

АВТНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Утверждаю
Декан факультета _____ Ж.В. Игнатенко
« 28 » _____ СЕНТЯ 10 2020 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Интернет-программирование

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) программы: Прикладная информатика в экономике

Квалификация выпускника бакалавр

Форма обучения очная, заочная

год начала подготовки – 2019

Разработана
Канд.экон.наук, доцент, доцент
_____ А.Ю. Орлова

Согласована
зав. выпускающей кафедры
_____ Ж.В. Игнатенко

Рекомендована
на заседании кафедры
от « 28 » _____ 10 2020 г.
протокол № 2 _____
Зав. кафедрой _____ Ж.В. Игнатенко

Одобрена
на заседании учебно-методической
комиссии факультета
от « 28 » _____ 10 2020 г.
протокол № 2 _____
Председатель УМК _____ Ж.В.
Игнатенко

Ставрополь, 2020 г.

Содержание

1. Цели освоения дисциплины	3
2. Место дисциплины в структуре ОПОП.....	3
3. Планируемые результаты обучения по дисциплине	4
4. Объем дисциплины и виды учебной работы	4
5. Содержание и структура дисциплины.....	6
5.1. Содержание дисциплины.....	6
5.2. Структура дисциплины	7
5.3. Занятия семинарского типа	8
5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа)	9
5.5. Самостоятельная работа	9
6. Образовательные технологии.....	10
7. Фонд оценочных средств (оценочные материалы) для текущего контроля успеваемости, промежуточной аттестации	10
8. Учебно-методическое и информационное обеспечение дисциплины	10
8.1. Основная литература.....	10
8.2. Дополнительная литература	11
8.3 программное обеспечение	11
8.4 профессиональные базы данных.....	11
8.5. Информационные справочные системы.....	11
8.6. Интернет-ресурсы.....	11
8.7. Методические указания по освоению дисциплины	11
9. Материально-техническое обеспечение дисциплины	14
10. Особенности освоения дисциплины лицами с ограниченными возможностями здоровья.....	14
Приложение 1.....	16

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью дисциплины «Интернет программирование» является формирование профессиональных компетенций будущего бакалавра направления 09.03.03 «Прикладная информатика».

Задачи дисциплины:

- ознакомление с основными методами и средствами разработки сетевых приложений;
- изучение особенностей архитектуры современных вычислительных сетей;
- приобретение навыков веб-программирования;
- знакомство с методами и технологиями обработки информации в вычислительных сетях;
- приобретение навыков анализа эффективности использования выбранных информационных технологий и систем.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина (Б.1.В.10) «Интернет программирование» входит в часть, формируемую участниками образовательных отношений, – обязательные дисциплины Блока 1 «Дисциплины (модули)» и находится в логической и содержательно-методической связи с другими дисциплинами.

Предшествующие дисциплины (курсы, модули, практики)	Последующие дисциплины (курсы, модули, практики)
Визуальное программирование	Управление проектами
Разработка программных приложений	Администрирование информационных систем
Информационные системы и технологии	Преддипломная практика
Мировые информационные ресурсы	
Разработка мобильных приложений	

Требования к «входным» знаниям, умениям и навыкам обучающегося, необходимым при освоении данной дисциплины

Знать:

- структуру, принципы реализации и функционирования информационных технологий, используемых при создании информационных систем, базовые и прикладные информационные технологии, инструментальные средства информационных технологий;
- методы разработки и интеграции информационных систем, их компонентов и информационных сервисов с использованием Delphi;
- методы программирования приложений и создания программных прототипов решения прикладных задач.

Уметь:

- использовать архитектурные и детализированные решения при проектировании систем;
- применять информационные технологии при проектировании информационных систем;
- разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования.

Владеть:

- моделями и средствами разработки архитектуры информационных систем;
- методами и средствами представления данных и знаний о предметной области, методами и средствами анализа информационных систем, технологиями реализации, внедрения проекта информационной системы, методологией

использования информационных технологий при создании информационных систем;

- навыками проведения тестирования компонентов ПО ИС;
- навыками программирования в современных средах.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Формируемые компетенции (код компетенции, уровень)	Планируемые результаты обучения на данном этапе формирования компетенции
ПК-2 способность разрабатывать и адаптировать прикладное программное обеспечение	Знать: принципы разработки и адаптации прикладного ПО; принцип действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основные типы селекторов и правила расчета приоритетов правил CSS и основные принципы разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.
	Уметь: разрабатывать и адаптировать прикладное ПО; использовать структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; использовать основные типы селекторов и правила расчета приоритетов правил CSS и основные принципы разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.
	Владеть: владеет навыками разработки и адаптации прикладного ПО; использования структуры запросов протокола HTTP и способов создания и передачи документов гипертекстового языка разметки HTML; использования основных типов селекторов и правил расчета приоритетов правил CSS и основных принципов разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.
ПК-8 способность проводить тестирование компонентов программного обеспечения	Знать: методы разработки многокомпонентных информационных систем, реализованных на различных платформах.
	Уметь: разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования.
	Владеть: навыками проведения тестирования компонентов ПО ИС; навыками программирования Web-интерфейса в современных средах.

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общий объем дисциплины составляет 4 зачетных единиц, 144 академических часа.

Очная форма обучения.

Вид учебной работы	Всего часов	Триместр
		9
Контактная работа (всего)	54	54
в том числе:		
1) занятия лекционного типа (ЛК)	18	18
из них		
– лекции	18	18
2) занятия семинарского типа (ПЗ)	36	36
из них		
– семинары (С)	-	-
– практические занятия (ПР)	36	36
– лабораторные работы (ЛР)	-	-
3) групповые консультации	-	-
4) индивидуальная работа	-	-
5) промежуточная аттестация	-	-
Самостоятельная работа (всего) (СР)	90	90
в том числе:		
Курсовой проект (работа)	-	-
Расчетно-графические работы	-	-
Контрольная работа	-	-
Реферат	-	-
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)	90	90
Подготовка к аттестации	-	-
Общий объем, час	144	144
Форма промежуточной аттестации		Диф. зачет

Заочная форма обучения.

