



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

«СОГЛАСОВАНО»
Руководитель ОП



подпись

Чеботарев А.Ю.
ФИО

«УТВЕРЖДАЮ»

Заведующий кафедрой информатики,
математического и компьютерного
моделирования, протокол



подпись

Чеботарев А.Ю.
ФИО
«11» июля 2019 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Web-программирование

Направление подготовки 01.03.02 Прикладная математика и информатика
(Системное программирование)

Форма подготовки очная

курс 2 семестр 3

лекции 00 час.

практические занятия 00 час.

лабораторные работы 34 час.

в том числе с использованием МАО лек. 0 /пр. 0 /лаб. 18 час.

всего часов аудиторной нагрузки 34 час.

в том числе с использованием МАО 18 час.

самостоятельная работа 38 час.

в том числе на подготовку к экзамену 36 час.

контрольные работы (количество) 2.

курсовая работа / курсовой проект не предусмотрены

зачет не предусмотрены

экзамен 3 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 10 января 2018 г. № 9

Рабочая учебная программа обсуждена на заседании кафедры информатики, математического и компьютерного моделирования, протокол № 18 от «09» июля 2019 г.

Заведующий кафедрой информатики, математического и компьютерного моделирования
Чеботарев А.Ю.

Составитель: ст. п. А.А. Сущенко

Оборотная сторона титульного листа РПД

II. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «__» _____ 202_ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «__» _____ 202_ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «__» _____ 202_ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

1. Цели и задачи освоения дисциплины:

Цель: является освоение современных интернет-технологий и сопутствующих областей знаний, методов и средств создания web-ресурсов, их продвижения и применения в различных видах деятельности

Задачи:

- Дать целостное представление о возможностях и структуре глобальной сети Internet.
- Дать представление о развитии и применении Internet технологий в профессиональной деятельности.
- Изучить методы и средства разработки web-приложений;
- Сформировать навыки практической работы по созданию сайтов.
- Развить логическое и алгоритмическое мышление.

Профессиональные компетенции выпускников и индикаторы их достижения:

Задача профессиональной деятельности	Объекты или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции	Основание (ПС, анализ иных требований, предъявляемых к выпускникам)
Тип задач профессиональной деятельности: организационно-управленческий				
		ПК-7 способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы	ПК-7.1 Знает основы составления планов работы с учетом ресурсов. ПК-7.2 Умеет составлять и контролировать план выполняемой работы. ПК-7.3 Владеет способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы.	
		ПК-8 способность к формированию технической отчетной документации и разработке технических документов	ПК-8.1 Знает техническую и отчетную документацию. ПК-8.2 Умеет формировать техническую и отчетную документацию и	

			разрабатывать технические документы. ПК-8.3 Владеет навыками формирования технической отчетной документации.	
--	--	--	---	--

Для формирования вышеуказанных компетенций в рамках дисциплины «Web-программирование» применяются следующие методы интерактивного обучения: лекция-беседа, метод автоматизированного обучения.

При выполнении различных видов работ используются следующие технологии:

- Проблемное обучение – стимулирование обучающихся к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы.
- Контекстное обучение – мотивация студентов к усвоению знаний путём выявления связей между конкретным знанием и его применением.
- Обучение на основе опыта – активизация познавательной деятельности студентов бакалавриата за счёт ассоциации и собственного опыта с предметом обучения.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Не предусмотрено учебным планом.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА И САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Лабораторная работа 1. Основы HTML.(2 час.)

Лабораторная работа 2. Использование DHTML для разработки сайта.(2 час.)

Лабораторная работа 3. Основы CSS.(2 час.)

Лабораторная работа 4. Использование языка JavaScript для программирования приложений на стороне клиента.(4час.)

Лабораторная работа 5. Создание сайта с применением JavaScript и модели DOM.(4 час.)

Лабораторная работа 6. Создание сайта с использованием библиотеки JQUERY. (4 час.)

Лабораторная работа 7. Программное рисование, использование CANVAS HTML5. (4 час.)

Лабораторная работа 8. Программное рисование и библиотека Mo tools
(4 час.)

Лабораторная работа 9. Основы PHP.(4 час.)

Лабораторная работа 10. Разработка клиент-серверного приложения.(4 час.)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	Неделя 1,2	Лабораторная работа «Основы HTML», самостоятельные задания	4 часа	лаб. работа №1
2	Неделя 3,4	Лабораторная работа «Использование DHTML для разработки сайта», самостоятельные задания	4 часа	лаб. работа №2
3	Неделя 5,6	Лабораторная работа «Основы CSS», самостоятельные задания	4 часа	лаб. работа №3
4	Неделя 7,8	Подготовка к тесту по разделу «Инструменты создания сайтов»	4 часа	лаб. работа №4
5	Неделя 9,10	Лабораторная работа «Использование языка JavaScript для программирования приложений на стороне клиента», самостоятельные задания	4 часа	лаб. работа №5
6	Неделя 11,12	Лабораторная работа «Создание сайта с применением JavaScript и модели DOM», самостоятельные задания	4 часа	лаб. работа №6
7	Неделя 13,14	Программирование приложений на стороне клиента», самостоятельные задания	4 часа	лаб. работа №7
8	Неделя 15,16	Лабораторная работа «Создание сайта с использованием библиотеки JQUERY», самостоятельные задания	4 часа	лаб. работа №8
9	Неделя 17	Подготовка к опросу по разделу «Программирование на стороне клиента»	4 часа	Защита отчетов
10	18 неделя	Подготовка к зачету	2 часа	Зачет

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине включает в себя характеристику заданий для самостоятельной работы; требования к представлению и оформлению

результатов самостоятельной работы; критерии оценки выполнения самостоятельной работы.

Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

Самостоятельная работа студентов состоит из подготовки практических работ, работы над рекомендованной литературой. При подготовке практических работ необходимо сначала прочитать основные понятия по теме. При выполнении задания нужно сначала понять, что требуется в задаче, какой теоретический материал нужно использовать, наметить план решения задачи. Рекомендуется использовать методические указания и материалы по курсу, электронные пособия, имеющиеся на сервере Школы естественных наук, библиотеке ДВФУ и в сети Интернет. При подготовке к экзамену нужно освоить теорию: разобрать определения всех понятий и методов, рассмотреть примеры и самостоятельно решить несколько типовых задач из каждой темы. При решении задач всегда необходимо комментировать свои действия и не забывать о содержательной интерпретации.

Требования к представлению и оформлению результатов самостоятельной работы

Результатом самостоятельной работы являются отчеты по практическим работам на выбранную по желанию студента тематику. В процессе подготовки отчетов к практическим работам у студентов развиваются навыки составления письменной документации и систематизации имеющихся знаний.

При составлении отчетов рекомендуется придерживаться следующей структуры:

1. Постановка задачи;
2. Математическая постановка задачи;
3. Описание метода решения;
4. Описание алгоритма метода;
5. Спецификация используемых функций и типов данных;
6. Описание тестов, на которых программа проходит проверку;
7. Результаты и демонстрация эксперимента.

