  
document.write("; " + pattn.exec("38 попугаев"));
```

Результат:

```
3;8
```

Проверка значений, введенных пользователем в поля формы для регистрации.

1) необходимо создать веб-страницу, содержащую форму с полями, следующего вида:



2) В тэге `<form>` добавьте обработчик события отправки данных вида:

```
onSubmit = "CheckData(); return false;"
```

В данном случае указана функция обработчик `CheckData()`. Оператор `return false;` предотвращает автоматическую отправку данных после выполнения функции-обработчика. Отправка данных будет выполняться из обработчика.

3) Добавьте на странице секцию JavaScript кода, описывающего функцию-обработчик:

```
function CheckData()
{
    var ans;

    ans = confirm("Вы уверены, что хотите отправить введенные данные ?");
    if (ans) submit();
}
```

Функция `CheckData()` в случае подтверждения со стороны пользователя самостоятельно вызывает метод `submit()` для передачи данных из формы.

4) Теперь необходимо добавить проверку значений, введенных в поля формы пользователем.

Прежде всего, необходимо убедиться в том, что заполнены все поля, обязательные для ввода. Для этого можно использовать проверку на равенство нулю длины строки (свойство `length`), являющейся значением узла дерева документа, соответствующего полю ввода, например: `document.getElementById("uname").value.length`. Следующая проверка должна контролировать структуру и содержимое полей. Для этого можно использовать объект `RegExp`, например:

```
var validEMail, pattn;

pattn = new RegExp("^[\.\_\-A-Za-z0-9]+?@[.\_\-A-Za-z0-9]+?\.?[A-Za-z0-9]{2,6}$");

validEMail = pattn.test(document.getElementById("email").value));
```

В данном фрагменте описана проверка структуры электронного адреса из поля формы с идентификатором "email". Для проверки был использован шаблон на основе регулярного выражения.

Задание 9

Создать структуру веб-документа.

В рабочей папке создайте папку «My_site». Внутри нее создайте папку «images», где будут храниться графические материалы Вашего сайта. Скопируйте в нее содержимое папки «Картинки для сайта». Откройте текстовый редактор «Bred3» и введите следующие теги: `<html> <head> </head> <body> </body> </html>`

Выберите «Файл» > «Сохранить» > «Ваша рабочая папка/my_site/index.html» (указывать расширение обязательно). Не закрывая «Блокнот» откройте файл `index.html` еще раз из папки. Он будет открыт браузером, установленным по умолчанию. Файл открыт

одновременно «Блокнотом» и браузером. Перейдите в «Блокнот» и между парными тегами `<head>` и `</head>` добавьте строку:

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

Эта строка служебной информации не видна, как и все прочие на странице сайта, но позволит правильно отображать информацию на любом ПК, в какой стране бы он не находился, независимо от настроек. Любой пользователь увидит текст, написанный на кириллице (`charset=windows-1251`). Исключением является тег `<title>`, содержание которого отображается в браузере как название вашей страницы (окна документа). Ниже строки с кодировкой введите строку: `<title>7 чудес Беларуси</title>`

Задание 10

Оформить фон веб-страницы.

Установка в качестве фона цвета. В качестве параметра тега `<body>` установите `bgcolor="blue"`. Т.е. у Вас должна получиться следующая строка: `<body bgcolor="blue">`

Перейдите в «Блокнот» и измените "blue" на "#f05fbb"

Установка в качестве фона рисунка Фон одно большое изображение. В этом случае страничка будет грузиться дольше. В качестве параметра тега `<body>` установите `background="images/fon1.jpg"`.

Должна получиться следующая строка: `<body background="images/fon1.jpg">`

Сохраните документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить» Фон фрагмент изображения. Фрагмент автоматически размножается браузером, поэтому его края должны хорошо стыковаться. Сейчас используем в качестве фона следующий фрагмент: В качестве параметра тега `<body>` установите `background="images/fon2.jpg"`.

Должна получиться следующая строка: `<body background="images/fon2.jpg">`

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить». Самостоятельно: установите фон сайта на Ваш вкус (цвет или рисунок). Обозначения цветов можно узнать из таблицы.

Задание 11

Оформить верхнюю часть веб-страницы

Все теги будем размещать между парными тегами `<body>` и `</body>`. Как правило, в верхней части веб-страницы размещается заголовок сайта в виде текста или логотипа (картинки). Мы же используем готовый логотип предварительно созданный в графическом редакторе. После тега `<body>` введите следующий участок кода:

```

```

Для того, чтобы разместить логотип по центру страницы, заключите введенную Вами строку между тегами `<center>` и `</center>`. Теперь отделим заголовок от остального содержимого страницы горизонтальной линией. В следующей строке введите текст:

```
<hr width="80%" size="5" color="white">
```

Самостоятельно: установите цвет, ширину и размер линии.

Задание 12

Оформить основную часть веб-страницы

Вывод данных о сайте. В качестве пропедевтической части отобразим на странице следующий текст (просто наберите следующей строкой его после тега `<hr.>`): На нашей планете есть множество естественных и искусственных объектов, которые поражают наше воображение. Это пирамиды в Египте и Мексике, истуканы на острове Пасхи, дворец Тадж-Махал в Индии, Стоунхендж в Великобритании, Ниагарский водопад в Канаде, Большой каньон в Колорадо и многие другие. Их называют «Чудеса света». Ежегодно сотни тысяч туристов приезжают посмотреть на эти диковины, принося ощутимый доход государствам, на территории которых они расположены. Теперь будем этот текст форматировать.

Прежде всего, отметим его как абзац. Для этого заключим текст между тегами <p> и </p>. Установим выравнивание при помощи параметра align="justify" (выравнивание по ширине). Должно получиться следующее: <p align="justify">на нашей расположены.</p> .

Самостоятельно: измените значение параметра align на "left" затем на "right" и посмотрите, как меняется позиция текста на странице. Установим теперь атрибуты текста (цвет, размер, гарнитуру). Для этого воспользуемся тегом . Заключите текст между тегами и .

Должно получиться следующее:

<p align="justify">на нашей расположены.</p>

Самостоятельно: измените значение параметров face, size и color по Вашему вкусу. Для акцентирования внимания на отдельных словах используем парные теги и (жирный), <i> и </i> (курсив), <u> и </u> (подчеркнутый). Заключите между этими все названия объектов, например: дворец тадж-махал в Индии, или <u>большой каньон</u> в Колорадо или <i>пирамиды</i> в Египте

Добавим в текст рисунок-коллаж. Между тегами <p> и вставьте

Самостоятельно: изменяя значения параметров align, width, height, hspace и vspace добейтесь оптимального взаимного расположения текста и рисунка.

Добавим следующий абзац: Однако не обязательно ехать в далекие страны, чтобы увидеть удивительные природные ландшафты, памятники старины и произведения зодчества. В нашей стране тоже есть уникальные объекты, включенные ЮНЕСКО в список Всемирного наследия человечества. Представляем Вам СЕМЬ ЧУДЕС БЕЛАРУСИ.

Самостоятельно: установите необходимые параметры абзаца и шрифта (по аналогии с предыдущим). Создадим маркированный список чудес:

```
<ul type="circle"> <li>мирский замок</li> <li> Софийский собор </li> <li>
Борисоглебская церковь </li> <li> Несвижский замок </li> <li> Брестская крепость </li> <li>
Беловежская пуца </li> <li> Озеро Нарочь </li> </ul>
```

Для того, чтобы список выводился по центру страницы, необходимо создать таблицу 1x1, т.е. добавить следующие теги:

```
<table align="center"> <tr> <td> <ul type="circle"> <li>мирский замок</li>. <li> Озеро
Нарочь </li> </ul> </td>
```

Самостоятельно: используя тег и его параметры face, size и color, установите подходящие размер, цвет и гарнитуру шрифта списка (можно всего списка сразу между тегами и или каждого элемента в отдельности - между тегами и). Как вариант, семь чудес оформить семью цветами радуги. Теперь создадим таблицу, которая будет содержать изображения чудес. Поскольку их семь, то таблица будет иметь 2 строки и 4 столбца.

Введите следующий участок кода:

```
<table border="1" align="center">
<tr align="center" valign="center">
<td>мирский замок
</td>
<td>софийский собор
</td>
<td>борисоглебская церковь
</td>
<td>несвижский замок
</td>
<tr align="center" valign="center">
<td>брестская крепость
</td>
<td>беловежская пуца
</td>
```

```
<td>озеро Нарочь
```

```
</td>
```

```
<td></td>
```

Теперь названия чудес заменим их изображениями: вместо текста «Мирский замок» между тегами `<td>` и `</td>` вставьте участок кода

```

```

Задание 13

Вставить ссылки

Внизу главной страницы нашего сайта расположена таблица, содержащая уменьшенные изображения. Сейчас добавим участок кода, позволяющий при щелчке на фотографии открыть отдельную страницу с ее полноразмерным изображением. Для этого коды в ячейках таблицы заключим в следующие теги `<a>` и ``:

```
<a href="images/mir1.jpg" target="_blank">
```

```

```

```
</a>
```

Аналогично проделайте для каждой ячейки таблицы.

Задание 14

Создать меню

Сайт содержит несколько страниц, поэтому требуется создать меню для перехода между ними. Для этого после строки

```
<hr WIDTH="80%" SIZE="5" color="white">
```

 (вверху сайта горизонтальная линия)

добавьте следующий участок кода:

```
<table border="2" width="80%" align="center" bgcolor="#f5f5f5">
```

```
<tr> <th><a href="index.html">главная</a></th>
```

```
<th><a href="sof.html">софийский собор</a></th>
```

```
<th><a href="nar.html">озеро Нарочь</a>
```

```
</th>
```

```
<th><a href="bor.html">борисоглебская церковь</a>
```

```
</th>
```

```
<th><a href="bre.html">брестская крепость</a>
```

```
</th>
```

```
<th><a href="mir.html">мирский замок</a>
```

```
</th>
```

```
<th><a href="bel.html">беловежская пуца</a>
```

```
</th>
```

```
<th><a href="nes.html">несвижский замок</a>
```

```
</th>
```

Тег `<th>` аналогичен тегу `<td>`, только текст в ячейке создается полужирный с выравниванием по центру.

Задание 15

Создать страницы сайта

Создадим шаблоны страниц, которые будут содержать информацию о каждом «Чуде Беларуси» в отдельности. Выберите меню «Файл» - «Сохранить как» и сохраните страницу в ту же папку, в которой содержится `index.html`, под именем `sof.html` (эта страница будет содержать информацию о Софийском соборе). Удалите ненужные строки кода так, чтобы осталось только:

```
<html>
```

```
<head>
```

```

<title>7 чудес Беларуси</title>
</head>
<body background="images/fon2.jpg" >
  <center></center> <hr width="80%" size="5" color="white">
<table border="2" width="80%" align="center" bgcolor="#f5f5f5"> <tr>
  <th><a href="index.html">главная</a>
</th>
  <th><a href="sof.html">софийский собор</a>
</th>
  <th><a href="nar.html">озеро Нарочь</a>
</th>
  <th><a href="bor.html">борисоглебская церковь</a>
</th>
  <th>
<a href="bre.html">брестская крепость</a>
</th>
  <th><a href="mir.html">мирский замок</a>
</th>
  <th><a href="bel.html">беловежская пуца</a>
</th>
  <th><a href="nes.html">несвижский замок</a>
</th>
</tr>
</table>
</body>
</html>

```

В заголовок (title) добавьте «Софийский собор», т.е. должно получиться <title>7 чудес Беларуси Софийский собор</title> Мы получили заготовку страницы о Софийском соборе. Теперь снова «Файл» «Сохранить как» и сохраните Вашу страницу под именем nar.html (эта страница будет содержать информацию об озере Нарочь).

В заголовок (title) добавьте «озеро Нарочь», т.е. должно получиться <title>7 чудес Беларуси озеро Нарочь</title> Мы получили заготовку страницы об озере Нарочь. Аналогично создайте страницы bor.html (о Борисоглебской церкви), bre.html (о Брестской крепости), mir.html (о Мирском замке), bel.html (о Беловежской пуце), nes.html (о Несвижском замке). Теперь верхнее меню позволит осуществлять переходы между страницами.

Задание 16

Наполнить страницы сайта контентом

Используя фотографии из папки «images» и текст из файла «описание 7 чудес света.doc», оформите все страницы сайта.

Задание 17

Создать страницу обратной связи

Создать и оформить страницу «Чудеса Вашей местности», на которой пользователь может отправить на материалы о известных ему достопримечательностях. Использовать текст <a Ваш адрес Размещение сайта в сети. Выполните следующие действия по поиску сервера в сети ПК: Пуск Поиск Компьютеры или людей Компьютер в сети Найти. Откройте папку htdocs. Создайте свою папку. Скопируйте в нее все файлы своего сайта. Запустите браузер. В строке адреса введите и нажмите Enter. Откроется страница с папками имеющихся сайтов. Откройте сайт щелчком по имени своей папки.



Рисунок 1 - Вид созданного сайта

Задание 18

Создать регулярные выражения

Постройте регулярное выражение, описывающее шаблон для проверки номера телефона, и внесите все необходимые изменения и дополнения в функцию *CheckData()*.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Изучить лекционный материал и материал, представленный в методических указаниях и в рекомендуемых источниках.

Основная литература

1. Савельев, А.О. HTML5. Основы клиентской разработки / А.О. Савельев, А.А. Алексеев. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 272 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429150> (10.05.2018).
2. Крахоткина, Е.В. Технологии разработки Internet-приложений: учебное пособие / Е.В. Крахоткина; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2016. - 124 с. : ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=459070>
3. Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 265 с.: ил.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428953> (09.11.2017).
4. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с.: ил.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429078> (09.11.2017).