Вид учебной работы	Всего часов	Триместр
		9
Контактная работа (всего)	16	16
в том числе:		
1) занятия лекционного типа (ЛК)	8	8
из них		
– лекции	8	8
2) занятия семинарского типа (ПЗ)	8	8
из них		
– семинары (С)	-	-
– практические занятия (ПР)	8	8
– лабораторные работы (ЛР)	-	-
3) групповые консультации	-	-
4) индивидуальная работа	-	-
5) промежуточная аттестация	0,3	0,3
Самостоятельная работа (всего) (СР)	127,7	127,7
в том числе:		
Курсовой проект (работа)	-	-

Расчетно-графические работы	-	-
Контрольная работа	-	-
Реферат	-	-
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)	124	124
Подготовка к аттестации	3,7	3,7
Общий объем, час	144	144
Форма промежуточной аттестации		Диф. зачет

5. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

5.1. Содержание дисциплины

№ раздела (темы)	Наименование раздела (темы)	Содержание раздела (темы)
1.	1. Принципы работы и структура Web-приложений на основе ASP.NET	Архитектура Web-приложений Краткое описание архитектуры ASP.NET и .NET Framework Краткие итоги
2.	2. Основы работы в VisualStudio .NET 2005	Начало работы с VisualStudio 2005 Создание нового приложения Редактирование Web-документов Установка точек прерывания и просмотр значений переменных Пошаговое исполнение Редактирование кода Окно SolutionExplorer. Структура ASP.NET приложения Работа со справочной системой Запуск приложения и организация виртуальных каталогов приложений Структура каталога приложения Краткие итоги
3.	3. Основы языка C#	Основные операторы языка C# Типы данных. Преобразования типов Массивы, перечисления, коллекции Работа со строками в C# Процедуры и функции Классы и структуры Краткие итоги
4.	4: Основы Web-программирования с использованием ASP.NET	Динамическое создание элементов управления Класс Page Коллекция ViewState, объекты Session и Application Использование строки запроса Объекты Response и Request Краткие итоги
5.	5. Принципы разработки пользовательского	Использование элементов CheckBox, CheckBoxList, RadioButton, RadioButtonList и BulletedList

	интерфейса приложения интернет-	Использование Image, ImageMap, ImageButton Использование Literal и HiddenField Использование Panel Использование LinkButton Проверка вводимых данных Краткие итоги
6.	6. Навигация по Web-приложению	Карты сайта Использование элементов управления TreeView, Menu и SiteMapPath Навигация с использованием программирования Использование MultiView и Wizard Краткие итоги
7.	7. Использование тем при оформлении Web-приложения	Создание и использование тем Совместное использование Themes и CSS Динамическое изменение темы Краткие итоги
8.	8. Использование кэширования в Web-приложениях	Основы кэширования в ASP.NET Кэширование вывода Фрагментное кэширование Краткие итоги
9.	9. Использование баз данных в приложениях ASP.NET	Основные сведения о модели доступа к данным ADO.NET Организация взаимодействия с БД Отсоединенные наборы данных Привязка и отображение данных Обновление данных Использование DataView для фильтрации и сортировки данных Реализация трехуровневой архитектуры доступа к данным в ASP.NET Краткие итоги

5.2. Структура дисциплины

Очная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов			
		Всего	Л	ПЗ (ПР)	СР
1.	Принципы работы и структура Web-приложений на основе ASP.NET	16	2	4	10
2.	Основы работы в VisualStudio .NET 2005	16	2	4	10
3.	Основы языка C#	16	2	4	10
4.	Основы Web-программирования с использованием ASP.NET	16	2	4	10
5.	Принципы разработки пользовательского интерфейса интернет-приложения	16	2	4	10
6.	Навигация по Web-приложению	16	2	4	10
7.	Использование тем при оформлении Web-приложения	16	2	4	10
8.	Использование кэширования в Web-приложениях	16	2	4	10
9.	Использование баз данных в приложениях	16	2	4	10

	ASP.NET				
	Общий объем:	144	18	36	90

Заочная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов			
		Всего	Л	ПЗ (ПР)	СР
1.	Принципы работы и структура Web-приложений на основе ASP.NET	16	1	1	14
2.	Основы работы в VisualStudio .NET 2005	14	1	1	12
3.	Основы языка C#	16	1	1	14
4.	Основы Web-программирования с использованием ASP.NET	16	1	1	14
5.	Принципы разработки пользовательского интерфейса интернет-приложения	15	1		14
6.	Навигация по Web-приложению	15		1	14
7.	Использование тем при оформлении Web-приложения	16	1	1	14
8.	Использование кэширования в Web-приложениях	16	1	1	14
9.	Использование баз данных в приложениях ASP.NET	16	1	1	14
	Промежуточная аттестация	4			
	Общий объем:	144	8	8	124

5.3. Занятия семинарского типа

Очная форма обучения.

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
1.	1.	ПР	1. HTML. История создания. Структура документов	4
2.	2.	ПР	2. CSS. История. Форматирование текста	4
3.	3.	ПР	3. CSS. Блочная верстка документов	4
4.	4.	ПР	4. Установка и настройка комплекса программных продуктов WAMP	4
5.	5.	ПР	5. PHP. Переменные и константы	4
6.	6.	ПР	6. PHP. Типы данных. Идентичность переменных	4
7.	7.	ПР	7. PHP. Работа со строками. Базовые функции	4
8.	8.	ПР	8. PHP. Работа с массивами. Базовые функции	4
9.	9.	ПР	9. PHP. Работа с файловой системой.	4

Заочная форма обучения.