Критерии оценки выполнения самостоятельной работы

Отчет по лабораторной работе должен полностью удовлетворять условию задачи. В случае некачественно выполненных отчетов (не соответствующих заявленным требованиям) результирующий балл за работу может быть снижен. Студент должен продемонстрировать отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком

и терминологией. Наличие письменного доклада, презентации и отчета о проделанной работе, является условием получения зачета.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Инструменты создания сайтов	ПК-7 ПК-8	знает	Тест (ПР-1)	Вопросы 1-24 Задание, тип 1
			умеет	Лабораторная работа (ПР-6)	
			владеет	Контрольная работа (ПР-2)	
2	Программирование на стороне клиента	ПК-7 ПК-8	знает	Собеседование (УО-1)	Вопросы 1-14 Задание, тип 2
			умеет	Лабораторная работа (ПР-6)	
			владеет	Контрольная работа (ПР-2)	
3	Программирование на стороне сервера	ПК-7 ПК-8	знает	Собеседование (УО-1)	Вопросы 15-30 Задание, тип 3
			умеет	Лабораторная работа (ПР-6)	
			владеет	Расчетно-графическая работа (ПР-12)	

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Боженюк А. В. Интеллектуальные интернет-технологии : учебник для вузов / А. В. Боженюк, Э. М. Котов, А. А. Целых. - Ростов-на-Дону : Феникс , 2010. – 382 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:419137&theme=FEFU>
2. Бурлакова Н. Н. Создание гипертекстовой среды : учебное пособие для вузов . ч. 1 . Статическое кодирование / Н. Н. Бурлакова. - Владивосток: Изд-во Дальневосточного федерального университета , 2012. – 85 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:681561&theme=FEFU>
3. Дронов В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. — СПб.: БХВ-Петербург, 2011. — 414 с.: ил. <http://znanium.com/bookread.php?book=351455>
4. Одиночкина, С.В. Web-программирование PHP [Электронный ресурс] : учебно-методическое пособие. — Электрон. дан. — СПб. : НИУ ИТМО (Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики), 2012. — 79 с.
http://e.lanbook.com/books/element.php?pl1_id=43562

5. Зудилова, Т.В. Web-программирование HTML [Электронный ресурс] : учебно-методическое пособие / Т.В. Зудилова, М.Л. Бурков. — Электрон. дан. — Спб. : НИУ ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики), 2012. — 70 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=40724

Дополнительная литература

(печатные и электронные издания)

1. Дунаев В.Я. Основы Web-дизайна. Учебник. БХВ-Петербург – 2013. - 479 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:702309&theme=FEFU>

2. Дунаев В.Я. HTML, скрипты и стили. Учебник. – СПб: БХВ-Петербург, 2012. - 485 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:692681&theme=FEFU>

3. Евсеев, Д.А. Web-дизайн в примерах и задачах [Текст]: учеб.пособие / Д.А. Евсеев, В.Р. Трофимов; Под. ред. В.В. Трофимова. – М.: КНОРУС, 2010. – 272 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:382740&theme=FEFU>

4. Прохоренок Н. HTML, JavaScript, PHP и MySQL джентльменский набор Web-Мастера. – СПб: БХВ-Петербург, 2008. - 622 с.
<https://lib.dvfu.ru:8443/lib/item?id=chamo:380700&theme=FEFU>

5. Гоше Х.Д. HTML5. Для профессионалов / Хуан Диего Гоше. - Санкт-Петербург: Питер, 2015. – 559 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:780190&theme=FEFU>

6. Гуриков С.Р. Интернет-технологии: Учебное пособие / С.Р. Гуриков. - М.: Форум: НИЦ ИНФРА-М, 2015. - 184 с.
<http://znanium.com/go.php?id=488074>

Перечень ресурсов информационно-телекоммуникационной сети

«Интернет»

1. Справочник Notepad++ - свободно распространяемый текстовый редактор с открытым исходным кодом – Режим доступа: <https://notepad-plus-plus.org/>;

2. Справочник по HTML. – Режим доступа: <https://htmlbook.ru/html>;

3. CSS справочник. – Режим доступа: <https://css.manual.ru>;

4. Справочник Javascript. – Режим доступа: <https://Javascript.ru>.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

1. Описание последовательности действий студента («сценарий изучения дисциплины»).

Для успешного изучения дисциплины «WEB-программирование» студенту необходимо:

1. При подготовке к лекции изучить конспект предыдущей лекции, а также при необходимости теоретический материал, представленный в системе BlackBoard. Далее ознакомиться с материалом следующей лекции (также через систему BlackBoard), вникнуть в суть изучаемой проблемы, подготовить вопросы.

2. На лекционном занятии тщательно конспектировать теоретический материал, участвовать в обсуждении, задавать вопросы.

3. При подготовке к лабораторным занятиям на основе материалов, представленных в системе BlackBoard, сначала понять задание лабораторной работы, найти теоретический материал, необходимый для работы, изучить алгоритм реализации задания, сформулировать вопросы преподавателю.

4. На лабораторном занятии сначала задать вопросы преподавателю по методике выполнения работы, затем выполнить задание «по образцу». Отправить преподавателю через систему BlackBoard выполненное задание на проверку, ответить на вопросы преподавателя. Прежде, чем приступить к выполнению самостоятельных заданий, обдумать алгоритм их реализации, задать вопросы преподавателю по сути заданий, спланировать их выполнение.

5. Самостоятельные задания лабораторной работы можно выполнять как на аудиторном занятии, так и самостоятельно во внеаудиторное время. При этом результат их реализации необходимо проверять через систему BlackBoard и только после этого отправлять преподавателю на проверку.

6. В течение недели выбрать время для работы со специальной литературой в библиотеке и для занятий на компьютере.

7. Самостоятельную работу организовывать в соответствии с графиком выполнения самостоятельной работы, приведенном в приложении 1.

2. Рекомендации по использованию материалов учебно-методического комплекса. Рекомендуется использовать методические указания и материалы по курсу «WEB-программирование», размещенные в системе BlackBoard, идентификатор курса FU50709-02.03.01-IT-01.

3. Рекомендации по работе с литературой. Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекций, использованию учебно-методического комплекса, представленного в

системе BlackBoard, изучаются и книги из списка основной и дополнительной литературы. Литературу по курсу можно изучать в библиотеке, брать книгу на дом или читать ее на компьютере (если это электронный ресурс). Полезно использовать несколько учебников, однако желательно придерживаться рекомендации преподавателя по выбору книг по каждому разделу. Не рекомендуется «заучивать» материал, желательно добиться понимания изучаемой темы дисциплины, а затем использовать изученный материал для реализации программ и/или сайтов. Кроме того, очень полезно выявить тенденции развития той или иной технологии разработки, выделить для себя направления дальнейшего изучения материала, для достижения более продвинутого уровня изучения дисциплины.

4. Рекомендации по подготовке к экзамену и зачету. Успешная подготовка к экзамену и зачету включает, с одной стороны, добросовестную работу в течение семестра, выполнение всех заданий преподавателя, а с другой – правильная организация процесса непосредственной подготовки. При подготовке к экзамену необходимо освоить теорию: разобрать определения всех понятий, повторить синтаксис и семантику изучаемых языков программирования, технологию разработки web-сайтов. Затем рассмотреть примеры и самостоятельно реализовать задания из каждой темы. При этом, если задания формулируются студентом самостоятельно, – достигается более продвинутый уровень изучения дисциплины.

Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

Расчетно-графическая работа

В расчетно-графической работе требуется создать web-приложение, используя стандартные инструменты разработки web-сайтов на стороне сервера и клиента. Приложение должно быть предназначено для организации работы с пользователем и решения некоторых задач выбранной предметной области. Результат работы – работающий на локальном сервере web-сайт и документация по его разработке.

