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

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

«СОГЛАСОВАНО»
Руководитель ОП
_____ Артемьева И.Л.

«УТВЕРЖДАЮ»
И.о. директора департамента
_____ Смагин С.В.
«15» июля 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Сетевые и интернет технологии
Направление подготовки 09.03.04 Программная инженерия
(Программная инженерия)
Форма подготовки очная

курс 3, 4 семестр 5, 6, 7
лекции 50 час.
практические занятия 00 час.
лабораторные работы 50 час.
в том числе с использованием МАО лек. 0 /пр. 0 /лаб. 34 час.
всего часов аудиторной нагрузки 100 час.
в том числе с использованием МАО 34 час.
самостоятельная работа 152 час.
в том числе на подготовку к экзамену 27 час.
контрольные работы (количество) не предусмотрены
курсовая работа / курсовой проект не предусмотрены
зачет 5, 6 семестр
экзамен 7 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 09.03.04 Программная инженерия утвержденного приказом Министерства образования и науки РФ от 19 сентября 2017 г. № 920 (с изменениями и дополнениями)

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 12.1 от « 25 » декабря 2019 г.

И.о. директора департамента программной инженерии и искусственного интеллекта Смагин С.В.

Составители: заведующая кафедрой прикладной математики, механики, управления и программного обеспечения Артемьева И.Л., д.т.н., профессор, Лось Р.П., старший преподаватель, Селезнев Т.Э., ассистент

Владивосток
2021

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании кафедры прикладной математики, механики, управления и программного обеспечения:

Протокол от «09» июля 2021 г. № 7.1

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

II. Рабочая программа пересмотрена на заседании департамента программной инженерии и искусственного интеллекта:

Протокол от «17» сентября 2021 г. № 9.1

И.о. директора департамента _____ Смагин С.В.
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20___ г. № _____

Директор департамента _____ _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20___ г. № _____

Директор департамента _____ _____
(подпись) (И.О. Фамилия)

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины – изучение спектра высокоуровневых интернет технологий, таких как популярные фреймворки и библиотеки, ознакомление с основными используемыми в них архитектурными подходами; усвоение и закрепление основных приемов, методов и принципов работы при создании программ для Интернет, усвоение навыков использования языков PHP и Java.

Задачи дисциплины:

1. Изучить и осмыслить основные определения, закономерности. Освоить базовые инструментальные средства по данной дисциплине.
2. Изучить язык предметной области, грамотно пользоваться необходимой терминологией.
3. Научиться оценивать корректность постановки задач данной предметной области, изучить корректные постановки классических задач.
4. Освоить методику построения алгоритма и проведения его анализа.
5. Изучить основные методики и подходы к разработке и проектирование web-приложений, освоить фундаментальные принципы верстки и шаблонизации.

Для освоения дисциплины необходимы базовые знания о языках программирования, практические навыки кодирования, теоретические знания и практические навыки в области реляционных СУБД, базовые теоретические знания в области компьютерных сетей, сетевых протоколах и уровнях передачи данных (модель ISO/OSI).

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы.

Общепрофессиональные компетенции выпускников и индикаторы их достижения:

Наименование категории (группы) общепрофессиональных компетенций	**Код и наименование общепрофессиональной компетенции**	**Код и наименование индикатора достижения общепрофессиональной компетенции**
	ОПК-3. Способен решать стандартные задачи профессиональной деятельности на	ОПК-3.1. Знает Современные средства автоматизации разработки интернет приложений.

	основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.2. Умеет использовать средства автоматизации разработки интернет приложений. ОПК-3.3. Владеет методами разработки и оценки качества интернет приложений.

Профессиональные компетенции выпускников и индикаторы их достижения:

Задача профессиональной деятельности	Объект 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: производственно-технологический				
Проведение работ по инсталляции программного обеспечения автоматизированных систем и загрузки баз данных; настройка параметров ИС и тестирование результатов настройки; ведение технической документации; техническое сопровождение ИС в процессе эксплуатации; применение Web технологий при	Программное обеспечение	ПК-9. Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных	ПК-9.1. Знает особенности проектирования интернет приложений ПК-9.2. Умеет использовать существующие средства создания интернет систем ПК-9.3. Владеет методами выбора подходящих инструментальных средств для разработки интернет приложения	06.028 Системный программист 06.022 Системный аналитик 06.004 Специалист по тестированию в области информационных технологий 06.001 Программист
		ПК-10. Владение навыками использования различных технологий разработки программного обеспечения	ПК-10.1. Знает особенности создания интернет приложений для разных классов операционных систем ПК-10.2. Умеет программировать настройку интерфейса интернет	