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
1.	1.	ПР	1. HTML. История создания. Структура документов	1

2.	2.	ПР	2. CSS. История. Форматирование текста	1
3.	3.	ПР	3. CSS. Блочная верстка документов	1
4.	4.	ПР	4. Установка и настройка комплекса программных продуктов WAMP	1
5.	6.	ПР	6. PHP. Типы данных. Идентичность переменных	1
6.	7.	ПР	7. PHP. Работа со строками. Базовые функции	1
7.	8.	ПР	8. PHP. Работа с массивами. Базовые функции	1
8.	9.	ПР	9. PHP. Работа с файловой системой.	1

5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа)

Не предусмотрены.

5.5. Самостоятельная работа

Очная форма обучения.

№ раздела (темы)	Виды самостоятельной работы	Количество часов
1.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
2.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
3.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
4.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
5.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
6.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
7.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
8.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
9.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10

Заочная форма обучения.

№ раздела (темы)	Виды самостоятельной работы	Количество часов
1.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14
2.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	12
3.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14
4.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14
5.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14
6.	Подготовка к лекциям и практическим занятиям.	14

	Изучение источников информации по дисциплине.	
7.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14
8.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14
9.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине:

- сбор, хранение, систематизация, обработка и представление учебной и научной информации;
- обработка различного рода информации с применением современных информационных технологий;
- самостоятельный поиск дополнительного учебного и научного материала, с использованием поисковых систем и сайтов сети Интернет, электронных энциклопедий и баз данных;
- использование электронной почты для рассылки и асинхронного общения, чата преподавателей и обучающихся, переписки и обсуждения возникших учебных проблем для синхронного взаимодействия
- дистанционные образовательные технологии (при необходимости).

Интерактивные и активные образовательные технологии

№ раздела (темы)	Вид занятия (Л, ПЗ, С, ЛР)	Используемые интерактивные образовательные технологии	Количество часов ОФО/ЗФО
1.	Л.	Дискуссия	4/1
2.	ПР.	Проблемное обучение	2/1
3.	Л.	Интерактивная лекция	2/1
6.	ПР.	Проблемное обучение	2/1
7.	Л.	Дискуссия	4/1
9.	Л.	Интерактивная лекция	2/1

Практическая подготовка обучающихся не предусмотрена

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Фонд оценочных средств (оценочные материалы) для текущего контроля успеваемости, промежуточной аттестации по дисциплине приводятся в приложении.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Основная литература

1. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2020. — 218 с. — (Высшее образование). — ISBN 978-5-534-00515-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/451207>.

2. Сысолетин, Е. Г. Разработка интернет-приложений : учебное пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под научной редакцией Л. Г. Доросинского. — Москва : Издательство Юрайт, 2020. — 90 с. — (Высшее образование). — ISBN 978-5-9916-9975-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/45334>).

8.2. Дополнительная литература

1. Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений: учебное пособие для вузов / В. В. Соколова. — Москва: Издательство Юрайт, 2020. — 175 с. — (Высшее образование). — ISBN 978-5-9916-6525-4. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/451366>.

8.3 Программнообеспечение

1. Microsoft Word,
2. Google Chrome,
3. Microsoft SharePoint Designer,
4. VisualStudio.NET 2005

8.4 Профессиональные базы данных

1. Электронная библиотечная система «СКСИ» [Электронный ресурс] – Режим доступа: <https://www.sksi.ru/environment/ebs/1363/>
2. Интернет университет информационных технологий [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/>
3. Электронная библиотечная система «Urait» [Электронный ресурс] – Режим доступа: <https://urait.ru/>
4. Электронная библиотека информационных технологий CITForum.ru [Электронный ресурс] – Режим доступа: (<http://citforum.ru>).
5. Виртуальная академия Microsoft [Электронный ресурс] – Режим доступа (<http://aka.ms/studentcourse>).

8.5. Информационные справочные системы

1. Информационно-справочная система для программистов[Электронный ресурс] – Режим доступа: <http://life-prog.ru>

8.6. Интернет-ресурсы

1. Академия ORACLE [Электронный ресурс] – Режим доступа: <https://academy.oracle.com/ru/>
2. Научная сеть Scipeople [Электронный ресурс] – Режим доступа: <http://scipeople.ru/>
3. Портал открытых данных [Электронный ресурс] – Режим доступа <https://data.gov.ru/>.

8.7. Методические указания по освоению дисциплины

Методические указания при работе над конспектом во время проведения лекции

В ходе лекционных занятий необходимо вести конспектирование учебного материала. Общие и утвердившиеся в практике правила и приемы конспектирования лекций:

Конспектирование лекций ведется в специально отведенной для этого тетради, каждый лист которой должен иметь поля, на которых делаются пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Необходимо записывать тему и план лекций, рекомендуемую литературу к теме. Записи разделов лекции должны иметь заголовки, подзаголовки, красные строки. Для

выделения разделов, выводов, определений, основных идей можно использовать цветные карандаши и фломастеры.

Названные в лекции ссылки на первоисточники надо пометить на полях, чтобы при самостоятельной работе найти и вписать их.

В конспекте дословно записываются определения понятий, категорий и законов. Остальное должно быть записано своими словами.

Каждому обучающемуся необходимо выработать и использовать допустимые сокращения наиболее распространенных терминов и понятий.

В конспект следует заносить всё, что преподаватель пишет на доске, а также рекомендуемые схемы, таблицы, диаграммы и т.д.

Методические указания по подготовке к практическим и лабораторным работам

Целью практических и лабораторных работ является углубление и закрепление теоретических знаний, полученных обучающимися на лекциях и в процессе самостоятельного изучения учебного материала, а, следовательно, формирование у них определенных умений и навыков.