Основные этапы разработки web-приложения

1. Разработка технического задания. Составление документации, детально описывающей структурную и поведенческую модели проекта. Простым языком: описываем коротко, на техническом языке, как будет выглядеть и как будет работать сайт или программа.

2. Создание дизайна и навигации. Детальная прорисовка и определение позиционирования, типовых структурных элементов проекта. Например: создание эскиза страницы сайта, разработка логотипа, выбор мест

расположения элементов навигации, оформление шрифтов и заголовков и многое другое.

3. Вёрстка. Описание типовых структурных элементов проекта на языке разметки, в соответствии с заданной спецификацией. Проще говоря: то, что разработано на 2-м этапе, необходимо привести к рабочему виду, используя разметки HTML и CSS.

4. Программирование. Описание структурной и поведенческой моделей проекта на языках программирования.

5. Тестирование и отладка. Выявление синтаксических и логических ошибок проекта и последующее их устранение.

Критерии оценки выполнения самостоятельной работы

Самостоятельная работа студентов включает расчетно-графическую работу, подготовку к устным опросам, тестирование и лабораторные работы.

Устные опросы

Устные опросы проводятся преподавателем по завершению изучения каждого раздела. Вопросы и задания приведены в приложении 2. Для подготовки используется основная и дополнительная литература по дисциплине «WEB-программирование», а также информация, размещенная в LMS BlackBoard.

Вопросы, возникающие в процессе подготовки, студент может задать преподавателю либо на консультациях, либо через специальное средство LMS BlackBoard.

Тестирование

Тестирование осуществляется на занятии через систему BlackBoard. Для подготовки тестов используются пробные тесты, размещенные в системе BlackBoard. Эти тесты не оцениваются преподавателем и служат элементом самоконтроля. Оба типа теста формируются на основе одной базы вопросов.

Лабораторные работы

Выполняются в соответствии с методическими указаниями к выполнению лабораторных работ, размещенных в системе BlackBoard. Выполненные работы отправляются преподавателю на проверку через LMS BlackBoard.

Требования к представлению и оформлению результатов самостоятельной работы

Результатом самостоятельной работы студентов являются:

1. Расчетно-графическая работа – web-сайт, реализованный стандартными средствами разработки web-приложений и оформленный в соответствии со стандартами. Созданные сайты обязательно отсылаются преподавателю через систему BlackBoard. Защита расчетно-графического

задания осуществляется только после того, как приложение проверено и оценено преподавателем в системе BlackBoard.

2. Тесты самоконтроля, выполненные в системе BlackBoard. Все попытки, выполненные пользователем, сохраняются. К основному тесту студент допускается только после прохождения на зачетном уровне данного вида тестов.

3. Самостоятельные задания к лабораторным работам, реализованные необходимыми средствами разработки web-приложений и оформленные в соответствии со стандартами. Реализованные задания обязательно отсылаются преподавателю через систему BlackBoard. Защита лабораторной работы осуществляется только после того, как задание проверено и оценено преподавателем в системе BlackBoard.

Правила оформления HTML/CSS кода

1. Протокол.

Не указывайте протокол при включении ресурсов на страницу. Опускайте название протокола (**http:**, **https:**) в ссылках на картинки или другие медиа-ресурсы, файлы стилей или скрипты, конечно, если эти файлы доступны по обоим протоколам. Отсутствие протокола делает ссылку относительной, что предотвращает смешивание ресурсов из разных протоколов и незначительно уменьшает размер файлов.

Не рекомендуется:

```
<script src="http://www.google.com/js/gweb/analytics/autotrack.js">
</script>
```

Рекомендуется:

```
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

2. Общее форматирование

2.1. Отступы

Всегда используйте для отступа два пробела. Не используйте табуляцию и не смешивайте табуляцию с пробелами.

Рекомендуется:

```
<ul>
  <li>
  <li>
</ul>
```

```
.example {
  color: blue;
}
```

2.2. Регистр.

Всегда пишите в нижнем регистре. Весь код должен быть написан в нижнем регистре: это относится к названиям элементов, названиям атрибутов, значениям атрибутов (кроме текста/**CDATA**), селекторам, свойствам и их значениям (кроме текста).

Не рекомендуется:

```
<A HREF="/">Домой </A>
```

Рекомендуется:

```

```

2.3. Пробелы в конце строки

Убирайте пробелы в конце строки. Пробелы в конце строк не обязательны и усложняют использование diff.

3. Общие мета правила

3.1. Кодировка

Используйте UTF-8 (без BOM). Убедитесь, что ваш редактор использует кодировку UTF-8 без метки порядка байтов (BOM). Указывайте кодировку в HTML шаблонах и документах с помощью

```
<meta charset="utf-8">.
```

Опускайте кодировку для css-файлов: для них UTF-8 задана по умолчанию.

3.2. Комментарии

По возможности поясняйте свой код, где это необходимо. Используйте комментарии, чтобы пояснить свой код: что он делает, за что отвечает, и почему используется выбранное решение.

4. Правила оформления HTML

4.1. Тип документа

Используйте HTML5. HTML5 (HTML синтаксис) рекомендуется для всех html-документов: **<!DOCTYPE html>**.

4.2. Валидность HTML

По возможности используйте валидный HTML. Используйте валидный HTML код, кроме случаев, когда использование не позволяет достичь размера файла, необходимого для нужного уровня производительности. Используйте такие инструменты как [W3C HTML validator \(англ.\)](#) чтобы проверить валидность кода.

4.3. Семантика

Используйте HTML так, как это было задумано. Используйте элементы (Иногда неверно называемые “тегами”) по назначению: заголовки для заголовков, **p** для абзацев, **a** для ссылок и т.д. Это облегчает чтение, редактирование и поддержку кода.

Не рекомендуется:

```
<div onclick="goToRecommendations();">All recommendations </div>
```

Рекомендуется:

```
<a href="recommendations/"> All recommendations </a>
```

4.4. Альтернатива для мультимедиа

Всегда указывайте альтернативное содержимое для мультимедиа.

Постарайтесь указать альтернативное содержимое для мультимедиа: например для картинок, видео или анимаций, заданных с помощью **canvas**. Для картинок это осмысленный альтернативный текст (**alt**), а для видео и аудио расшифровка текста и подписи если это возможно.

Не рекомендуется:

```

```

Рекомендуется:

```

```

4.5. Разделение ответственности

Разделяйте структуру, оформление и поведение. Держите структуру (разметка), оформление (стили) и поведение (скрипты) отдельно и постарайтесь свести взаимодействие между ними к минимуму. Убедитесь, что документы и шаблоны содержат только HTML, и что HTML служит только для задания структуры документа. Весь код, отвечающий за оформление, перенесите в файлы стилей, а код отвечающий за поведение — в скрипты. Старайтесь сократить их пересечения к минимуму, включая в шаблоны минимальное количество файлов стилей и скриптов.

Отделение структуры от представления и поведения помогает облегчить поддержку кода. Изменение шаблонов и HTML-документов всегда занимает больше времени чем изменение файлов стилей или скриптов.

4.6. Атрибут *type*

Не указывайте атрибут type при подключении стилей и скриптов в документ. Не используйте атрибут **type** при подключении стилей (кроме вариантов когда используется что-то кроме CSS) и скриптов (кроме вариантов когда это не JavaScript). Указывать атрибут **type** в данном случае не обязательно потому что HTML5 использует **text/css** и **text/javascript** по умолчанию. Это будет работать даже в старых браузерах.

Не рекомендуется:

```
<link rel="stylesheet" href="//www.google.com/css/maia.css" type="text/css">
```

Рекомендуется:

```
<link rel="stylesheet" href="//www.google.com/css/maia.css">
```

5. Правила форматирования HTML

5.1.Форматирование

Выделяйте новую строку для каждого блочного, табличного или списочного элемента и ставьте отступы для каждого дочернего элемента. Независимо от стилей заданных для элемента (CSS позволяет изменить поведение элемента с помощью свойства **display**), переносите каждый блочный или табличный элемент на новую строку.

Также ставьте отступы для всех элементов вложенных в блочный или табличный элемент.

6. Правила оформления CSS

6.1 Идентификаторы и названия классов

Используйте шаблонные или имеющие смысл имена классов и идентификаторы. Вместо использования шифров, или описания внешнего вида элемента, попробуйте в имени класса или идентификатора выразить смысл его создания, или дайте ему шаблонное имя...

Рекомендуется выбирать имена, отражающие сущность класса, потому что их проще понять и, скорее всего, не понадобится менять в будущем.

6.2. Названия идентификаторов и классов

Для идентификаторов и классов используйте настолько длинные имена, насколько нужно, но настолько короткие, насколько возможно. Попробуйте сформулировать, что именно должен делать данный элемент, при этом будьте кратки насколько возможно. Такое использование классов и идентификаторов вносит свой вклад в облегчение понимания и увеличение эффективности кода.

6.3. Селекторы типа

Избегайте использование имен классов или идентификаторов с селекторами типа (тега) элемента.

6.4. 0 и единицы измерения

Не указывайте единицы измерения для нулевых значений. Не указывайте единицы измерения для нулевых значений если на это нет причины.

6.5. 0 в целой части дроби

Не ставьте "0" в целой части дробных чисел. Не ставьте **0** в целой части в значениях между -1 и 1.

6.6. Кавычки в ссылках

Не используйте кавычки в ссылках. Не используйте кавычки ("" , ") с **url()**.

6.7. Шестнадцатеричные названия цветов

Используйте трехсимвольную шестнадцатеричную запись где это возможно. Трехсимвольная шестнадцатеричная запись для цветов короче и занимает меньше места.

Не рекомендуется:

```
color:#eebbcc;
```

Рекомендуется:

```
color:#ebc;
```

6.8. Разделители в классах и идентификаторах

Разделяйте слова в идентификаторах и именах классов с помощью дефиса или подчеркивания. Не используйте ничего, кроме дефиса или подчеркивания, для соединения слов и сокращений в селекторах, чтобы повысить удобство чтения и легкость понимания кода.

7. Правила форматирования CSS

7.1. Упорядочивание объявлений

Сортируйте объявления по алфавиту. Задавайте объявления в алфавитном порядке, чтобы получить согласованный код, с которым легко работать. При сортировке игнорируйте браузерные префиксы. При этом, если для одного свойства используются несколько браузерных префиксов, они также должны быть отсортированы (например **-moz** должен быть перед **-webkit**)

7.2. Отступы в блоках.

Всегда ставьте отступы для содержимого блоков. Всегда ставьте отступы для любого блочного содержимого, Например для правил внутри правил или объявлений, чтобы отобразить иерархию и облегчить понимание кода.

7.3. После объявлений

Ставьте точку с запятой после каждого объявления. После каждого объявления ставьте точку с запятой для согласованности кода и облегчения добавления новых свойств.

7.4. После названий свойств

Используйте пробелы после двоеточий в объявлениях. Всегда используйте один пробел после двоеточия (но не до) в объявлениях, для порядка в коде.

7.5. Отделение селектора и объявления

Отделяйте селекторы и объявления переносом строки. Начинайте каждый селектор или объявление с новой строки.

Не рекомендуется:

```
a:focus, a:active {  
  position: relative; top: 1px;  
}
```

Рекомендуется:

```
h1,
```


h2,
h3 {
font-weight: normal;
line-height: 1.2;
}

7.6. Разделение правил

Разделяйте правила пустой строкой. Всегда ставьте пустую строку между правилами.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Аудиторные занятия по дисциплине включают лекции и лабораторные работы. Для проведения лекционных занятий необходима аудитория со следующим оборудованием:

- мультимедийный проектор Optima EX542I – 1 шт;
- аудио усилитель QVC RMX 850 – 1 шт;
- колонки – 1 шт;
- ноутбук;
- ИБП – 1 шт;
- настенный экран;
- микрофон – 1 шт.

Лабораторные работы проводятся в компьютерном классе, в котором должно быть установлено:

- 15 моноблоков Lenovo C360G-i34164G500UDK с установленным стандартным программным обеспечением, доступом к сети Интернет.
- мультимедийный проектор OptimaEX542I – 1 шт;
- настенный экран;
- аудио усилитель QVC RMX 850 – 1 шт;
- колонки – 1 шт;
- ИБП – 1 шт;
- микрофон – 1 шт.
- документ-камера.

VIII. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине созданы фонды оценочных средств:

Категория (группа) профессиональных компетенций	Категория (группа) общепрофессиональных компетенций	Наименование оценочного средства
<p>ПК-7 способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы</p> <p>ПК-8 способность к формированию технической отчетной документации и разработке технических документов</p>	<p>ПК-7.1 Знает основы составления планов работы с учетом ресурсов. ПК-7.2 Умеет составлять и контролировать план выполняемой работы. ПК-7.3 Владеет способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы</p> <p>ПК-8.1 Знает техническую и отчетную документацию. ПК-8.2 Умеет формировать техническую и отчетную документацию и разрабатывать технические документы. ПК-8.3 Владеет навыками формирования технической отчетной документации</p>	<p>Доклад и презентация</p>
<p>ПК-7 способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы</p> <p>ПК-8 способность к формированию технической отчетной документации и разработке технических документов</p>	<p>ПК-7.1 Знает основы составления планов работы с учетом ресурсов. ПК-7.2 Умеет составлять и контролировать план выполняемой работы. ПК-7.3 Владеет способностью составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы</p> <p>ПК-8.1 Знает техническую и отчетную документацию. ПК-8.2 Умеет формировать техническую и отчетную документацию и разрабатывать технические документы. ПК-8.3 Владеет навыками формирования технической отчетной документации</p>	<p>Отчет по программе</p>

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
				текущий контроль	промежуточная аттестация
1	Инструменты создания сайтов	ПК-7 ПК-8	знает	Тест (ПР-1)	Вопросы 1-24
			умеет	Лабораторная работа (ПР-6)	Задание, тип 1
			владеет	Контрольная работа (ПР-2)	
2	Программирование на стороне клиента	ПК-7 ПК-8	знает	Собеседование (УО-1)	Вопросы 1-14
			умеет	Лабораторная работа (ПР-6)	Задание, тип 2
			владеет	Контрольная работа (ПР-2)	
3	Программирование на стороне сервера	ПК-7 ПК-8	знает	Собеседование (УО-1)	Вопросы 15-30
			умеет	Лабораторная работа (ПР-6)	Задание, тип 3
			владеет	Расчетно-графическая работа (ПР-12)	

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Оценочные средства для промежуточной аттестации

Перечень типовых вопросов к зачету

1. Введение в Web- технологии, основные понятия и определения (URL, Internet, WWW, HTTP, FTP, ...)
2. Доменные имена, IP - адрес, DNS, Web- сайты.
3. Понятие технологии клиент- сервер, Web-сервер, сервер БД, почтовый сервер, файловый сервер.
4. Характеристика программного обеспечения, используемого при создании Web -страниц.
5. Введение в язык HTML, элементы языка HTML.
6. HTML, простейшее форматирование текста, управление выравниванием текста.
7. HTML, оформление абзацев, заголовков, горизонтальная линейка.
8. HTML, управление шрифтом.
9. HTML, управление цветом, дополнительные варианты оформления.
10. Локальные гиперссылки в рамках Web-страницы, формирование гиперссылок в пределах сайта, якоря.
11. HTML, маркированные списки, смешанные списки.
12. HTML, нумерованные списки, смешанные списки.
13. Гиперссылки в пределах сайта, организация переходов средствами гиперссылок.
14. Работа с таблицами в языке HTML.

15. Управление шириной столбцов, высотой строк, объединение ячеек в таблицах средствами тегов языка HTML.
16. Формирование фоновых изображений на Web-страницах.
17. Графика на Web-страницах, вставка изображений.
18. CSS. Встраивание таблиц стилей в документ. Свойства CSS для шрифтов и текста. Свойства CSS для полей, отступов и границ. Свойства CSS для фона и цвета.
19. CSS. Каскадные таблицы стилей (в отдельных файлах). Визуальные фильтры, производимый эффект и параметры.
20. CSS. Каскадные таблицы стилей: стили и селекторы.
21. CSS. Позиционирование: свободное, абсолютное, фиксированное и относительное размещение.
22. CSS. Селекторы: селекторы классов и идентификаторов, селекторы атрибутов и псевдоселекторы.
23. CSS. Форматирование страниц с помощью языка иерархических стилевых спецификаций, использование каскадных таблиц стилей.
24. CSS. Цвет. Фоновые изображения. Объединение свойств фона.

Вопросы к зачету\экзамену

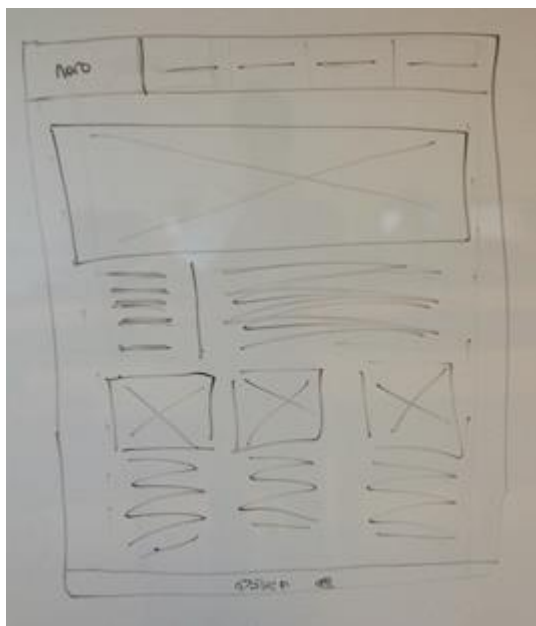
1. JavaScript. доступ к значениям элементов форм. Объект frames и динамическое создание фрейма.
2. JavaScript. объект Array. Обобщенные объекты.
3. JavaScript. объект images и его свойства. Динамическое изменение изображений, создание простейших анимаций, подгрузка рисунков.
4. JavaScript. объект Math, его свойства и методы. Объект String, его свойства и методы.
5. JavaScript. свойства объекта document.body, метод document.write() и объект history. Объект window и параметры метода window.open().
6. Характеристика библиотеки JQuery
7. Ссылки на элементы интернет-страницы.
8. Установка свойств элементов и стилей.
9. Характеристика библиотеки Mootools
10. Реализация основных концепций объектно-ориентированного программирования средствами Mootools
11. Элемент canvas. Его свойство и характеристики
12. Рисование 2d изображений.
13. Преобразование изображений.
14. Использование таймера.
15. Серверный язык PHP. Синтаксис. Включение PHP-сценария в HTML-документ.

16. Серверный язык PHP. Организация ветвлений.
17. Серверный язык PHP. Понятие класса. Основные компоненты класса.
18. Серверный язык PHP. Абстрагирование, инкапсуляция, модульность и иерархия
19. Серверный язык PHP. Хранение и использование данных пользователя. Способы хранения. Хранение данных в файлах.
20. Серверный язык PHP. Хранение данных в файлах. Открытие файла. Функция fopen(). Режимы файла. Чтение файла. Запись в файл.
21. Серверный язык PHP. Организация счетчика посещений.
22. Серверный язык PHP. Обработка форм.
23. Серверный язык PHP. Массивы, наследование.
24. Базы данных MySQL. Базовая архитектура баз данных для web. Транзакция базы данных для web.
25. Базы данных MySQL. Доступ к базе данных из web. Выбор базы данных.
26. Базы данных MySQL. Доступ к базе данных из web. Структура процедуры доступа. Установка соединения.
27. Базы данных MySQL. Доступ к базе данных из web. Фильтрация входных данных.
28. Базы данных MySQL. Доступ к базе данных из web. Выполнение запроса к базе данных.
29. Базы данных MySQL. Доступ к базе данных из web. Получение результатов запроса.
30. Базы данных MySQL. Доступ к базе данных из web. Организация поиска в базе данных.

Перечень типовых задач

Задание тип 1, по разделу «Инструменты создания сайтов»

Создать сайт, используя HTML и CSS следующей структуры:



Задание тип 2, по разделу «Программирование на стороне клиента»

На языке JavaScript, используя DOM модель и библиотеку JQuery реализовать галерею, в которой реализована смена картинки через заданный интервал времени, выбор и загрузка картинки в отдельный блок

Задание тип 3, по разделу «Программирование на стороне сервера»

Разработать базу данных, используя MySQL, для предметной области «Библиотека»:

1. «В некоторой библиотеке хранятся книги. Каждая книга имеет название, относится к одному жанру, имеет одного или нескольких авторов, издана в определенном издательстве. В библиотеке может быть несколько экземпляров одной книги. Каждый экземпляр имеет уникальный шифр.
2. Пользователями библиотеки являются читатели, о которых сохраняется информация об их фамилии, адресе и телефоне. Каждый читатель может взять несколько экземпляров книг, в базе должны храниться сведения о взятой книге, дате получения и возврата».

Программно создать таблицы, занести данные и реализовать выборку информации по запросам пользователей.

Принцип составления экзаменационного билета заданий к зачету

Первый вопрос является теоретическим и предназначен для оценивания порогового уровня освоения дисциплины. Второй вопрос – представляет собой задание, которое необходимо выполнить на компьютере. Это может быть создание небольшого сайта, решение некоторой задачи на языках JavaScript и PHP, проект структуры приложения пользователя.

Критерии выставления оценки студенту на экзамене\зачете по дисциплине «WEB-программирование»

Баллы	Оценка	Требования к сформированным компетенциям
-------	--------	--

(рейтингов ой оценки)	экзамена/зачета (стандартная)	
86-100	«отлично»/ «зачтено»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил материал по различным Интернет-технологиям, в том числе и технологиям разработки сайтов, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет проектировать web-приложения, используя инструменты web-разработки, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, свободно использует компьютер для реализации web-приложений, выбирает эффективную методологию при проектировании и реализации, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач, связанных с созданием приложений в области профессиональной деятельности.