реализации удаленного доступа в системах клиент – сервер и распределенных вычислений			приложений для разных классов операционных систем ПК-10.3. Владеет методами проверки работоспособности создаваемых интернет систем для разных классов операционных систем	

Для формирования вышеуказанных компетенций в рамках дисциплины «Сетевые и интернет технологии» применяются следующие методы активного/ интерактивного обучения: метод проектов, дискуссия.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Лекционный материал (50 час.)

Раздел 1 (2 часа) Основы WWW, HTML, HTTP.

Тема 1. Основы World Wide Web (WWW). Основы HTML. Каскадные таблицы стилей. Протокол HTTP. Динамический HTML. DOM и клиентские скрипты. Общий шлюзовый интерфейс (CGI). Расширяемый язык разметки XML. Технологии на основе XML.

Раздел 2 (2 часа) Веб-сервер. Установка, настройка, принцип работы.

Тема 2. Модули веб-сервера. ISAPI и apache modules. Установка Apache 1.3.29 под Windows XP. Установка PHP 4.3.4 под Windows. Установка PHP под Linux. Дистрибутив Денвер. Дистрибутив XAMPP

Раздел 3 (10 часов) PHP.

Тема 3. (3 час) Введение в PHP. История PHP. Возможности PHP. Установка и настройка ПО. Первая PHP-программа. Основной синтаксис. Разделение инструкций. Комментарии. Переменные, константы и операторы. Переменные. Константы. Операторы. Типы данных. Тип boolean (булев или логический тип). Тип integer (целые). Тип float (числа с плавающей точкой). Тип string (строки). Одинарные кавычки. Двойные кавычки. Heredoc. Тип array (массив). Определение при помощи array(). Определение с помощью синтаксиса квадратных скобок. Тип object (объекты). Тип resource (ресурсы). Тип Null. Функции в PHP. Объекты и классы в PHP.

Тема 4. (1 час) Управляющие конструкции. Условные операторы. Оператор if. Оператор else. Оператор elseif. Альтернативный синтаксис. Оператор switch.
Циклы: While, do... while, For, foreach. Операторы передачи управления: Brea k, continue. Операторы включения: include, require. Решение задачи

Тема 5. (1 час) Обработка запросов с помощью PHP. Основы клиент-серверных технологий. Протокол HTTP и способы передачи данных на сервер. Форма запроса клиента. Методы. Использование HTML-форм для передачи данных на сервер. Для метода GET. Для метода POST. Обработка запросов с помощью PHP. Пример обработки запроса с помощью PHP

Тема 6. (1 час) Работа с массивами данных. Массивы. Операции с массивами.

Функция count. Функция in_array. Функция array_search. Функция array_keys. Функция array_unique. Сортировка массивов. Сортировка массива по ключам. Сортировка с помощью функции, заданной пользователем. Применение функции ко всем элементам массива. Выделение подмассива. Функция array_slice. Функция array_chunk. Сумма элементов массива

Тема 7. (1 час) Работа со строками. Строки. Поиск элемента в строке. Выделение подстроки. Функция strstr. Функция substr. Замена вхождения подстроки. Функция str_replace. Функция substr_replace. Разделение и соединение строки. Строки, содержащие html-код

Тема 8. (2 часа) Работа с файловой системой. Создание файла. Функция fopen. Закрытие соединения с файлом. Запись данных в файл. Функция fwrite. Чтение данных из файла. Функция fread. Функция fgets. Функция fgetss. Функция fgetc. Функция readfile. Функция file. Функция file_get_contents. Проверка существования файла. Функция file_exists. Функция is_writable. Функция is_readable. Удаление файла. Загрузка файла на сервер

Раздел 4 (2 час) Работа с СУБД в PHP.