В ходе подготовки к практическим и лабораторным работам необходимо прочитать конспект лекции, изучить основную литературу, ознакомиться с дополнительной литературой, выполнить выданные преподавателем задания. При этом учесть рекомендации преподавателя и требования программы. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы. Желательно при подготовке к практическим и лабораторным работам по дисциплине одновременно использовать несколько источников, раскрывающих заданные вопросы.

Методические указания по организации самостоятельной работы

Самостоятельная работа приводит обучающегося к получению нового знания, упорядочению и углублению имеющихся знаний, формированию у него профессиональных навыков и умений.

Самостоятельная работа выполняет ряд функций:

- развивающую;
- информационно-обучающую;
- ориентирующую и стимулирующую;
- воспитывающую;
- исследовательскую.

Виды самостоятельной работы, выполняемые в рамках курса:

1. Проработка и повторение лекционного материала
2. Подготовка к практическим занятиям
3. Подготовка к лабораторным занятиям
4. Реферат
5. Подготовка к аттестации

Обучающимся рекомендуется с самого начала освоения курса работать с литературой и предлагаемыми заданиями в форме подготовки к очередному аудиторному занятию. При этом актуализируются имеющиеся знания, а также создается база для усвоения нового материала, возникают вопросы, ответы на которые обучающийся получает в аудитории.

Можно отметить, что некоторые задания для самостоятельной работы по курсу имеют определенную специфику. При освоении курса обучающийся может пользоваться библиотекой вуза, которая в полной мере обеспечена соответствующей литературой. Значительную помощь в подготовке к очередному занятию может оказать имеющийся в учебно-методическом комплексе краткий конспект лекций. Он же может использоваться и для закрепления полученного в аудитории материала.

Методические указания по работе с литературой

Всю литературу можно разделить на учебники и учебные пособия, оригинальные научные монографические источники, научные публикации в периодической печати. Из них можно выделить литературу основную (рекомендуемую), дополнительную и литературу для углубленного изучения дисциплины.

Изучение дисциплины следует начинать с учебника, поскольку учебник – это книга, в которой изложены основы научных знаний по определенному предмету в соответствии с целями и задачами обучения, установленными программой.

При работе с литературой следует учитывать, что имеются различные виды чтения, и каждый из них используется на определенных этапах освоения материала.

Предварительное чтение направлено на выявление в тексте незнакомых терминов и поиск их значения в справочной литературе. В частности, при чтении указанной литературы необходимо подробнейшим образом анализировать понятия.

Сквозное чтение предполагает прочтение материала от начала до конца. Сквозное чтение литературы из приведенного списка дает возможность обучающемуся сформировать свод основных понятий из изучаемой области и свободно владеть ими.

Выборочное – наоборот, имеет целью поиск и отбор материала. В рамках данного курса выборочное чтение, как способ освоения содержания курса, должно использоваться при подготовке к лабораторным практикумам по соответствующим разделам.

Аналитическое чтение – это критический разбор текста с последующим его конспектированием. Освоение указанных понятий будет наиболее эффективным в том случае, если при чтении текстов обучающийся будет задавать к этим текстам вопросы. Часть из этих вопросов сформулирована в приведенном в ФОС перечне вопросов для собеседования. Перечень этих вопросов ограничен, поэтому важно не только содержание вопросов, но сам принцип освоения литературы с помощью вопросов к текстам.

Целью изучающего чтения является глубокое и всестороннее понимание учебной информации.

Есть несколько приемов изучающего чтения:

1. Чтение по алгоритму предполагает разбиение информации на блоки: название; автор; источник; основная идея текста; фактический материал; анализ текста путем сопоставления имеющихся точек зрения по рассматриваемым вопросам; новизна.

2. Прием постановки вопросов к тексту имеет следующий алгоритм:

- медленно прочитать текст, стараясь понять смысл изложенного;
- выделить ключевые слова в тексте;
- постараться понять основные идеи, подтекст и общий замысел автора.

3. Прием тезирования заключается в формулировании тезисов в виде положений, утверждений, выводов.

К этому можно добавить и иные приемы: прием реферирования, прием комментирования.

Важной составляющей любого солидного научного издания является список литературы, на которую ссылается автор. При возникновении интереса к какой-то обсуждаемой в тексте проблеме всегда есть возможность обратиться к списку относящейся к ней литературы. В этом случае вся проблема как бы разбивается на составляющие части, каждая из которых может изучаться отдельно от других. При этом важно не терять из вида общий контекст и не погружаться чрезмерно в детали, потому что таким образом можно не увидеть главного.

Методические указания к разработке и проведению проблемного обучения.

Под проблемным обучением понимается такая организация учебного процесса, которая предполагает создание под руководством учителя проблемных ситуаций и активную самостоятельную деятельность учащихся по их разрешению, в результате чего и

происходит творческое овладение предметными знаниями, умениями, навыками (ЗУН) и развитие творческих способностей.

Данный вид обучения:

1. направлен на самостоятельный поиск учащимися новых понятий и способов действий;
2. предполагает последовательное и целенаправленное выдвижение перед учащимися познавательных проблем, разрешение которых (под руководством учителя) приводит к активному усвоению новых знаний;
3. обеспечивает особый способ мышления, прочность знаний и творческое их применение в практической деятельности.

При проблемном обучении преподаватель не сообщает готовых знаний, а организует учащихся на их поиск: понятия, закономерности, теории познаются в ходе поиска, наблюдений, анализа фактов, мыслительной деятельности.

Необходимыми составляющими проблемного обучения являются следующие понятия: «проблема», «проблемная ситуация», «гипотеза», «эксперимент».

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации дисциплины требуется следующее материально-техническое обеспечение (специальные помещения):

- для проведения занятий лекционного типа
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для проведения занятий семинарского типа, практических занятий
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для проведения, текущего контроля и промежуточной аттестации
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для групповых и индивидуальных консультаций
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для самостоятельной работы:
помещение, оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Института

10. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (тьютора), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

– присутствие тьютора, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе, записывая под диктовку),

– письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,

– специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),

– индивидуальное равномерное освещение не менее 300 люкс,

– при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

– присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе, записывая под диктовку),

– обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;

– обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата:

– письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются тьютору;

– по желанию студента задания могут выполняться в устной форме.