76-85	«хорошо»/ «зачтено»	Оценка «хорошо» выставляется студенту, если он твердо знает материал по различным Интернет-технологиям в том числе и технологиям разработки сайтов, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет языки HTML, CSS, JavaScript, PHP при решении практических вопросов и задач, связанных с созданием web-приложений в области профессиональной деятельности, владеет необходимыми навыками и приемами использования инструментов разработки сайтов.
61-75	«удовлетворительн о»/ «зачтено»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала по интернет-технологиям, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении синтаксических и семантических особенностей языков HTML, CSS, JavaScript, PHP, испытывает затруднения при проектировании и реализации web-приложения.
0-60	«неудовлетворител ьно»/«не зачтено»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала по основам интернет-технологий, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, связанные с созданием web-приложений. Оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине

.Оценочные средства для текущей аттестации

Вопросы для собеседований по дисциплине «WEB- программирование»

Раздел «Программирование на стороне клиента»

1. JavaScript. Язык клиентских сценариев. Объект navigator, свойства браузера, объект document и свойства документа.
2. JavaScript. Арифметические, логические и операторы сравнения.
3. JavaScript. Ввод и вывод данных средствами JavaScript. Типы данных.
4. JavaScript. Динамический HTML: программирование окон и фреймов, свойства документа, работа с формами.
5. JavaScript. Операторы цикла и условного перехода.
6. JavaScript. Переменные и их область действия. Операторы языка JavaScript.
7. JavaScript. Символы-разделители и переводы строк. Комментарии. Литералы. Идентификаторы.

8. JavaScript. События мыши. Обработка нажатия и отпускания клавиши мыши.
9. JavaScript. События мыши. Обработка движения указателя мыши.
10. JavaScript. События клавиатуры. Обработка ввода символа.
11. JavaScript. События клавиатуры. Обработка нажатия/отпускания клавиши.
12. JavaScript. События, связанные с компонентами HTML-страницы.
13. JavaScript. Сценарии обработки событий. Объект EVENT и его атрибуты.
14. JavaScript. Числа и работа с числами. Логические значения в JavaScript.
15. JavaScript. гиперссылки (метки) и массив объектов links (anchors). Объект Date и его методы.
16. JavaScript. Использование объектной модели документа.
17. JavaScript и библиотека JQuery.
18. JavaScript и библиотека Mootools.
19. Объекты в JavaScript. Иерархия объектов.
20. Новые элементы web-страниц в HTML5.
21. Структура страницы HTML 5.
22. Программное рисование
23. Особенности использования canvas
24. Трансформация изображений
25. Реализация движения

Раздел «Программирование на стороне сервера»

1. Web сервер и логика его работы.
2. Протокол HTTP. Структура запросов и ответов.
3. CGI. Способы передачи данных. Запоминание состояния.
4. HTML формы (описание и использование).
5. PHP. Особенности языка.
6. PHP. Основные конструкции программирования
7. PHP. Выражение
8. PHP. Простые типы данных
9. PHP. Описание и использование составных типов данных
10. PHP. Процедуры и функции.
11. PHP. Обработка входных данных.
12. PHP. Работа с GET и POST.
13. PHP. Доступ к базам данных.
14. СУБД MySQL. Характеристика и особенности.
15. Язык SQL. Создание и удаление баз данных и таблиц в MySQL.
16. Типы данных в MySQL.

17. Ключи, индексы, автоинкрементирование. Изменение структуры таблиц.

18. MySQL и PHP. Запросы на добавление, изменение и удаление данных.

19. MySQL и PHP. Запросы на извлечение данных. Объединения. Псевдонимы, группировка и упорядочивание.

Критерии оценки:

✓ 100-85 баллов выставляется студенту, если его ответ показывает прочные знания основных положений изучаемого раздела дисциплины, отличается глубиной и полнотой раскрытия темы; студент показывает свободное владение терминологическим аппаратом; умение объяснять синтаксис и семантику языков HTML, CSS, JavaScript и PHP, делать выводы и обобщения, давать аргументированные ответы, приводить примеры интернет-технологий; обучающийся свободно владеет монологической речью, логичностью и последовательностью ответа; умеет алгоритмически описывать проблему и данные из выбранной предметной области.

✓ 85-76 баллов выставляется студенту, если его ответ, обнаруживает прочные знания основных положений изучаемого раздела дисциплины, отличается глубиной и полнотой раскрытия темы; обучающийся показывает владение терминологическим аппаратом; умение объяснять синтаксис и семантику языков HTML, CSS, JavaScript и PHP, делать выводы и обобщения, давать аргументированные ответы; студент демонстрирует свободное владение монологической речью, логичностью и последовательностью ответа; умение алгоритмически описывать проблему и данные из выбранной предметной области. Однако допускается одна - две неточности в ответе.

✓ 75-61 балл выставляется студенту, если его ответ, свидетельствует в основном о знании основных положений изучаемого раздела дисциплины, отличается недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками объяснения синтаксис и семантику языков HTML, CSS, JavaScript и PHP, недостаточным умением давать аргументированные ответы и применять WEB-программирование; студент недостаточно владеет монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; студент демонстрирует неумение алгоритмически описывать проблему данные из выбранной предметной области.

✓ 60-50 баллов выставляется студенту, если его ответ, обнаруживает незнание процессов основных положений изучаемого раздела дисциплины, отличается неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками использования языков HTML, CSS,

JavaScript и PHP; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; студент демонстрирует неумение алгоритмически описывать выбранную предметную область.

Тестовые задания по дисциплине «WEB-программирование»

Каждому студенту формируется индивидуальный тест, в который входит 20 вопросов, выбранных случайным образом из списка заданий. В тесте реализованы следующие типы тестовых заданий:

- выбор одного варианта ответа;
- выбор нескольких вариантов ответа;
- ввод числового значения.

Типовых заданий теста по разделу «Основы структурного программирования»

1. С помощью какого тега можно объявить версию языка HTML?
 - a. <head> и </head>
 - b. <html> и </html>
 - c. <!doctype>
 - d. <body> и </body>
2. Какие теги не имеют закрывающего?
 - a. <tr>
 - b. <hr>
 - c.
 - d.
 - e. <head>
3. Какой размер в пунктах будет у текста
<p>Пример текста</p>,
если на странице задан следующий стиль?
body { font-size: 24px; }
p { font-size: 50%; }

Критерии оценки теста, состоящего из 20 вопросов

- ✓ 14-20 баллов – считается, что тест пройден.
- ✓ 0-13 баллов – тест не засчитывается

Комплекты заданий для выполнения расчетно-графической работы по дисциплине «WEB-программирование»

Расчетно-графическая работа предполагает реализацию web-приложения, используя стандартные инструменты web-разработки. Для создания приложения должна быть выбрана некоторая предметная область.

При разработке необходимо выполнить следующие этапы: разработка технического задания, создание дизайна и навигации, верстка, программирование, отладка и тестирование.

Типовое задание

Разработать web-приложение «Интернет магазин», которое позволяет:

- просматривать товары с различной степенью детализации;
- выполнять поиск товаров по разным критериям;
- сравнивать между собой два или несколько товаров;
- наполнять «корзину» выбранными для покупки товарами;
- корректировать список товаров в корзине;
- оформлять покупку;
- выполнять авторизацию пользователя.

Разрабатываемое приложение должно быть реализовано на основе клиент-серверной технологии, включать несколько страниц, удобные средства навигации.

Критерии оценки расчетно-графической работы

✓ 10-8 баллов выставляется студенту, если студент полностью выполнил расчётно-графическое задание. Фактических ошибок, связанных с пониманием проблемы, нет; выполнены все этапы создания web-приложения; семантических и синтаксических ошибок в программах нет; выдержаны правила оформления HTML/CSS кода. При защите студент отвечает на все вопросы преподавателя.

✓ 7-6 баллов – работа выполнена полностью; есть незначительные погрешности в реализации отдельных компонент приложения, программах или в организации интерфейса с пользователем; выдержаны правила оформления HTML/CSS кода. При защите студент отвечает на все вопросы преподавателя.

✓ 5-4 балла – работа выполнена полностью, есть погрешности в реализации отдельных компонент приложения, в программах или в организации интерфейса с пользователем, связанные с непониманием предметной области. При защите студент не отвечает на 1-2 вопроса преподавателя.

✓ 1-3 балла – работа выполнена не полностью. Допущены ошибки в реализации отдельных компонент приложения, в программах или в организации интерфейса с пользователем, связанные с непониманием предметной области или неумением использования инструментов web-разработки. При защите студент не отвечает на более, чем на 2 вопроса преподавателя.

Лабораторные работы по дисциплине «WEB-программирование»

Структура каждой лабораторной работы следующая:

1. Реализация web-страницы или ее фрагмента в качестве образца рассматриваемой технологии.
2. Задания для самостоятельной работы.

Все лабораторные работы можно проверить через специальное средство, реализованное в системе BlackBoard. Выполненные лабораторные работы необходимо защитить, но предварительно они должны быть отправлены преподавателю на проверку через систему BlackBoard.

Типовая лабораторная работа на тему «HTML и CSS»

1. Создать web-страницу *Index.html* и сохранить ее в папке **Lab3**.