Тема 9. (1 час) Взаимодействие PHP и MySQL. Построение интерфейса для добавления информации. Установка соединения. Выбор базы данных. Получение списка полей таблицы. Отображение списка полей в html-форму. Запись данных в базу данных. Отображение данных, хранящихся в MySQL

Тема 10. (0.5 часа) Авторизация доступа с помощью сессий. Авторизация доступа. Механизм сессий. Настройка сессий. Работа с сессиями. Создание сессии. Регистрация переменных сессии. Удаление переменных сессии. Безопасность

Тема 11. (0.5 часа) Регулярные выражения. Понятие регулярного выражения. Регулярные выражения в PHP. Синтаксис регулярных выражений. Подвыражения (подшаблоны). Повторения (квантификаторы). Модификаторы PCRE. Регулярные выражения для "продвинутых". Обратная ссылка. Утверждения. Условные подвыражения

Раздел 5 Веб-фреймворки (8 час.).

Тема 12. Назначение, основные принципы устройства популярных фреймворков. DRY. Паттерны MVC, ORM. URL dispatching. Шаблонирование и кэширование.

Раздел 6 Безопасность интернет ПО. (8 час.)

Тема 13. Интернет-специфика безопасности. Виды атак. [D]DoS, Code/SQL инъекции, XSS-уязвимости, phishing. Типичные ошибки при написании кода, методы защиты.

Раздел 7 Оптимизация нагруженных интернет проектов. (8 час.)

Тема 14. Упаковка стилей и библиотек. Оптимизация работы с БД. Кэширование, виды и стратегии кэширования. Nginx, его преимущества.

Раздел 8 Жизненный цикл современного интернет проекта (8 час.)

Тема 15. Разделение frontend/backend разработки, их специфика. Команда проекта. Идеология «вечной беты». Итеративный подход к разработке.

Раздел 9 (2 часа) Перспективы развития веб-технологий.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы (50 час.)

Лабораторная работа № 1. Основы WWW. Базовые понятия верстки. HTML, CSS. (4 часа)

Лабораторные работы 2-3. Установка и настройка виртуального веб-сервера. (8 часов)

Лабораторные работы 4-8. PHP. Базовые понятия и парадигмы. Типизация. Операторы. Работа с СУБД. (10 часов)

Лабораторная работа 9. JavaScript. jQuery. (4 часа)

Лабораторная работа № 10. Создание интернет-приложения, с использованием фреймворка Django (10 часов)

Лабораторная работа № 11. Создание интернет-приложения, с использованием фреймворка Ruby on Rails (8 часов)

Лабораторная работа № 12. Создание интернет-приложения, с использованием фреймворка Yii (8 часов)

Ш. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Сетевые и интернет технологии» представлено в разделе VIII и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1.	Основы WWW, HTML, HTTP.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 1-8
			умеет владеет	ПР-6 Лабораторная работа 1	
2.	Веб-сервер. Установка, настройка, принцип работы.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 9-15
			умеет владеет	ПР-6 Лабораторные работы 2-3	
3.	PHP.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 16-58
			умеет владеет	ПР-6 Лабораторные работы 4-8	
4.	Работа с СУБД в PHP.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 1-17
			умеет владеет	ПР-6 Лабораторные работы 4-9	
5.	Веб-фреймворки	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен вопросы № 59-64
			умеет владеет	ПР-6 Лабораторные работы 10-12	
6.	Безопасность интернет ПО.	ОПК-3	знает	УО-1	Экзамен,

		ПК-9 ПК-10	умеет владеет	ПР-6 Лабораторные работы 10-12	вопросы 65-70
7.	Оптимизация нагруженных интернет проектов.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен, вопросы 71-73
			умеет владеет	ПР-6 Лабораторные работы 10-12	
	Жизненный цикл современного интернет проекта	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен вопросы №74-78
			умеет владеет	ПР-6 Лабораторные работы 10-12	
	Перспективы развития веб-технологий.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен, вопрос 79

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в разделе IX.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

1. Молчанова Л.А., Прудникова Л.И. Java в примерах и задачах: учеб.-метод. пособие [для вузов]. Владивосток: Изд-во Тихоокеанского экономического университета. – 2011. Режим доступа: http://lib.dvfu.ru:8080/lib/item?id=chamo:359168&theme=FEFU
2. Лисьев, Г. А. Программное обеспечение компьютерных сетей и web-серверов : учеб. пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. — Москва : ИНФРА-М, 2019. — 145 с. — (Высшее образование: Бакалавриат). — www.dx.doi.org/10.12737/textbook_5a93ba6860adc5.11807424. - ISBN 978-5-16-106225-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/1002586
3. Савельева Н.В. Основы программирования на PHP. Курс лекций [Электронный ресурс]: учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий/

Савельева Н.В.— Электрон. текстовые данные.— Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017.— 264 c.— Режим доступа: http://www.iprbookshop.ru/67381.html.