Приложение 1

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ
ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
по дисциплине «Интернет программирование»**

1. Показатели и критерии оценки результатов освоения дисциплины

Результаты обучения	Показатели оценивания	Критерии оценивания	Процедуры оценивания
ПК-2 способность разрабатывать и адаптировать прикладное программное обеспечение			
Знает принципы разработки и адаптации прикладного ПО; принцип действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основные типы селекторов и правила расчета приоритетов правил CSS и основные принципы разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.	Знание принципов разработки и адаптации прикладного ПО; принципов действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основных принципов разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.	Полнота и правильность изложения принципов разработки и адаптации прикладного ПО; принципов действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основных принципов разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.	Устный опрос
Умеет разрабатывать и адаптировать прикладное ПО; использовать структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; использовать основные типы селекторов и	Использование умений разработки и адаптации прикладного ПО; принципов действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основных принципов разработки клиентской части веб-приложения с	Обоснованность использования умений разработки и адаптации прикладного ПО; принципов действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основных принципов разработки клиентской части веб-приложения с использованием	Практическое задание

правила расчета приоритетов правил CSS и основные принципы разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.	использованием технологий HTML5, CSS3 и JavaScript.	технологий HTML5, CSS3 и JavaScript.	
Владеет навыками разработки и адаптации прикладного ПО; использования структуры запросов протокола HTTP и способов создания и передачи документов гипертекстового языка разметки HTML; использования основных типов селекторов и правил расчета приоритетов правил CSS и основных принципов разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.	Применение навыков разработки и адаптации прикладного ПО; принципов действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основных принципов разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.	Точность применения навыков разработки и адаптации прикладного ПО; принципов действия и структуру запросов протокола HTTP и способы создания и передачи документов гипертекстового языка разметки HTML; основных принципов разработки клиентской части веб-приложения с использованием технологий HTML5, CSS3 и JavaScript.	Практическое задание
ПК-8 Способность проводить тестирование компонентов программного обеспечения			
Знает методы разработки многокомпонентных информационных систем, реализованных на различных платформах.	Знание методов разработки многокомпонентных информационных систем, реализованных на различных платформах.	Полнота и правильность изложения методов разработки многокомпонентных информационных систем, реализованных на различных платформах.	Устный опрос
Умеет разрабатывать и отлаживать эффективные алгоритмы и программы с	Использование умений разрабатывать и отлаживать эффективные алгоритмы и	Обоснованность использования умений разрабатывать и отлаживать эффективные алгоритмы и	Практическое задание

использованием современных технологий программирования.	программы с использованием современных технологий программирования.	программы с использованием современных технологий программирования.	
Владеет навыками проведения тестирования компонентов ПО ИС; навыками программирования Web-интерфейса в современных средах.	Применение навыков проведения тестирования компонентов ПО ИС; навыками программирования Web-интерфейса в современных средах.	Точность применения навыков разрабатывать и отлаживать эффективные алгоритмы и программы с использованием современных технологий программирования.	Практическое задание
ПК-2 ПК-8			Промежуточная аттестация: дифференциальный зачет

2. Методические материалы, определяющие процедуры оценивания

2.1. Методические материалы, определяющие процедуры оценивания в рамках текущего контроля успеваемости

Устные опросы проводятся во время лекций, практических занятий и возможны при проведении промежуточной аттестации в качестве дополнительного испытания при недостаточности результатов тестирования. Основные вопросы для устного опроса доводятся до сведения студентов на предыдущем занятии.

Количество вопросов определяется преподавателем.

Время проведения опроса от 10 минут до 1 академического часа.

Устные опросы строятся так, чтобы вовлечь в тему обсуждения максимальное количество обучающихся в группе, проводить параллели с уже пройденным учебным материалом данной дисциплины и смежными курсами, находить удачные примеры из современной действительности, что увеличивает эффективность усвоения материала на ассоциациях.

Критерии и шкала оценки устного опроса

Развернутый ответ студента должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях.

«отлично» ставится, если:

1) студент полно излагает материал, дает правильное определение основных понятий;

2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные;

3) излагает материал последовательно и правильно с точки зрения норм литературного языка.

«хорошо» - студент дает ответ, удовлетворяющий тем же требованиям, что и для «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.

«удовлетворительно» – студент обнаруживает знание и понимание основных положений данной темы, но:

1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил;

2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры;

3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.

«неудовлетворительно» ставится, если студент обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке, которые являются серьезным препятствием к успешному овладению последующим материалом.

Практические задания выполняются студентами на практических занятиях. Студентам необходимо выполнить практические задания, указанные преподавателем. Результаты работы сохранить в файлах. После выполнения заданий необходимо преподавателю продемонстрировать результаты работы и быть готовым ответить на вопросы и продемонстрировать выполнение отдельных пунктов заданий. Защита выполненных практических заданий осуществляется на практическом занятии.

Критерии и шкала оценки практических заданий

«отлично» ставится, если: студент самостоятельно и правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя изученные понятия.

«хорошо» ставится, если: студент самостоятельно и в основном правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя изученные понятия.

«удовлетворительно» ставится, если: студент в основном решил учебно-профессиональную задачу, допустил несущественные ошибки, слабо аргументировал свое решение, используя в основном изученные понятия.

«неудовлетворительно» ставится, если: студент не решил учебно-профессиональную задачу.

2.2 Методические материалы, определяющие процедуры оценивания в рамках промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме **дифференцированного зачета**.

Дифференцированный зачет служит формой проверки успешного выполнения студентами практических заданий и усвоения знаний на занятиях лекционного и семинарского типа.