5. Информационные технологии. HTML и XHTML: учебное пособие / А.И. Костюк, С.М. Гушанский, М.Ю. Поленов, Б.В. Катаев; Министерство образования и науки Российской Федерации, Южный федеральный университет, Инженерно-технологическая академия. - Таганрог: Издательство Южного федерального университета, 2015. - 131 с.: ил. - Библиогр. в кн. - ISBN 978-5-9275-1329-1; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=461923> (03.04.2018).
6. Лыткина, Е.А. Основы языка HTML : учебное пособие / Е.А. Лыткина, А.Г. Глотова ; Министерство образования и науки Российской Федерации, Федеральное государственное

автономное образовательное учреждение высшего профессионального образования Северный (Арктический) федеральный университет им. М.В. Ломоносова. - Архангельск : САФУ, 2014. - 104 с. : табл., ил. - Библиогр. в кн. - ISBN 978-5-261-01010-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=436328> (10.05.2018).

Дополнительная литература

7. Информационные технологии: лабораторный практикум / Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет»; авт.-сост. С.В. Говорова, М.А. Лапина. - Ставрополь: СКФУ, 2016. - 168 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=459048> (09.11.2017).

8. Информационные Web-технологии / Ю. Громов, О.Г. Иванова, Н.Г. Шахов, В.Г. Однолько ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2014. - 96 с. : ил. - Библиогр. в кн. - ISBN 978-5-8265-1365-1 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=277935> (10.05.2018).

9. Малашкевич, В.Б. Интернет-программирование : лабораторный практикум / В.Б. Малашкевич ; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с. : ил. - Библиогр.: с. 82. - ISBN 978-5-8158-1854-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=476400> (10.05.2018).

10. Технологии защиты информации в компьютерных сетях / Н.А. Руденков, А.В. Пролетарский, Е.В. Смирнова, А.М. Суоров. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 369 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428820> (10.05.2018).

11. Применение каскадных таблиц стилей (CSS) : курс / П. Храмцов, С. Брик, А. Русак, А. Сурин. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 82 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 978-5-94774-648-8 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429258> (10.05.2018).

Контрольные вопросы для самопроверки

1. Каким образом выполняются веб-приложения на клиентском компьютере?
2. Какие ограничения и почему накладываются на клиентские приложения?
3. Какие инструменты используются для разработки интернет приложений?
4. В чем отличие между JavaScript и JScript?
5. К какому классу языков относится язык JScript?
6. Какую структуру имеет программа на языке JScript?
7. Какие типы данных поддерживаются в JScript?
8. Какие операторы используются в JScript?
9. В чем состоит особенность работы с объектами в JScript?
10. Что такое HTML?
11. Чем характеризуется технология HTML?

Лабораторная работа 3. Принципы веб-дизайна

Цель работы:

1. Ознакомление с основными понятиями и принципами веб-дизайна.
2. Изучение основных возможностей по разработке веб-страниц и веб-сайтов.

Задание:

1. Изучить теоретические сведения

Порядок выполнения:

- 1) прочитать лекцию, теоретические сведения и рассмотреть приведенные в лекции образцы решения заданий;
- 2) выполнять задания лабораторного занятия аналогично.

Форма отчетности:

Подготовить отчет, содержащий следующие пункты:

- Титульный лист;
- Содержание;
- Краткая теория;
- Результаты работы (скриншоты выполненных заданий);
- Выводы.

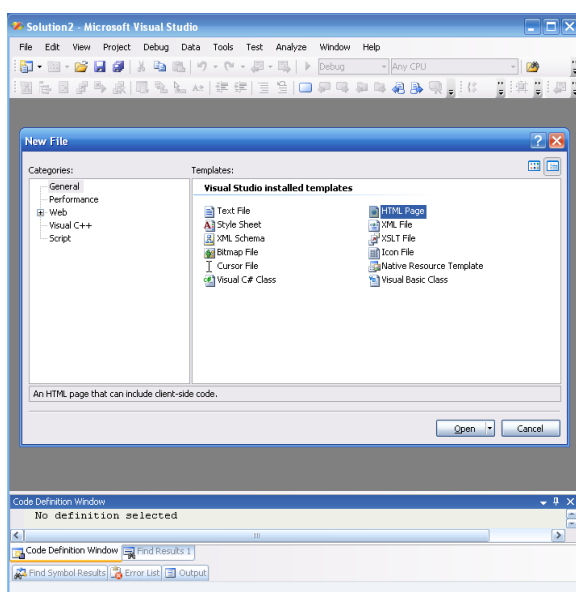
Выполнить лабораторное задание и сохранить созданные файлы в папке с именем ЛР Веб фамилия группа.

Задания для самостоятельной работы:

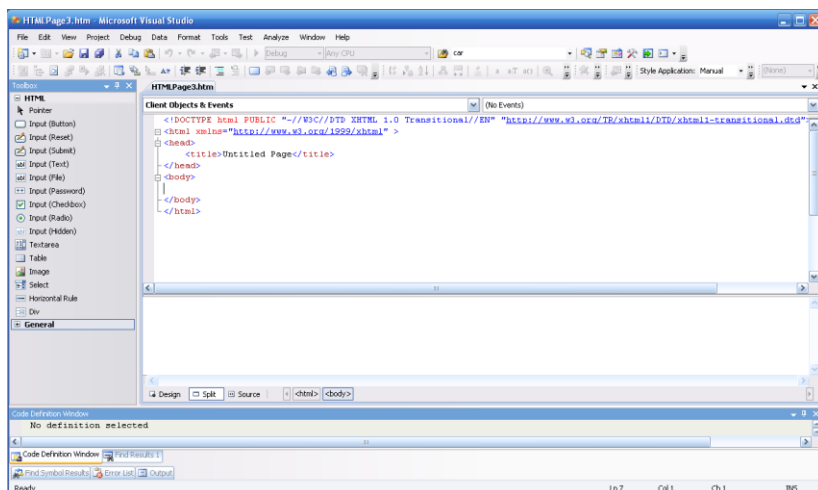
Задание 1

Создать HTML страницу в *Microsoft Visual Studio. NET*.

Для дизайна веб-страниц в Visual Studio. NET необходимо выбрать раздел меню *File => New* и в открывшемся окне выбрать тип файла *HTML Page*:



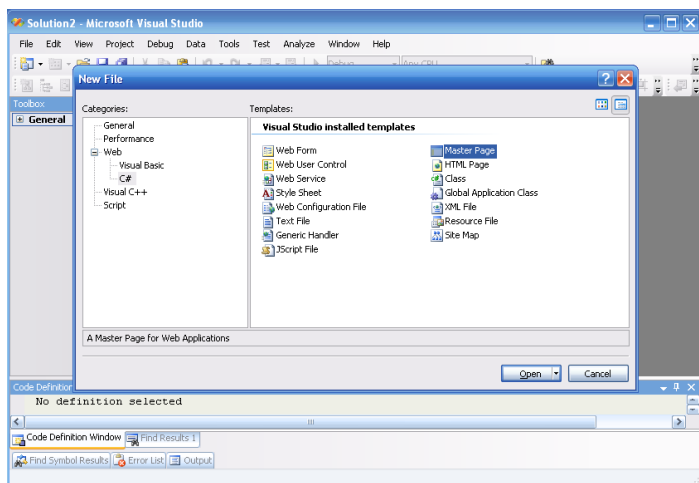
Для дизайна имеется панель Toolbox, содержащая набор основных элементов html страницы (подключается через меню: *View > Toolbox*):



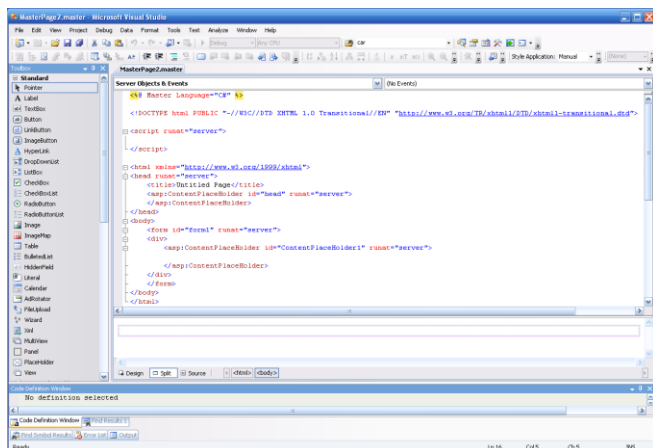
Задание 2

Создать мастер-страницы в *Microsoft Visual Studio .NET*.

Для создания мастер-страницы необходимо выбрать раздел меню *File => New* и в открывшемся окне выбрать тип файла *Web > C# > Master Page*:



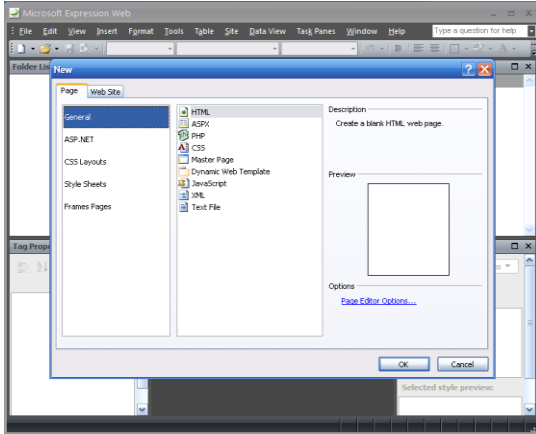
После этого появится окно мастер-страницы, содержащей пустую форму и два контейнера содержимого *ContentPlaceHolder* (в заголовке и в теле документа):



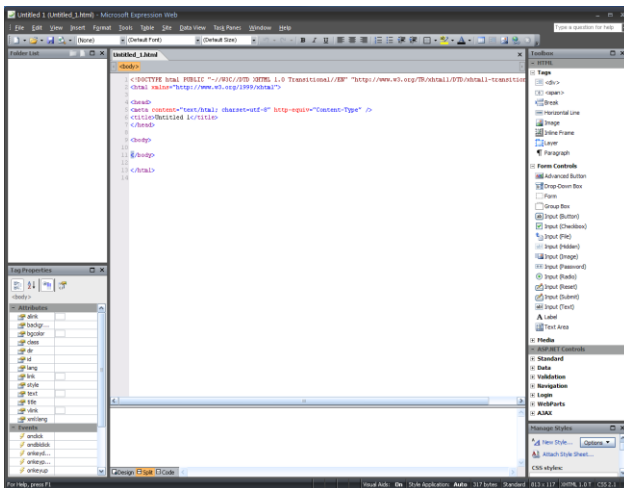
Задание 3

Ознакомьтесь с *Microsoft Expression Web*

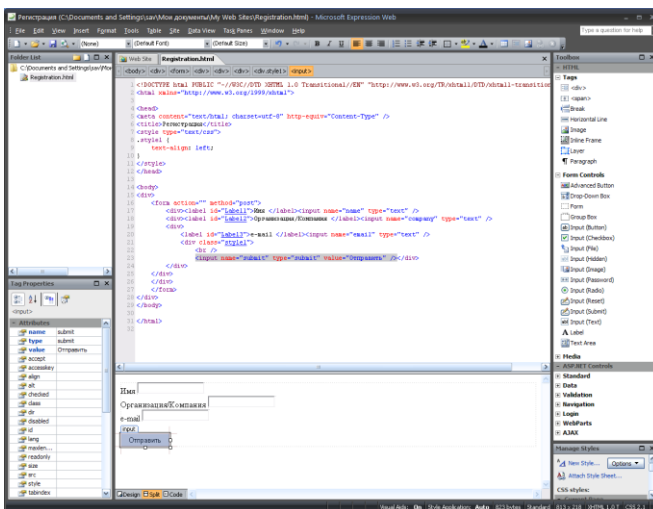
Создание нового документа. *Microsoft Expression Web* можно использовать в качестве визуального редактора документов разнообразных форматов: HTML, ASP.NET, PHP, CSS, JavaScript, XML. Поддерживается работа с шаблонами, мастер-страницами и веб-сайтами в целом. Выбрав раздел меню *File > New*, вызовите следующее диалоговое окно:



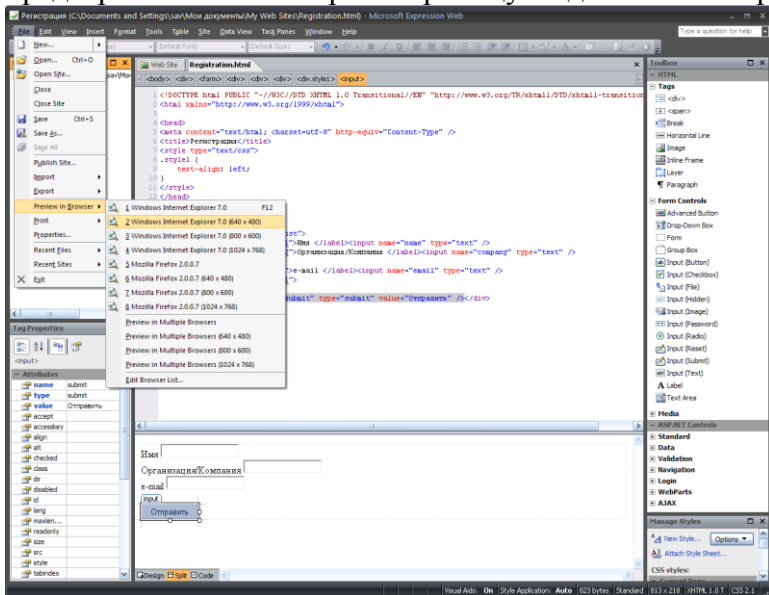
Выберите формат документа HTML. В результате появится следующее окно, содержащее пустую HTML страницу:



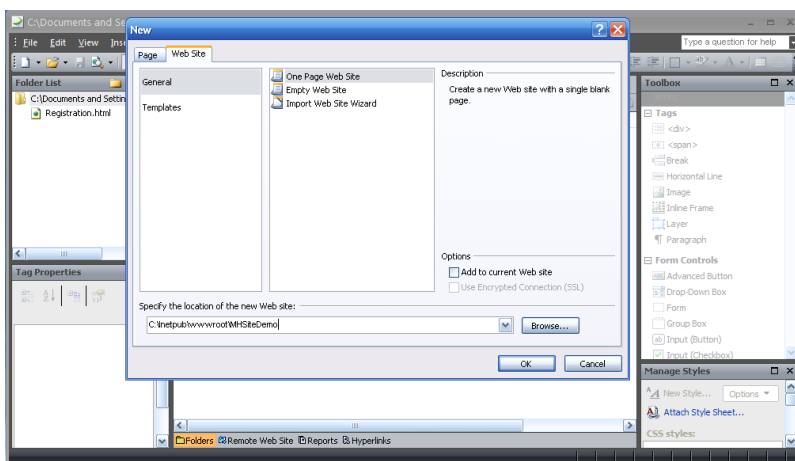
Используя панель инструментов (Toolbox), создайте веб-страницу для регистрации пользователя следующего вида:



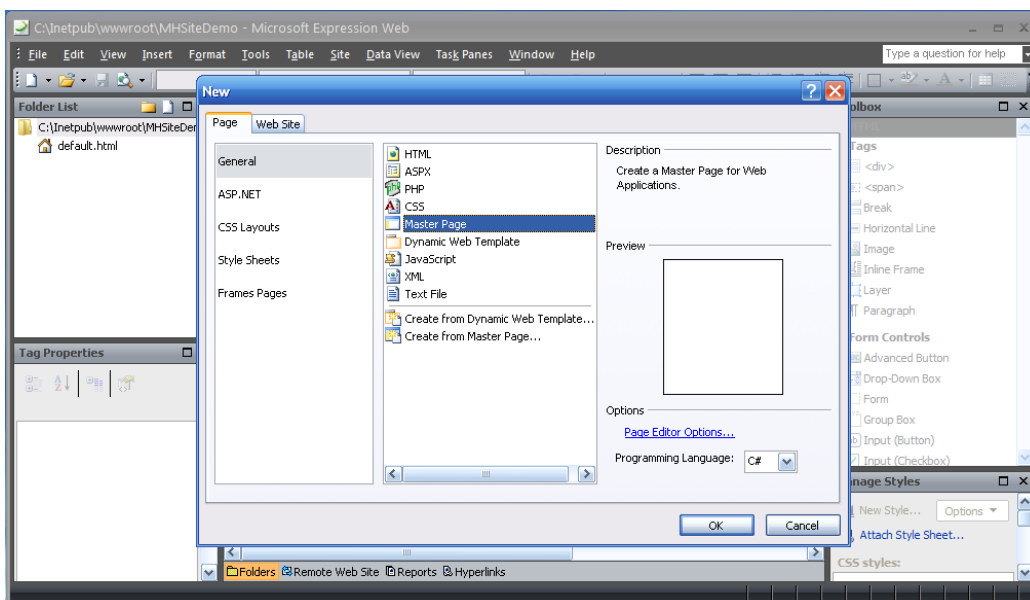
Для удобства вставки элементов формы рекомендуется использовать в качестве контейнера элемент `<div>`. После сохранения документа (*File > Save As*) можно предварительно посмотреть страницу в одном из веб-браузеров:



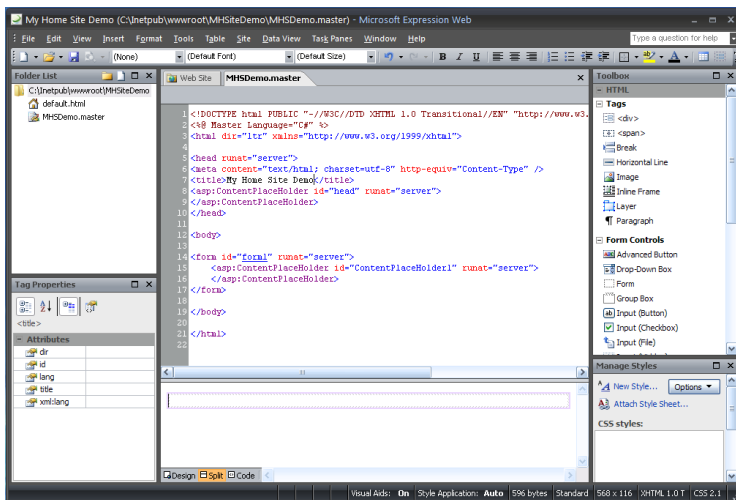
Создание веб-сайта. Для создания пустого веб-сайта выберите раздел меню *File > New*, затем на закладке *Web Site* выберите *Empty Web Site* и укажите расположение сайта:



В качестве шаблона для создания страниц веб-сайта можно использовать *мастер-страницу*:



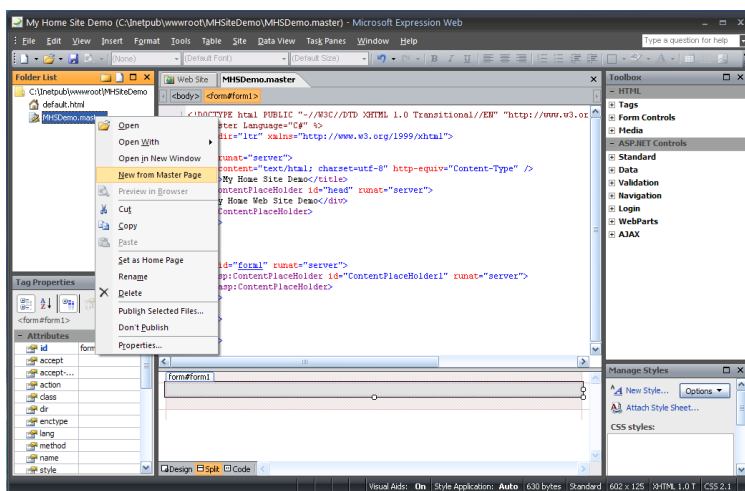
Созданная мастер-страница содержит два контейнера содержимого *ContentPlaceHolder*: в заголовке и в теле документа:



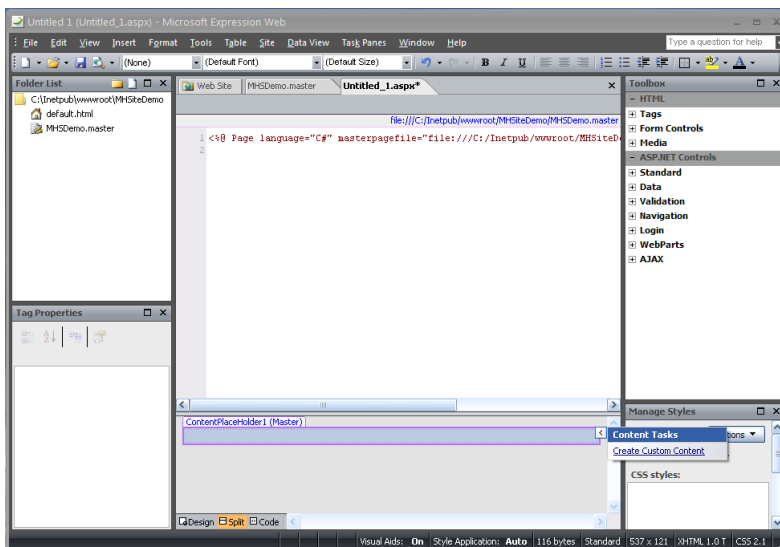
Все изменения, вносимые в эту *мастер-страницу* будут автоматически реплицироваться во всех документах веб-сайта, созданных на ее основе. Вся специфика вновь создаваемых документов должна быть локализована в пределах контейнера содержимого (*ContentPlaceHolder*). Например, в заголовке создадим общее заглавие “*My Home Web Site Demo*” для всех страниц веб-сайта:

```
<head runat="server">
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<title>My Home Site Demo</title>
  <asp:ContentPlaceHolder id="head" runat="server">
    <div>My Home Web Site Demo</div>
  </asp:ContentPlaceHolder>
</head>
```

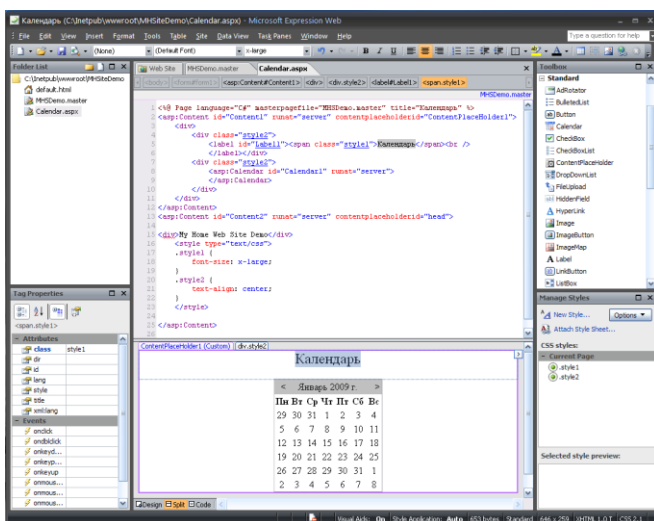
Для создания новой веб-страницы на основе мастер-страницы необходимо выбрать в панели списка файлов мастер-страницу и затем в контекстном меню (вызывается кликом правой кнопки мыши) выбрать раздел “*New from Master Page*”:



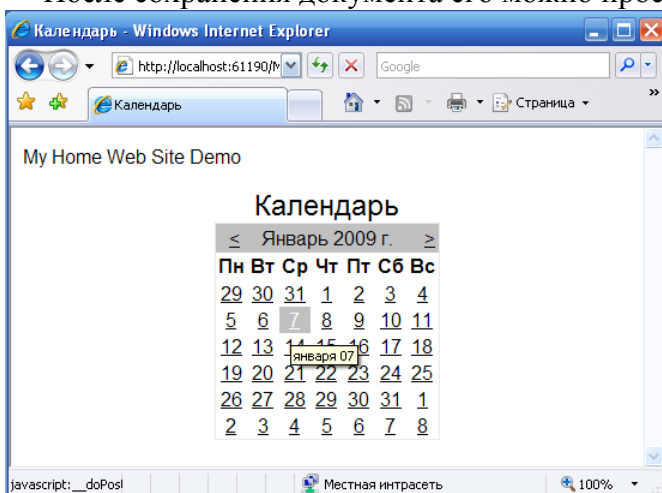
Далее, в режиме отображения “*Split*” для элемента “*ContentPlaceHolder1*” по стрелке справа от него следует выбрать из контекстного меню раздел “*Create Custom Content*”:



В результате в коде документа появятся пара тегов `</asp:Content></asp:Content>`, между которыми может быть добавлено новое содержимое страницы. В качестве примера рассмотрим следующее содержимое:



После сохранения документа его можно просмотреть в веб-браузере:



Задание 4 На страницу сайта вставить логотип

Вставьте логотип на страницах веб-сайта путем внесения необходимых изменений в мастер-страницу сайта.

Задание 5

Создать страницу регистрации пользователя

Создайте страницу регистрации пользователя веб-сайта, созданного в Microsoft Expression Web, на основе мастер-страницы.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Изучить лекционный материал и материал, представленный в методических указаниях и в рекомендуемых источниках.

Основная литература

1. Савельев, А.О. HTML5. Основы клиентской разработки / А.О. Савельев, А.А. Алексеев. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 272 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429150> (10.05.2018).

2. Крахоткина, Е.В. Технологии разработки Internet-приложений: учебное пособие / Е.В. Крахоткина; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2016. - 124 с. : ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=459070>

3. Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 265 с.: ил.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428953> (09.11.2017).

4. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с.: ил.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429078> (09.11.2017).

5. Информационные технологии. HTML и XHTML: учебное пособие / А.И. Костюк, С.М. Гушанский, М.Ю. Поленов, Б.В. Катаев; Министерство образования и науки Российской Федерации, Южный федеральный университет, Инженерно-технологическая академия. - Таганрог: Издательство Южного федерального университета, 2015. - 131 с.: ил. - Библиогр. в кн. - ISBN 978-5-9275-1329-1; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=461923> (03.04.2018).

6. Лыткина, Е.А. Основы языка HTML : учебное пособие / Е.А. Лыткина, А.Г. Глотова ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования Северный (Арктический) федеральный университет им. М.В. Ломоносова. - Архангельск : САФУ, 2014. - 104 с. : табл., ил. - Библиогр. в кн. - ISBN 978-5-261-01010-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=436328> (10.05.2018).

Дополнительная литература

7. Информационные технологии: лабораторный практикум / Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет»; авт.-сост. С.В. Говорова, М.А. Лапина. - Ставрополь: СКФУ, 2016. - 168 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=459048> (09.11.2017).

8. Информационные Web-технологии / Ю. Громов, О.Г. Иванова, Н.Г. Шахов, В.Г. Однолько ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2014. - 96 с. : ил. - Библиогр. в кн. - ISBN 978-5-8265-

1365-1 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=277935> (10.05.2018).

9. Малашкевич, В.Б. Интернет-программирование : лабораторный практикум / В.Б. Малашкевич ; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2017. - 96 с. : ил. - Библиогр.: с. 82. - ISBN 978-5-8158-1854-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=476400> (10.05.2018).

10. Технологии защиты информации в компьютерных сетях / Н.А. Руденков, А.В. Пролетарский, Е.В. Смирнова, А.М. Суоров. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 369 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428820> (10.05.2018).

11. Применение каскадных таблиц стилей (CSS) : курс / П. Храмцов, С. Брик, А. Русак, А. Сурин. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 82 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 978-5-94774-648-8 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429258> (10.05.2018).

Контрольные вопросы для самопроверки

1. Что такое портал?
2. Что такое веб-портал?
3. Какие типы веб-порталов вы знаете?
4. Для чего предназначены корпоративные порталы?
5. Какую структуру имеет типичный корпоративный портал?
6. Какие вы знаете средства для создания порталов?
7. Какую возможность предоставляют пользователю Веб 2.0 сайты?

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

1. Microsoft Windows Professional Russian
2. Microsoft Office Russian
3. Антивирусное программное обеспечение Kaspersky Security
4. Справочно-правовая система «Консультант Плюс»

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

<i>Вид занятия</i>	<i>Наименование аудитории</i>	<i>Перечень основного оборудования</i>	<i>№ ПЗ</i>
1	2	3	4
Лк	Лекционная аудитория (мультимедийный класс)	Интерактивная доска SMART Board 680i2/Unifl, Интерактивный планшет Wacom PL-720, Колонки Microlab Solo-7C, Ноутбук Samsung R610<NP-R610-FS08>, Телевизор плазменный Samsung 63 PS-63A756T1M	-
ПЗ	Дисплейный класс	Системный блок AMD A10-7800 Radeon R7 (12 шт.), Системный блок для слабовидящих пользователей AMD A10-7850K (1 шт.), Монитор Philips233 V5QHABP (13 шт.)	ПЗ № 1-13
кр	Читальный зал №1	Оборудование 10 ПК i5-2500/H67/4Gb(монитор TFT19	-

		Samsung); принтер HP LaserJet P2055D	
CP	Читальный зал №1	Оборудование 10 ПК i5- 2500/Н67/4Gb(монитор TFT19 Samsung); принтер HP LaserJet P2055D	-

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

№ компетенции	Элемент компетенции	Раздел	Тема	ФОС
ОПК-4	способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	1. Введение в современные Web-технологии	1.1. Основные понятия	Экзаменационный вопрос 1
			1.2. Особенности Web-технологий.	Экзаменационный вопрос 2
			1.3. Протоколы интернет	Экзаменационный вопрос 3
			Языки Web-программирования	Экзаменационный вопрос 4
		2. Основы Web-технологий	2.1. Язык разметки гипертекста HTML	Экзаменационный вопрос 5-12
			2.2. Каскадные таблицы стилей CSS.	Экзаменационный вопрос 13