4. Введение в СУБД MySQL [Электронный ресурс]/ — Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 228 с.— Режим доступа: http://www.iprbookshop.ru/73650.html.
5. Хартл, М. Ruby on Rails для начинающих / М. Хартл. — Москва : ДМК Пресс, 2017. — 572 с. — ISBN 978-5-97060-429-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/90110 (дата обращения: 06.04.2020). — Режим доступа: для авториз. пользователей.
6. Перепелица Ф.А. Разработка интерактивных сайтов с использованием jQuery [Электронный ресурс]/ Перепелица Ф.А.— Электрон. текстовые данные.— Санкт-Петербург: Университет ИТМО, 2015.— 144 c.— Режим доступа: http://www.iprbookshop.ru/68076.html.

Дополнительная литература

1. Скляр Д., Трахтенберг А. PHP. Рецепты программирования. – СПб: Русская редакция БХВ-Петербург. -2007. Режим доступа: http://lib.dvfu.ru:8080/lib/item?id=chamo:381965&theme=FEFU
2. Бенкен Е.С. PHP, MySQL, XML программирование для Интернета. СПб: БХВ-Петербург. -2008. Режим доступа: http://lib.dvfu.ru:8080/lib/item?id=chamo:382736&theme=FEFU
3. Хабибуллин, Самоучитель JAVA. - СПб: БХВ-Петербург. -2001. Режим доступа: http://lib.dvfu.ru:8080/lib/item?id=chamo:672412&theme=FEFU
4. PHP. Web- профессионалам. Пер. с англ. К. Косентино. – СПб: Питер. - 2001. Режим доступа: http://lib.dvfu.ru:8080/lib/item?id=chamo:15249&theme=FEFU
7. Головатый А., Каплан-Мосс Д.. Django. Подробное руководство // Символ-Плюс, 2010. - 560 с.
8. Брюс А. Тейт, Курт Ниббс. Ruby on Rails. Быстрая веб-разработка. // БХВ-Петербург, 2008. - 224 с.
9. Ю. Жуков. Основы веб-хакинга. Нападение и защита // Питер, 2011. - 176 с

10. Мациевский Н. Разгони свой сайт. Методы клиентской оптимизации веб-страниц: Учебное пособие.- Интернет университет Информационных технологий БИНОМ. Лаборатория знаний. 2009.
11. Гаврилов А.В,, Климентов С.В., Цопа Е.А. Программирование на Java: конспект лекций.- СПб: ИТМО. 2010. – 130 с.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Информационный портал. Все о компьютере и программировании для начинающих http://info-comp.ru/knigigutnalifile/
2. Основы Java для начинающих http://javabegin.ru/free
3. Список документации по PHP http://php.find-info.ru/
4. Язык программирования PHP. Уроки PHP http://www.alfafreedesign.ru/menu-php.html
5. Язык программирования PHP http://php.find-info.ru/php/012/index.htm
6. Примеры на PHP http://php.find-info.ru/php/articles/article_2.htm
7. http://www.twirpx.com/file/248121/Головатый А., Каплан-Мосс Д.. Django. Подробное руководство // Символ-Плюс, 2010. - 560 с.
8. http://www.djangobook.com - The Django Book.
9. http://www.yiiframework.ru/doc/guide/ru/index - Полное руководство по Yii на русском.
10. http://russian.railstutorial.org/chapters/beginning - Изучение Rails на примерах.
11. http://nginx.org/ru/docs/ - Документация по Nginx.
12. Справочная информация по программированию http://codingcraft.ru/glossary.php

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Microsoft Office / Open Office.

2. Интегрированные среды разработки программ (NetBeans, Eclipse, Qt Creator, Embarcadero RAD Studio, или Microsoft Visual Studio).

3. Программное обеспечение электронного ресурса сайта ДВФУ, включая ЭБС ДВФУ.

При осуществлении образовательного процесса бакалаврами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.

2. Электронно-библиотечная система IPRbooks.

3. Информационная система "ЕДИНОЕ ОКНО доступа к образовательным ресурсам".

4. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лекции проводятся с использованием проектора и мультимедийного комплекса для проведения лекций внутренней системы портала ДВФУ. Лабораторные занятия проводятся в специализированном компьютерном классе.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина изучается в следующих организационных формах: лекционное занятие; лабораторное занятие; самостоятельное изучение теоретического материала; самостоятельное выполнение индивидуального проекта; индивидуальные и групповые консультации.