Результаты дифференцированного зачета определяются на основании результатов текущего контроля успеваемости студента в течение периода обучения.

Критерии и шкала оценки дифференцированного зачета

Дифференцированный зачет выставляется по совокупному результату текущего контроля успеваемости студента по дисциплине:

«отлично» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «хорошо и отлично», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «хорошо и отлично» при этом совокупный средний балл оценок не ниже 4,6.

«хорошо» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «хорошо, отлично, удовлетворительно», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «хорошо, отлично, удовлетворительно», при этом совокупный средний балл оценок не ниже 3,8.

«удовлетворительно» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «удовлетворительно», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «удовлетворительно», при этом совокупный средний балл оценок ниже 3,8.

«неудовлетворительно» - не выполнены условия для получения оценки «удовлетворительно».

3. Типовые контрольные задания

Типовые задания для текущего контроля успеваемости

3.1. Типовые вопросы для устного опроса при текущем контроле

Тема 1. Принципы работы и структура Web-приложений на основе ASP.NET

- Архитектура Web-приложений
- Краткое описание архитектуры ASP.NET и .NET Framework

- Краткие итоги
- Тема 2. Основы работы в VisualStudio .NET 2005
- Начало работы с VisualStudio 2005
 - Создание нового приложения
 - Редактирование Web-документов
 - Установка точек прерывания и просмотр значений переменных
 - Пошаговое исполнение
 - Редактирование кода
 - Окно SolutionExplorer. Структура ASP.NET приложения
 - Работа со справочной системой
 - Запуск приложения и организация виртуальных каталогов приложений
 - Структура каталога приложения
 - Краткие итоги
- Тема 3. Основы языка C#
- Основные операторы языка C#
 - Типы данных. Преобразования типов
 - Массивы, перечисления, коллекции
 - Работа со строками в C#
 - Процедуры и функции
 - Классы и структуры
 - Краткие итоги
- Тема 4: Основы Web-программирования с использованием ASP.NET
- Динамическое создание элементов управления
 - Класс Page
 - Коллекция ViewState, объекты Session и Application
 - Использование строки запроса
 - Объекты Response и Request
 - Краткие итоги
- Тема 5. Принципы разработки пользовательского интерфейса интернет-приложения
- Использование элементов CheckBox, CheckBoxList, RadioButton, RadioButtonList и BulletedList
 - Использование Image, ImageMap, ImageButton
 - Использование Literal и HiddenField
 - Использование Panel
 - Использование LinkButton
 - Проверка вводимых данных
 - Краткие итоги
- Тема 6. Навигация по Web-приложению
- Карты сайта
 - Использование элементов управления TreeView, Menu и SiteMapPath
 - Навигация с использованием программирования
 - Использование MultiView и Wizard
 - Краткие итоги
- Тема 7. Использование тем при оформлении Web-приложения
- Создание и использование тем
 - Совместное использование Themes и CSS
 - Динамическое изменение темы

- Краткие итоги
- Тема 8. Использование кэширования в Web-приложениях
 - Основы кэширования в ASP.NET
 - Кэширование вывода
 - Фрагментное кэширование
 - Краткие итоги
- Тема 9. Использование баз данных в приложениях ASP.NET
 - Основные сведения о модели доступа к данным ADO.NET
 - Организация взаимодействия с БД
 - Отсоединенные наборы данных
 - Привязка и отображение данных
 - Обновление данных
 - Использование DataView для фильтрации и сортировки данных
 - Реализация трехуровневой архитектуры доступа к данным в ASP.NET
 - Краткие итоги
- Тема 10. Введение в PHP
 - История PHP
 - Возможности PHP
 - Установка и настройка ПО
 - Первая PHP-программа
- Тема 11. Основы синтаксиса PHP
 - Основной синтаксис
 - Разделение инструкций
 - Комментарии
 - Переменные, константы и операторы
 - Типы данных
 - Решение задачи
- Тема 12. Управляющие конструкции
 - Условные операторы
 - Циклы
 - Операторы передачи управления
 - Операторы включения
 - Решение задачи

3.2. Типовые практические задания

Практическое занятие 1. HTML. История создания. Структура документов

1. **Цель работы:** Изучить основные принципы создания HTML документов. Ознакомится с перечнем основных элементов HTML документов.

2. СОДЕРЖАНИЕ РАБОТЫ

2.1. HTML. ИСТОРИЯ СОЗДАНИЯ

Среди ключевых технологий, на которых держится современный Интернет, одной из важнейших и, как это ни странно, вызывающих наиболее ожесточенные споры является язык HTML, предназначенный для разметки и оформления документов World Wide Web. Необычайно интересно проследить историю развития этого языка - историю, в которой

столкнулись противоположные подходы к проблеме компьютерного представления текста и которая послужила ареной сотрудничества и противоборства крупнейших компьютерных компаний, определяющих пути развития и будущее Интернета.

Начало истории языка **HTML** было положено в 1986 году, когда Международной организацией по стандартизации (ISO) был принят стандарт ISO-8879. Он носил название **Standard Generalized Markup Language** (Стандартный Обобщённый Язык Разметки). **SGML** представлял из себя обобщенный метаязык, позволяющий строить системы логической, структурной разметки любых разновидностей текстов. При этом управляющие коды или теги не несли никакой информации о внешнем виде документа, а только задавали его логическую структуру.

Благодаря таким ограничениям, размеченный текст без труда интерпретировался любой компьютерной программой и любым устройством вывода. При этом вид и размер текста задавался исключительно настройками последних. Все это обеспечило небывалую совместимость и универсальность SGML. SGML получил широкое признание и стал активно использоваться в больших проектах. Но в целом этот язык был громоздок и труден для изучения. Главный побочный эффект SGML заключался в том, что он пытался сделать слишком много и быть чересчур гибким.