```
<HTML>
  <HEAD>
    <TITLE> Cascading Style Sheets </TITLE>
    <link rel="stylesheet" type="text/css" href="main.css">
  </HEAD>
  <BODY>
    <div id="content">
      <div id="menu">
        <ul>
          <li><a class = "active" href="#index.html">CSS </a></li>
          <li><a href="#">Синтаксис CSS </a></li>
          <li><a href="#">Вставка CSS</a></li>
          <li><a href="#">CSS текст</a></li>
          <li><a href="#">CSS шрифт</a></li>
          <li><a href="#">Блоки</a></li>
        </ul>
      </div>
      <div id="text">
        <p> CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.
        </p>
        <p >
          
          Основным понятием CSS является стиль – т. е. набор правил оформления и форматирования, который может быть применен к различным элементам страницы. Для присвоения какому-либо элементу определенных характеристик необходимо один раз описать этот элемент и определить это описание как стиль, а в дальнейшем просто указывать, что элемент получить свойства выбранного стиля.
        </p>
      </div>
    </div>
  </BODY>
</HTML>
```

2. Создать файл *main.css* и сохранить его в папке **Lab3**.

```
html, body {
  background: #164B69;
  font-size: 14px;
  font-family: arial;
```

```

margin: 0;
padding: 0;
}
p{
text-align : justify;
text-indent : 30px;
word-spacing : 5px;
line-height: 2.0;
}
#content {
width: 700px;
min-height: 700px;
background: #E3E3E3;
margin : 30px;
padding: 20px;
/* center this div */
margin-left: auto;
margin-right: auto;
}
#menu {
text-align: center; /* центрирование меню */
font-weight: bold;
}
#menu ul {
display: inline-block; /* необходимо, чтобы центрировать меню, в противном
случае UL тэг не будет реагировать на свойство
text-align*/
list-style-type: none; /*ul без маркера перед каждым элементом*/
margin: 0;
padding: 0;
}
#menu li {
float: left; /* чтобы меню стало горизонтальным */
}
#menu a {
display: block; /* используется, чтобы padding работал, как положено*/
padding: 10px 20px 10px 20px;
color: #000;
text-decoration: none; /* ссылки без подчеркивания*/
}
/* изменение цвета меню, когда на него наводится курсор */
#menu a:hover {
background: #DCDCDC;
}
#menu .active{
background: #DCDCDC;
}
#logo {
float:right;
margin : 10px;
}

```

3. Html-страница в браузере должна выглядеть следующим образом (без видео):



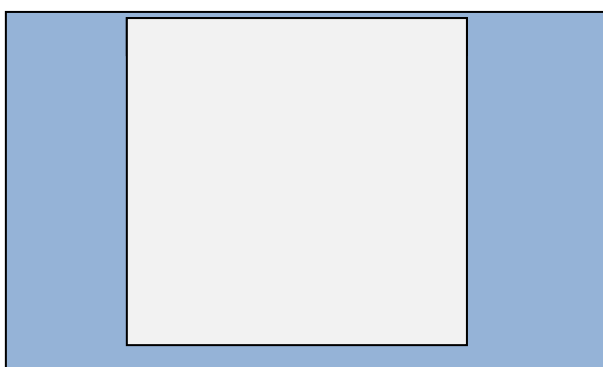
4. Вставить видео с YouTube:

- Перейти по ссылке <http://www.youtube.com/watch?v=iu0ztxdsFis> ;
- Выбрать вкладку «*Поделиться*»;
- Щелкнуть по вкладке «*Сгенерировать HTML- код*»;
- скопировать html-код и вставить его в *Index.html*;
- изменить атрибуты *width* и *height*
- добавить css-стиль в файл *main.css*:

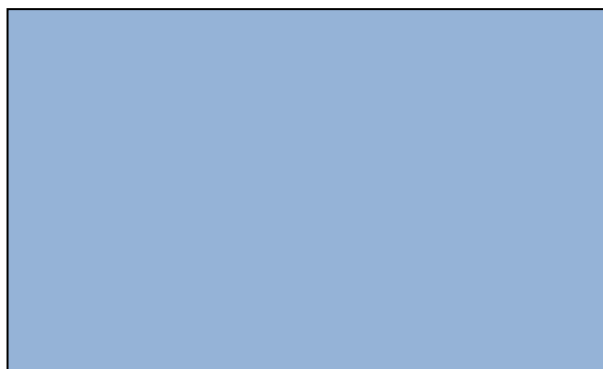
```
iframe {  
margin-left : 200px;  
width : 420px;  
height : 315px;  
}
```

Задания для самостоятельной работы

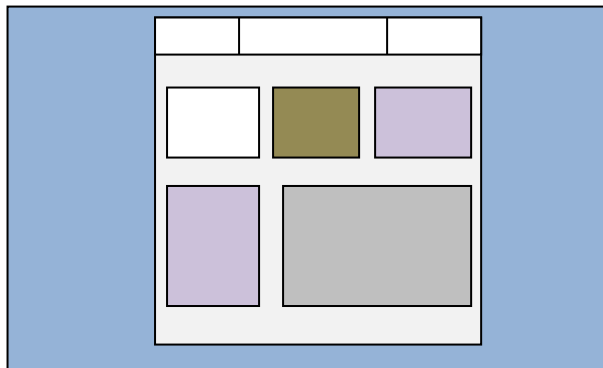
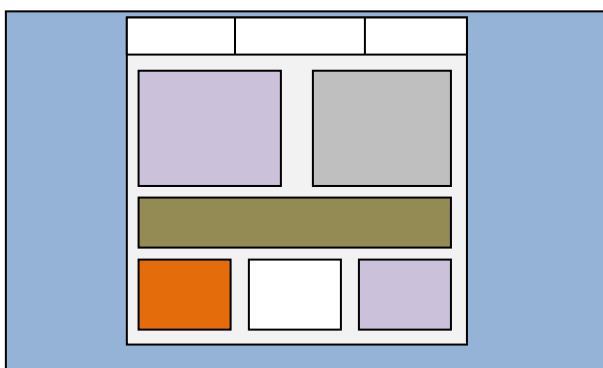
1. Создать страницу "Синтаксис CSS" и реализовать переходы между этими двумя страницами.
2. Создать новую страницу (любую из предложенных в меню), используя один из следующих макетов:



3



4



Критерии оценки лабораторной работы

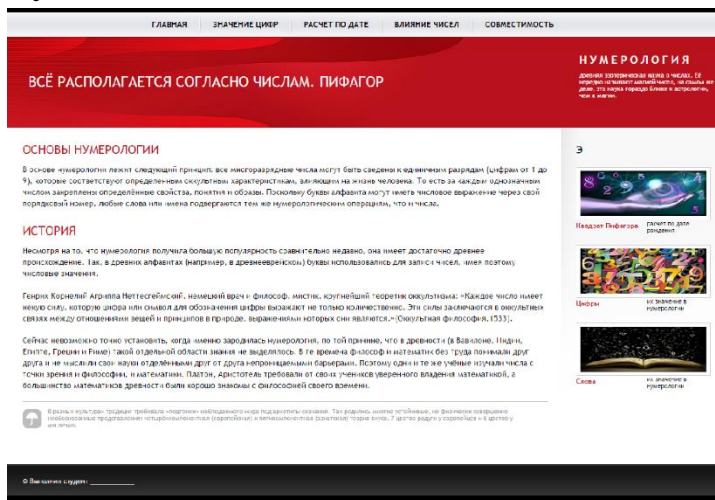
- ✓ 10-8 баллов выставляется студенту, если он выполнил все задания лабораторной работы, в том числе и самостоятельные. Фактических ошибок, связанных с пониманием проблемы, нет; семантических и синтаксических ошибок в программах нет; все инструменты web-разработки используются верно. При защите студент отвечает на все вопросы преподавателя.