Работа на лекции

Работа на лекции – это сложный процесс, который включает в себя такие элементы как слушание, осмысление и собственно конспектирование. Для того, чтобы лекция выполнила свое назначение, важно подготовиться к ней и ее записи еще до прихода преподавателя в аудиторию, поскольку в первые минуты лекции объявляется тема лекции, формулируется ее основная цель. Без этого дальнейшее восприятие лекции становится сложным. Важно научиться слушать преподавателя во время лекции. Здесь не следует путать такие понятия как слышать и слушать. Слышать можно не слушая, с чем мы часто сталкиваемся. Таким образом, слушание лекции состоит из нескольких этапов, начиная от слышания (первый шаг в процессе осмысленного слушания) и заканчивая оценкой сказанного.

Слушание является лишь одним из элементов хорошего усвоения лекционного материала. Однако, одного слушания недостаточно. Даже самая

хорошая память не в состоянии удержать тот поток информации, который сообщается во время лекции, поэтому его необходимо фиксировать, записывать – научиться вести конспект лекции, где формулировались бы наиболее важные моменты, основные положения, излагаемые лектором. Для ведения конспекта лекции следует использовать тетрадь. Ведение конспекта на листочках не рекомендуется, поскольку они не так удобны в использовании и часто теряются. При оформлении конспекта лекции необходимо оставлять поля, где студент может записать свои собственные мысли, возникающие параллельно с мыслями, высказанными лектором, а также вопросы, которые могут возникнуть в процессе слушания, чтобы получить на них ответы при самостоятельной проработке материала лекции, при изучении рекомендованной литературы или непосредственно у преподавателя в конце лекции.

Составляя конспект лекции, следует оставлять значительный интервал между строчками. Это связано с тем, что иногда возникает необходимость вписать в первоначальный текст лекции одну или несколько строчек, имеющих принципиальное значение и почерпнутых из других источников. Расстояние между строками необходимо также для подчеркивания слов или целых групп слов (такое подчеркивание вызывается необходимостью привлечь внимание к данному месту в тексте при повторном чтении). Обычно подчеркивают определения, выводы.

Главным отличием конспекта лекции от текста является свертывание текста. При ведении конспекта удаляются отдельные слова или части текста, которые не выражают значимую информацию, а развернутые обороты речи заменяют более лаконичными или же синонимичными словосочетаниями. При конспектировании основную информацию следует записывать подробно, а дополнительные и вспомогательные сведения, примеры – очень кратко. Особенно важные моменты лекции, на которые следует обратить особое внимание лектор, как правило, читает в замедленном темпе, что позволяет сделать их запись дословной. Также важно полностью без всяких изменений вносить в тетрадь схемы, таблицы, чертежи и т.п., если они предполагаются в лекции. Для того, чтобы совместить механическую запись с почти дословным фиксированием наиболее важных положений, можно использовать системы условных сокращений. В первую очередь сокращаются длинные слова и те, что повторяются в речи лектора чаще всего. При этом само сокращение должно быть по возможности кратким.

Приемы сокращений.

- Сокращение аббревиатурой – основные термины, повторяющиеся наиболее часто, могут быть выделены как ключевые слова и обозначены начальными заглавными буквами этих слов. Ключевые слова в первый раз записываются полностью, после них в скобках приводится их аббревиатура, далее в тексте будет фигурировать только аббревиатура. Например: язык программирования (ЯП), программное обеспечение (ПО). Ключевых слов не должно быть много, иначе может возникнуть путаница в их использовании.
- Сокращение слов до начальной части, базируясь на корне (например: аппарат (апп.), однократный (однокр.).
- Сокращение общепринятых вспомогательных слов (например: таким образом (т.о.), главным образом (гл. обр.), может быть (м.б.), смотри (см.), так называемый (т.н.), какой-либо (к-л).
- Использование латинского алфавита (например: максимум (max), минимум (min), температура (t).
- Использованием эквивалентных выражений или слов английского языка, (например: использование (use), если (if), переменный (var), постоянный (const)).
- Использование математических знаков (например: больше (>), меньше (<)).