В 1989 году сотрудник Европейского института физики частиц (CERN) **Тим Бернерс-Ли** (Tim Berners-Lee) выдвинул предложение о создании Системы гипертекстовых документов. В 1990 году он назвал ее **World Wide Web** (Всемирная паутина). Одной из составляющих системы был язык гипертекстовой разметки. Его созданием Бернерс-Ли занялся в 1990 году, когда разрабатывал первый браузер — программу, позволяющую просматривать гипертекстовые документы.

В 1991 году группой Бернерс-Ли, работавшей над созданием системы передачи гипертекстовой информации через Интернет, было принято решение, выбрать SGML в качестве основы для нового языка разметки гипертекстовых документов. Этот язык был назван **HTML (Hyper Text Markup Language, "язык разметки гипертекста")**.

HTML разделял все особенности идеологии SGML, т.е. подразумевалась только логическая разметка текста. Например, в **HTML версии 1.2** (июнь 1993 г.) присутствовало около 40 тегов. И только три из них отвечали за физические параметры отображения документа (но эти теги не рекомендовались к использованию). А первым графическим браузером в то время была программа **Mosaic**, разработанная в Национальном центре суперкомпьютерных приложений. [1]

В апреле 1994 г. началась подготовка спецификации следующей версии языка - 2.0. Этим уже занимался **Консорциум W3 (W3C)**, унаследовавший от CERN власть и авторитет в мире **WWW**. В том же году было решено разбить язык HTML, на уровни. Это было сделано, чтобы обеспечить обратную совместимость версий: каждый новый уровень непременно включал в себя предыдущие. Было выделено четыре уровня:

– **Уровень 0** — обязательный для поддержки всеми браузерами. Включает в себя заголовки, ссылки, списки.

– **Уровень 1** — добавляются рисунки и элементы выделения текста (полужирным, курсивом, подчеркнутым).

– **Уровень 2** — добавляются формы, позволяющие пользователю вводить информацию.

– **Уровень 3** — добавляются таблицы, позволяющие размещать на страницах информацию табличного характера и, кроме этого, точно задавать положение объектов на странице.

Язык **HTML 2.0** (1995 г.) является, например, языком второго уровня, с его помощью нельзя верстать страницы со сложным дизайном, так как это невозможно без таблиц. [2]

Самым большим прорывом явился **HTML 3.2**. Он включил в себя такие нововведения, как теги для создания таблиц, вставки рисунков, примечаний, разметку математических формул и др. Наиболее важным нововведением явились **CSS** (таблицы стилей), которые позволяли задавать единый вид всему сайту. Они имели свою структуру, свой синтаксис и применялись вместе с HTML.

Примечательна **история HTML 3.2**. Это скорее история первых браузеров. С созданием WWW практически сразу началось его коммерческое освоение. В начале 1994 года группа разработчиков браузера Mosaic под предводительством Джеймса Кларка основала корпорацию **Netscape Communications** и вскоре выпустила первую версию коммерческого браузера **Netscape**. Спрос на него, при отсутствии альтернативы, превысил все ожидания и сделал Netscape к концу 1995 года самой быстрорастущей компанией в мировой истории.

Чтобы закрепить лидерство **Netscape Communications** вводила в HTML все новые и новые усовершенствования. И эти усовершенствования поддерживались только браузером **Netscape**. Практически все новые теги, предложенные Netscape, были направлены на улучшение внешнего вида документа и расширение возможностей его форматирования. Такая политика компании принесла ей впечатляющий успех. Девять из десяти используемых в то время браузеров были версии **Netscape Navigator**. И продавались они миллионами штук.

Компания **Microsoft** изначально не предавала серьезного значения коммерческим перспективам WWW. Однако невероятный взлет Netscape заставил Microsoft изменить свое мнение. Летом 1996 года на свет появился браузер Internet Explorer 3.0, который поддерживал почти все расширения Netscape. Одновременно с разработкой конкурентного браузера Microsoft навела порядок в мире HTML, взяв под свою опеку Консорциум W3. В итоге был создан стандарт HTML 3.2, который, по сути, всего лишь описывал большинство расширений Netscape.

К концу 1996 года практически все браузеры поддерживали HTML 3.2, и благодаря этому Web-дизайн испытал небывалый взлет. Появилась возможность проектировать и отображать на экране сложные композиции графических элементов, ничем не уступающие печатным изданиям. Это положило начало эре Web-дизайна.

В 1997 году появилась спецификация языка **HTML 4.0**. Она включала поддержку фреймов, унифицированную процедуру вставки различных объектов, поддержку каскадных таблиц стилей (CSS). Кроме того, были усовершенствованы формы и таблицы.

Версия **HTML 4.01** стала стандартом в 1999 году и господствует в мире WWW уже более 10 лет. Но ее очень скоро потеснит **HTML 5.0** – новая версия языка гипертекстовой разметки. Над разработкой **HTML 5.0** трудится рабочая группа W3C. В эту группу входят представители таких компаний, как **Microsoft, Opera, Mozilla, Google, IBM, Apple** и другие. Новый стандарт позволит более эффективно управлять мультимедийным содержимым. Также будет улучшена совместимость с новыми языками web-программирования. И наконец, появятся новые теги, расширяющие возможности web-дизайна. [3]

2.2. СТРУКТУРА ДОКУМЕНТА

HTML — теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — *тегами*. Элементы могут быть *пустыми*, то есть не содержащими никакого текста и других данных (например, тег перевода строки
). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь *атрибуты*, определяющие какие-либо их свойства (например, размер шрифта для элемента font). Атрибуты указываются в открывающем теге. Вот примеры фрагментов HTML-документа:

Выделенный текст между двумя тегами.

Пример пустого элемента:

Здесь элемент содержит атрибут href, то есть гиперссылку на сайт ВУЗа.