			2.3. Основы скриптового языка JavaScript.	Экзаменационный вопрос 14
		3. Введение в Web-дизайн	3.1. Основы веб-дизайна	Экзаменационный вопрос 15 -16
			3.2. Верстка и дизайн Web-страниц	Экзаменационный вопрос 17
		ПК-8	способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	

2. Экзаменационные вопросы

№ п/п	Компетенции		ВОПРОСЫ К ЭКЗАМЕНУ	№ и наименование раздела	
	Код	Определение			
1	2	3	4	5	
1.	ОПК-3	способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	1. Основные понятия: всемирная паутина; интернет; Web-технологии; контент; URI (Uniform Resource Identifier — «унифицированный идентификатор ресурса») страницы.	1. Введение в современные Web-технологии	
			2. Web-технологии. особенности; достоинства и недостатки; интеграционная роль.		
			3. Макеты Web-сайтов. Классификация макетов: фиксированные; резиновые; эластичные; адаптивные; комбинированные. Преимущества и недостатки.		
			4. Формы. Теги описания форм.		
			5. Web-серверы и web-браузеры. Взаимодействие Web-сервера и Web-браузера.		
			6. Фреймы. Использование фреймов для верстки документа HTML, достоинства и недостатки.		
			7. HTTP (HyperText Transfer Protocol — «протокол передачи гипертекста»). Структура. Методы запросов.		
			8. Ссылки. Абсолютные и относительные ссылки, понятие относительного пути. Виды ссылок: посещенные; непосещенные; активные. Атрибуты ссылок. Якоря.		
			9. Оформление документа HTML с помощью каскадных таблиц стилей.		2. Основы Web-технологий
			10. HTML (HyperText Markup Language — «язык разметки гипертекста»). Структура документа HTML. Основные теги.		
			11. Списки. Нумерованные списки. Маркированные списки. Списки определений.		
			12. Теги. Парные и непарные теги. Атрибуты тегов. Типы тегов: теги верхнего уровня; теги заголовка документа; блочные элементы; строчные элементы; универсальные элементы; теги для таблиц; теги для фреймов.		
			13. CSS (Cascading Style Sheets — «каскадные таблицы стилей»). Способы использования каскадных		
2.	ОПК-4	способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности			

		таблиц стилей в документе HTML.	
		14. JavaScript, способы использования в документе HTML.	3. Введение в Web-дизайн
		15. Таблицы. Использование таблиц для верстки документа HTML, достоинства и недостатки.	
		16. Веб-дизайна. Составляющие веб-дизайна	
		17. Верстка Web-страниц	

3. Описание показателей и критериев оценивания компетенций

Показатели	Оценка	Критерии
<p>Знать (ОПК-4):</p> <ul style="list-style-type: none"> - технологии создания интернет-проектов; - языки: HTML5, CSS3, JavaScript, JQuery; - программное обеспечение для создания интернет-приложений; - предметную область разработки интернет-приложения; - перспективы развития предметной области; <p>(ПК-8):</p> <ul style="list-style-type: none"> - структуру и принципы работы всемирной паутины; - основные виды современных Web-технологий и принципы их функционирования; <p>Уметь (ОПК-4):</p> <ul style="list-style-type: none"> - разрабатывать интернет-сайт или интернет-приложение по заданной тематике; - проводить оптимизацию структуры сайта и файлов; - готовить текстовый, графический, видео и аудио контент для сайта; - опубликовать сайт в интернете; - создать онлайн-предприятие для своей будущей 	отлично	<p>Свободно и уверенно разрабатывает интернет-сайт или интернет-приложение по заданной тематике, отлично владеет навыками технологий создания интернет-проектов; знает языки: HTML5, CSS3, JavaScript, JQuery, предусмотренные учебной программой, знает типичные ошибки и возможные сложности при подготовке текстового, графического, видео и аудио контента для сайта. Демонстрирует на высоком уровне навыки разработки веб-сайтов. Грамотно использует основные виды современных Web-технологий и принципы их функционирования. Способен решать задачи производственной и технологической деятельности на профессиональном уровне, включая: разработку алгоритмических и программных решений в области интернет программирования.</p>
	хорошо	<p>В большинстве случаев разрабатывает интернет-сайт или интернет-приложение по заданной тематике, владеет навыками технологий создания интернет-проектов; знает языки: HTML5, CSS3, JavaScript, JQuery, предусмотренными учебной программой. В большинстве случаев знает типичные ошибки и возможные сложности при подготовке текстового, графического, видео- и аудио- контента для сайта. Допускает единичные серьезные ошибки в решении проблем, испытывает сложности в редко встречающихся или сложных случаях решения проблем, не знает типичных ошибок и возможных сложностей при</p>

<p>профессиональной деятельности; (ПК-8):</p> <ul style="list-style-type: none"> - структурировать информацию для её публикации на Web-ресурсах; - применять современные Web-технологии в профессиональной деятельности; 		<p>решении той или иной проблемы. Демонстрирует на достаточном уровне навыки разработки веб-сайтов. Изредка использует при этом основные виды современных Web-технологий и принципы их функционирования и способность решать задачи производственной и технологической деятельности на профессиональном уровне, включая: разработку алгоритмических и программных решений в области интернет программирования.</p>
<p>Владеть (ОПК-4):</p> <ul style="list-style-type: none"> – навыками создания сайтов и веб-приложений по заданной тематике; - навыками написания технического задания, документации пользователя, администратора и разработчика <p>(ПК-8):</p> <ul style="list-style-type: none"> – навыками разработки макетов Web-страниц, отвечающих современным требованиям; - навыками администрирования систем управления контентом; - навыками разработки шаблонов для систем управления контентом. 	<p>удовлетворительно</p>	<p>Допускает ошибки в разрабатывает интернет-сайт или интернет-приложение по заданной тематике, отлично владеет навыками технологий создания интернет-проектов; знает языки: HTML5, CSS3, JavaScript, JQuery, предусмотренные учебной программой. Демонстрирует на низком уровне способность применять навыки разработки веб-сайтов. В отдельных случаях способен правильно решать только типичные, наиболее часто встречающиеся проблемы, задачи при разработке Веб-сайтов. Демонстрирует на низком уровне навыки использования основных видов современных Web-технологий и принципов их функционирования; способность решать задачи производственной и технологической деятельности на профессиональном уровне, включая: разработку алгоритмических и программных решений в области интернет программирования.</p>