По окончании лекции работа студента на этом не прекращается. Начинается процесс его самообразования. Следует проработать (расшифровать) сделанные записи. Этот процесс состоит из нескольких этапов:

- чтение записей, сделанных в процессе слушания и конспектирования лекции, еще раз просматривается важное, существенное в развитии мысли;
- уточнение с помощью книги не вполне ясного;
- контроль себя осуществляется путем привлечения справочной литературы и т.д

Лабораторные работы

В результате выполнения лабораторных работ студент должен изучить основной спектр высокоуровневых интернет технологий, таких как популярные фреймворки и библиотеки, основные используемые в них архитектурные подходы; усвоить и закрепить основные приемы, методы и

принципы работы при создании программ для Интернет, получить навыков использования средств разработки интернет приложений.

Лабораторная работа № 1. Основы WWW. Базовые понятия верстки. HTML, CSS. (4 часа)

- ***Цель:*** получить практические навыки работы в области верстки сайтов посредством HTML и CSS. На практичке изучить тонкости отображения верстки в различных браузерах.
- ***Теоретическая часть:***
    1. Муссиано Ч., Кеннеди Б. HTML и XHTML. Подробное руководство, 2008, ISBN: 5-93286-104-5
    2. Дженнифер Нидерст Роббинс. Web-дизайн. Справочник, ISBN 978-5-91136-039-9, 0-596-00987-9, 2008
    3. Алан Купер, Роберт Рейман, Дэвид Кронин. Алан Купер об интерфейсе. Основы проектирования взаимодействия, 2009
    4. Нильсен Я. Веб-дизайн. Книга Якоба Нильсена, 2006
    5. Раскин Дж. Интерфейс. Новые направления в проектировании компьютерных систем, 2004
    6. Мейер Э.А. CSS - каскадные таблицы стилей. Подробное руководство (3-е издание), 2008
    7. Кристофер Шмитт. CSS. Рецепты программирования (3-е издание), 2011
    8. Джеффри Зельдмани. Мастерская CSS. Профессиональное применение web-стандартов, 2007
    9. Джеффри Зельдман. Web-дизайн по стандартам, ISBN 5-477-00106-2, 0-7357-1201-8; 2005 г.
- ***Постановка задачи:*** разработать HTML страницу со следующими элементами: заголовки, таблицы, списки, ссылки, блочные элементы, изображения. Для заголовков показать различную верстку для различных уровней при помощи CSS. Таблицы отверстать CSS, показать объединение ячеек, как вертикальное так и горизонтальное. Для списков при помощи CSS показать изменение отступов и маркеров.
- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Разработка структуры программы

4. Программирование

- ***Форма отчета:*** готовая HTML страница.

Лабораторные работы 2-3. Установка и настройка виртуального веб-сервера. (8 часов)

- ***Цель:*** получить практические навыки работы в области установки и настройки UNIX систем для целей организации на их базе виртуальных веб-серверов.
- ***Теоретическая часть:***

1. систем, 2004

- ***Постановка задачи:*** создать виртуальную машину на ОС Ubuntu. Установить и настроить на ней LAMP-сервер. Создать пару виртуальных хостов, продемонстрировать их работу в браузере основной ОС компьютера.
- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Установка и настройка ПО
- ***Форма отчета:*** готовая виртуальная машина на ОС Ubuntu с работающим на ней веб-сервером и парой тестовых виртуальных хостов.

Лабораторные работы 4-8. PHP. Базовые понятия и парадигмы. Типизация. Операторы. Работа с СУБД. (12 часов)

Лабораторная работа № 4. (2 часа)

- ***Цель:*** получить практические навыки работы с разработанным в рамках ЛР веб-сервером, языком программирования PHP. На практике увидеть разницу между интерпретируемыми и компилируемыми языками програмирования.
- ***Теоретическая часть:***

1. Котеров Д., Костарев А. PHP 5. Наиболее полное руководство (2-е издание), 2008
2. Шлосснейгл Дж. Профессиональное программирование на PHP, 2006
3. Зандстра М. PHP. Объекты, шаблоны и методики программирования, 2011
4. Скляр Д., Трахтенберг А. PHP. Сборник рецептов, 2005

- ***Постановка задачи:*** разработать программу на PHP, которая выводит на экран таблицу состоящую из 100 строк и 5 колонок. Для вывода таблицы использовать циклы.
- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Разработка структуры программы
    4. Программирование
- ***Форма отчета:*** скрипт на PHP расположенный на отдельном виртуальном хосте настроенного и рабочего виртуального сервера.