Регистр, в котором набрано имя элемента и имена атрибутов, в HTML значения не имеет. Элементы могут быть вложенными. Например, следующий код:

```
<b>
  Этот текст будет полужирным,
  <i>a этот - ещё и курсивным</i>
</b>
```

Кроме элементов, в HTML-документах есть и *сущности* — «специальные символы». Сущности начинаются с символа амперсанда и имеют вид *&имя*; или *&#NNNN*; где *NNNN* — код символа в Юникоде в десятичной системе счисления.

Например, *©*; — знак авторского права (©). Как правило, сущности используются для представления символов, отсутствующих в кодировке документа, или же для представления «специальных» символов: *&*; — амперсанда (&), *<*; — символа «меньше» (<) и *>*; — символа «больше» (>), которые некорректно записывать «обычным» образом, из-за их особого значения в HTML. Список основных сущностей указан в таблице 1.1. [2]

Таблица 1.1 – HTML – сущности.

Символ	Сущность
Пробелы	
	<i>&nbsp</i> ;
	<i>&ensp</i> ;
	<i>&emsp</i> ;
Спецсимволы (HTML)	
"	<i>&quot</i> ;
&	<i>&amp</i> ;
<	<i>&lt</i> ;
>	<i>&gt</i> ;
Œ	<i>&OElig</i> ;
œ	<i>&oelig</i> ;
Š	<i>&Scaron</i> ;
š	<i>&scaron</i> ;
ÿ	<i>&Yuml</i> ;
^	<i>&circ</i> ;
~	<i>&tilde</i> ;
–	<i>&ndash</i> ;
—	<i>&mdash</i> ;

СИМВОЛ	СУЩНОСТЬ
‘	‘
’	’
‚	‚
“	“
”	”
„	„
†	†
‡	‡
‰	‰
‹	‹
›	›
€	€
¡	¡
¢	¢
£	£
¤	¤
¥	¥
¦	¦
§	§
¨	¨
©	©
ª	ª
«	«
»	»
¬	¬
	­
®	®
–	¯
°	°
±	±
²	²
³	³
´	´
µ	µ
¶	¶
·	·
¸	¸
¹	¹
º	º
¼	¼
½	½
¾	¾

Символ	Сущность
¿	¿
Стрелочки, указатели, посылатели	
←	←
↑	↑
→	→
↓	↓
↔	↔
↶	↵
⇐	⇐
⇑	⇑
⇒	⇒
⇓	⇓
⇔	⇔

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, должен начинаться со строки объявления версии HTML `<!DOCTYPE...>`, которая обычно выглядит примерно так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Эта строка поможет браузеру определить, как правильно интерпретировать данный документ. В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 4.01. Как видно из примера, самый короткий html-документ состоит буквально из одной строки.

На самом деле одной строкой дело не обходится. После объявления версии и типа документа необходимо обозначить его начало и конец. Это делается с помощью тега-контейнера `<HTML>`. Необходимо отметить, что любой HTML-документ должен содержать элемент HTML. [1]

Затем, между тегами `<HTML>` и `</HTML>` следует разместить заголовок и тело документа. Вот как должен выглядеть ваш базовый HTML-файл перед началом работы:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Заголовок документа</TITLE>
  </HEAD>
  <BODY>
    Текст документа
  </BODY>
</HTML>
```

Из схемы видно, что документ состоит из двух основных блоков – "заголовка" и "тела документа". Заголовок определяется с помощью элемента HEAD, а тело – элементом BODY.

Заголовок содержит "техническую" информацию о документе, хотя чаще всего используется только для обозначения его названия.

В теле документа находится все то, что отображается на странице: текст, картинки, таблицы. Соответственно, большинство наших HTML-экспериментов будет проводиться в пространстве между тегами <BODY> и </BODY>.

Значимым преимуществом HTML является то, что документы можно создавать в любом текстовом редакторе.

Словарь основных HTML элементов можно найти в приложении А.

Типовые задания для промежуточной аттестации

3.3 Типовые контрольные вопросы для устного опроса на дифференцированном зачете

1. Принципы работы и структура Web-приложений на основе ASP.NET
2. Архитектура Web-приложений
3. Краткое описание архитектуры ASP.NET и .NET Framework
4. Основы работы в VisualStudio .NET 2005
5. Начало работы с VisualStudio 2005
6. Создание нового приложения
7. Редактирование Web-документов
8. Установка точек прерывания и просмотр значений переменных
9. Пошаговое исполнение
10. Редактирование кода
11. Окно SolutionExplorer. Структура ASP.NET приложения
12. Работа со справочной системой
13. Запуск приложения и организация виртуальных каталогов приложений
14. Структура каталога приложения
15. Основы языка C#
16. Основные операторы языка C#
17. Типы данных. Преобразования типов
18. Массивы, перечисления, коллекции
19. Работа со строками в C#
20. Процедуры и функции
21. Классы и структуры
22. Основы Web-программирования с использованием ASP.NET
23. Динамическое создание элементов управления
24. Класс Page
25. Коллекция ViewState, объекты Session и Application
26. Использование строки запроса
27. Объекты Response и Request
28. Использование кэширования в Web-приложениях
29. Основы кэширования в ASP.NET
30. Кэширование вывода
31. Фрагментное кэширование
32. Использование баз данных в приложениях ASP.NET
33. Основные сведения о модели доступа к данным ADO.NET
34. Организация взаимодействия с БД

35. Отсоединенные наборы данных
36. Привязка и отображение данных
37. Обновление данных
38. Использование DataView для фильтрации и сортировки данных
39. Реализация трехуровневой архитектуры доступа к данным в ASP.NET
40. Введение в RНР
41. История RНР
42. Возможности RНР
43. Установка и настройка ПО
44. Первая RНР-программа
45. Основы синтаксиса RНР
46. Основной синтаксис
47. Разделение инструкций
48. Переменные, константы и операторы
49. Типы данных
50. Управляющие конструкции
51. Условные операторы
52. Циклы
53. Операторы передачи управления
54. Операторы включения