	<p>неудовлетворительно</p>	<p>Неспособен разрабатывать интернет-сайт или интернет-приложение по заданной тематике, не владеет навыками технологий создания интернет-проектов; не знает языки: HTML5, CSS3, JavaScript, JQuery, предусмотренные учебной программой. Не владеет методами создания веб-сайтов; способностью решать задачи производственной и технологической деятельности на профессиональном уровне, включая: разработку алгоритмических и программных решений в области интернет программирования.</p>

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Дисциплина Интернет-программирование направлена на ознакомление с современными Web - технологиями; на получение теоретических знаний и практических навыков разработки веб-сайтов для их дальнейшего использования в практической деятельности.

Изучение дисциплины Интернет-программирование предусматривает:

- лекции,
- лабораторные работы;
- практические занятия;
- самостоятельную работу.

В ходе освоения раздела **1** Введение в современные Web-технологии студенты должны уяснить Основные понятия и особенности Web-технологий. Необходимо овладеть навыками и умениями применения изученного для реализации тех или иных проектов в конкретных ситуациях.

В ходе освоения раздела **2** Основы Web-технологий студенты должны уяснить Язык разметки гипертекста HTML, каскадные таблицы стилей CSS, основы скриптового языка JavaScript. Необходимо овладеть навыками и умениями применения изученных технологий для реализации тех или иных проектов в конкретных ситуациях.

В ходе освоения раздела **3** Введение в Web-дизайн студенты должны уяснить принципы компоновки, верстки и дизайна Web-страницы. Необходимо овладеть навыками и умениями применения изученных принципов для реализации тех или иных проектов в конкретных ситуациях.

Обучающимся необходимо овладеть навыками и умениями применения изученных методов для разработки и реализации профессионально ориентированных проектов в последующей учебной деятельности.

В процессе изучения дисциплины рекомендуется на первом этапе обратить внимание на специфику компьютерных технологий и умение выбирать методы решения различных задач.

Овладение ключевыми понятиями является основой усвоения учебного материала по дисциплине.

При подготовке к зачету рекомендуется особое внимание необходимо уделить рекомендациям и замечаниям преподавателей, ведущих аудиторные занятия по дисциплине.

В процессе проведения лабораторных и практических работ происходит закрепление знаний, формирование умений и навыков реализации представления о методах обработки информации с помощью компьютерных технологий.

Самостоятельную работу необходимо начинать с чтения лекций и учебников.

В процессе консультации с преподавателем обучающийся выясняет наличие пробелов в знаниях и способах решения разных ситуаций.

Работа с литературой является важнейшим элементом в получении знаний по дисциплине. Прежде всего, необходимо воспользоваться списком рекомендуемой по данной дисциплине литературой. Дополнительные сведения по изучаемым темам можно найти в периодической печати и Интернете.

При проведении аудиторных занятий в виде разнообразных тренингов и ситуаций рекомендуется активно участвовать и отмечать уровень собственных знаний и умение общаться в сочетании с внеаудиторной работой.

АННОТАЦИЯ

рабочей программы дисциплины

Интернет-программирование

1. Цель и задачи дисциплины

Цель изучения дисциплины Интернет-программирование – изучение и освоение современных Web - технологий для разработки и эксплуатации интернет ресурсов.

Задачей изучения дисциплины является:

- освоение основных концепций и принципов разработки интернет-приложений,
- освоение основных концепций и принципов программирования в интернет,
- освоение современных технологий для разработки Web-сайтов,
- приобретение навыков по использованию технологий HTML, CSS, JavaScript в интернет.
- приобретение навыков по программированию в интернет
- приобретение практических навыков публикации и сопровождения Web-сайтов и Web-приложений.

2. Структура дисциплины

2.1 Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: Лк.-17 час.; ЛР – 34 час., СР - 57 час.

Общая трудоемкость дисциплины составляет . 144 часа, 4 зачетные единицы

2.2 Основные разделы дисциплины:

- 1 – Введение в современные Web-технологии
- 2 – Основы Web-технологий
- 3 – Введение в Web-дизайн

3. Планируемые результаты обучения (перечень компетенций)

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ОПК-4 - способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

ПК-8 - способностью программировать приложения и создавать программные прототипы решения прикладных задач.

4. Вид промежуточной аттестации: экзамен

*Протокол о дополнениях и изменениях в рабочей программе
на 20__-20__ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

2. В рабочую программу по дисциплине вносятся следующие изменения:

Протокол заседания кафедры № _____ от «___» _____ 20 ____ г.,
(разработчик)

Заведующий кафедрой _____
(подпись)

(Ф.И.О.)

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.03 Прикладная информатика от «12» марта 2015 г. № 207

для набора 2014 года: и учебным планом ФГБОУ ВО «БрГУ» для заочной формы обучения от «03» июля 2018 г. № 413

для набора 2015 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «03» июля 2018 г. № 413, заочной формы обучения от «03» июля 2018 г. № 413

для набора 2016 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «05» мая 2016 г. № 342

для набора 2017 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06» марта 2017 г. №125, заочной формы обучения от «06» марта 2017 г. №125

для набора 2018 года и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «12» марта 2018 г. № 130, заочной формы обучения от «12» марта 2018 г. № 130

Программу составил:

Сыгодина М.В., доцент базовой кафедры МиИТ, к.т.н _____

Рабочая программа рассмотрена и утверждена на заседании базовой кафедры МиИТ

от «19» декабря 2018 г., протокол № 8

И.о. заведующего базовой кафедрой МиИТ _____ Е.И. Луковникова

СОГЛАСОВАНО:

И.о. заведующего выпускающей базовой кафедрой МиИТ _____ Е.И. Луковникова

Директор библиотеки _____ Т.Ф. Сотник

Рабочая программа одобрена методической комиссией факультета ФЭиУ

от «28» декабря 2018 г., протокол № 4

Председатель методической комиссии факультета _____ Е.В. Трапезникова

СОГЛАСОВАНО:

Начальник учебно-методического управления _____ Г.П. Нежевец

Регистрационный № _____