Лабораторная работа № 5. (2 часа)

- ***Цель:*** получить практические навыки работы с разработанным в рамках ЛР веб-сервером, языком программирования PHP. На практике увидеть разницу между интерпретируемыми и компилируемыми языками програмирования.
- ***Теоретическая часть:***

1. Котеров Д., Костарев А. PHP 5. Наиболее полное руководство (2-е издание), 2008
2. Шлосснейгл Дж. Профессиональное программирование на PHP, 2006
3. Зандстра М. PHP. Объекты, шаблоны и методики программирования, 2011
4. Скляр Д., Трахтенберг А. PHP. Сборник рецептов, 2005

- ***Постановка задачи:*** доработать программу из ЛР 1 таким образом, чтобы размеры таблицы можно было указать в GET запросе к скрипту двумя параметрами. Операторами if else сделать обработку возможных ошибок.
- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Разработка структуры программы
    4. Программирование
- ***Форма отчета:*** скрипт на PHP расположенный на отдельном виртуальном хосте настроенного и рабочего виртуального сервера.

Лабораторная работа № 6. (2 часа)

- ***Цель:*** получить практические навыки работы с разработанным в рамках ЛР веб-сервером, языком программирования PHP. На практике

увидеть разницу между интерпретируемыми и компилируемыми языками програмирования.

- ***Теоретическая часть:***

1. Котеров Д., Костарев А. PHP 5. Наиболее полное руководство (2-е издание), 2008
2. Шлосснейгл Дж. Профессиональное программирование на PHP, 2006
3. Зандстра М. PHP. Объекты, шаблоны и методики программирования, 2011
4. Скляр Д., Трахтенберг А. PHP. Сборник рецептов, 2005

- ***Постановка задачи:*** доработать программу из ЛР 1 таким образом, чтобы размеры таблицы можно было передать в скрипт POST запросом из формы, расположенной на этой же странице. В поле должно быть три поля: количество строк, количество столбцов и флаг указывающий, нужно ли выводить нумерацию строк в отдельной колонке.
- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Разработка структуры программы
    4. Программирование
- ***Форма отчета:*** скрипт на PHP расположенный на отдельном виртуальном хосте настроенного и рабочего виртуального сервера.

Лабораторная работа № 7. (2 часа)

- ***Цель:*** получить практические навыки работы с разработанным в рамках ЛР веб-сервером, языком программирования PHP. На практике увидеть разницу между интерпретируемыми и компилируемыми языками програмирования.
- ***Теоретическая часть:***

1. Котеров Д., Костарев А. PHP 5. Наиболее полное руководство (2-е издание), 2008
2. Шлосснейгл Дж. Профессиональное программирование на PHP, 2006
3. Зандстра М. PHP. Объекты, шаблоны и методики программирования, 2011
4. Скляр Д., Трахтенберг А. PHP. Сборник рецептов, 2005

- ***Постановка задачи:*** реализовать классический механизм авторизации пользователя на странице через сессии. Создать две страницы: на первой форма регистрации пользователя, на второй форма авторизации пользователя. Пользователи хранятся в БД (MySQL или PostgresSQL на

выбор). При регистрации проверять уникальность пользователя по логину/адресу почты. При авторизации выодить сообщение об успешной авторизации или же ошибке.

- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Разработка структуры программы
    4. Программирование
- ***Форма отчета:*** скрипт на PHP расположенный на отдельном виртуальном хосте настроенного и рабочего виртуального сервера.

Лабораторная работа № 8. (2 часа)

- ***Цель:*** получить практические навыки работы с разработанным в рамках ЛР веб-сервером, языком программирования PHP. На практике увидеть разницу между интерпретируемыми и компилируемыми языками програмирования.
- ***Теоретическая часть:***
    1. Котеров Д., Костарев А. PHP 5. Наиболее полное руководство (2-е издание), 2008
    2. Шлосснейгл Дж. Профессиональное программирование на PHP, 2006
    3. Зандстра М. PHP. Объекты, шаблоны и методики программирования, 2011
    4. Скляр Д., Трахтенберг А. PHP. Сборник рецептов, 2005
- ***Постановка задачи:*** реализовать механизм построение меню сайт с любым уровнем вложенности иерархии. Создать две страницы: на первой производится добавление пунктов меню в древовидную структуру, на второй производится вывод меню вертикально в виде вложенных списков. Допускается любой уровень вложенности меню. На странице добавления пунктов должен быть реализован механизм авторизации, разработанный в ЛР 4. Помимо возможности добавлять и редактировать пункты меню, пользвателю должна быть предоставлена возможность менять их сортировку. Пункты меню зранить в БД.
- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Разработка структуры программы
    4. Программирование

- ***Форма отчета:*** скрипт на PHP расположенный на отдельном виртуальном хосте настроенного и рабочего виртуального сервера.