- ✓ 7-6 баллов – работа выполнена полностью; студент выполнил все предложенные в лабораторной работе задания, одно самостоятельное задание реализовано не для всех исходных данных или есть погрешности в коде; все инструменты web-разработки используются верно. При защите студент отвечает на все вопросы преподавателя.
- ✓ 5-4 балла – работа выполнена полностью. Два самостоятельных задания реализованы не для всех входных данных или есть значительные погрешности в коде; часть инструментов web-разработки используются неверно. При защите студент не отвечает на 1-2 вопроса преподавателя.
- ✓ 1-3 балла – работа выполнена не полностью. Самостоятельные задания не выполнены или студент демонстрирует слабое владение инструментами web-разработки. При защите студент не отвечает более, чем на 2 вопроса преподавателя.

Комплекты заданий для контрольных работы по дисциплине «WEB-программирование»

Раздел «Инструменты создания сайтов»

Типовой вариант

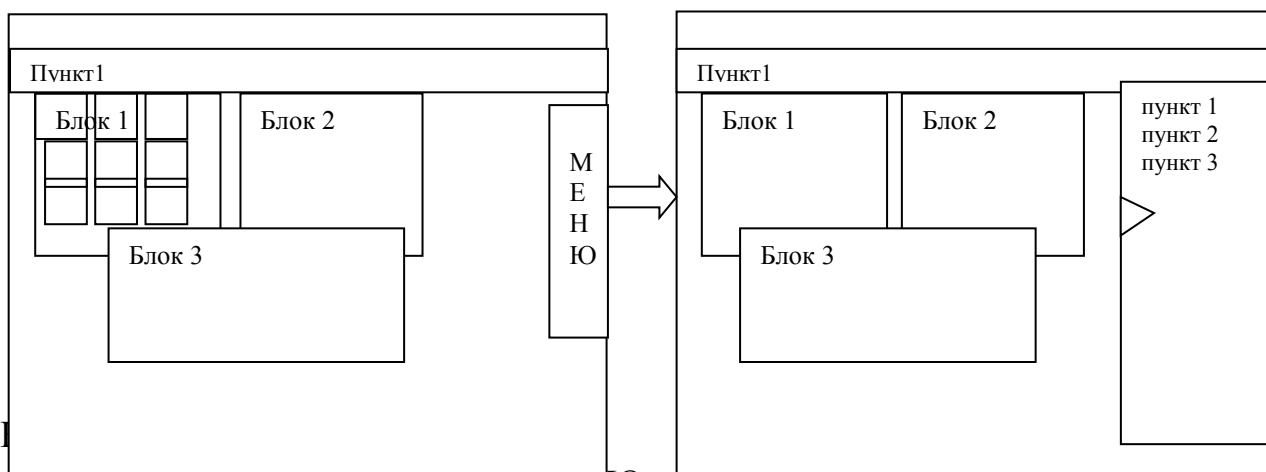
Используя HTML и CSS создать страницы *Главная* и *Значение цифр*.
Необходимые рисунки и тексты скачать из системы BlackBoard.



Раздел «Программирование на стороне клиента»

Типовой вариант

Реализовать следующую страницу, используя HTML, CSS, JavaScript
библиотеку JQuery :



- справа – по щелчку на олок МЕНЮ раскрывается меню с пунктами, по щелчку по кнопке - меню сворачивается.

- Блок 1: Картинки расположены в 3 строки и три столбца, по щелчку на любую – она выводится в увеличенном размере поверх всех блоков страницы. Закрытие картинки осуществляется по щелчку по ней.

- Блок 2: используется для размещения формы, которая включает поле для ввода, многострочное текстовое поле для вывода, кнопки очистить и запустить. Реализовать следующую задачу:

Ввести n- размерность массива. Заполнить массив случайным образом, вывести его в поле вывода по 4 элемента в строке. Вывести в поле

вывода сумму элементов стоящих на четных местах и произведение элементов, стоящих на нечетных.

- Блок 3: Текстовый блок, в котором реализовать смену цвета последовательную смену цвета фона через некоторое количество миллисекунд (количество цветов – не менее 6).

Критерии оценки контрольной работы

✓ 10-8 баллов выставляется студенту, если студент выполнил контрольную работу полностью. Фактических ошибок, связанных с пониманием заданий, нет; семантических и синтаксических ошибок в программах нет; все элементы на странице отформатированы верно; все инструменты web-разработки используются верно. При защите студент отвечает на все вопросы преподавателя.

✓ 7-6 баллов – работа выполнена полностью; студент выполнил все задания контрольной, но есть погрешности в коде или в форматировании элементов; все инструменты web-разработки используются верно. При защите студент отвечает на все вопросы преподавателя.

✓ 5-4 балла – работа выполнена полностью; однако есть существенные погрешности в коде или в форматировании элементов; часть инструментов web-разработки используются неверно. При защите студент не отвечает на 1-2 вопроса преподавателя.

✓ 1-3 балла – работа выполнена не полностью. Большинство заданий выполнены неверно или студент демонстрирует слабое владение инструментами web-разработки. При защите студент не отвечает более, чем на 2 вопроса преподавателя.