Лабораторная работа 9. JavaScript. jQuery. (4 часа)

- ***Цель:*** получить практические навыки работы с разработанным в рамках ЛР веб-сервером, языком программирования PHP. На практике увидеть разницу между интерпретируемыми и компилируемыми языками программирования.
- ***Теоретическая часть:***
    1. Котеров Д., Костарев А. PHP 5. Наиболее полное руководство (2-е издание), 2008
    2. Шлосснейгл Дж. Профессиональное программирование на PHP, 2006
    3. Зандстра М. PHP. Объекты, шаблоны и методики программирования, 2011
    4. Скляр Д., Трахтенберг А. PHP. Сборник рецептов, 2005
    5. Флэнаган Д. JavaScript. Подробное руководство (5-е издание), 2008
    6. Гудман Д. JavaScript и DHTML. Сборник рецептов. Для профессионалов, 2004
    7. Беэр Б., Иегуда К. jQuery. Подробное руководство по продвинутому JavaScript
    8. Ресиг Дж. JavaScript. Профессиональные приёмы программирования [2008, PDF/HTML, RUS]
- ***Постановка задачи:*** доработать скрипты написанные в рамках ЛР 5 Темы 3 таким образом, чтобы выводимое вертикально меню преобразовывалась посредством jQuery и JavaScript в горизонтальное меню с выпадающими подменю. Отверстать дизайн меню при помощи CSS. Дизайн на своё усмотрение.
- ***Порядок выполнения:***
    1. Анализ задачи
    2. Формальная постановка
    3. Разработка структуры программы
    4. Программирование
- ***Форма отчета:*** скрипт на PHP расположенный на отдельном виртуальном хосте настроенного и рабочего виртуального сервера.

Лабораторная работа № 10.Создание интернет-приложения, с использованием фреймворка Django (10 часов)

Лабораторная работа № 11.Создание интернет-приложения, с использованием фреймворка Ruby on Rails (8 часов)

Лабораторная работа № 12.Создание интернет-приложения, с использованием фреймворка Yii (8 часов)

Работа с литературными источниками

В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на поиск и на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выполнения индивидуального проекта, выявить широкий спектр мнений по изучаемой проблеме.

Самостоятельная работа студента

Основными формами самостоятельной работы студента являются:

- подготовка к лекциям, лабораторным занятиям, экзамену, презентации,
- изучение обязательной и дополнительной литературы,
- поиск информации по изучаемым темам в периодических изданиях и Интернете,
- изучение в рамках программы курса тем, не выносимых на лекции,
- оформление отчетов по лабораторным работам.

Контроль за выполнением работы студента производится в виде контроля каждого этапа работы (см. приложение 1).

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекции проводятся с использованием проектора и внутренней системы портала ДВФУ. Лабораторные занятия проходят в аудиториях, оборудованных компьютерами типа Lenovo C360G-i34164G500UDK с лицензионными программами MicrosoftOffice 2013 и аудио-визуальными средствами проектор Panasonic DLPProjectorPT-D2110XE, плазма LG FLATRON M4716CCBAM4716CJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi.

VIII. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Самостоятельная работа обучающихся подразумевает обязательную подготовку к лабораторным занятиям (оформление отчетов), изучение основной и дополнительно литературы по дисциплине, подготовку к текущему контролю и промежуточной аттестации в конце семестра, консультации преподавателей

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1-54 неделя обучения	Работа с литературой	25	Собеседование
2	1-54 неделя обучения	Оформление отчетов по лабораторным работам	25	Проверка отчетов
3	18,54 неделя	Подготовка к промежуточной аттестации	27	Экзамен
	ВСЕГО		125 час - СРС 27 час - Контроль	

Рекомендации по работе с литературой

Для более эффективного освоения и усвоения материала рекомендуется ознакомиться с теоретическим материалом по той или иной теме до проведения лабораторного занятия. Всю учебную литературу желательно изучать «под конспект».

Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

Работу с теоретическим материалом по теме можно проводить по следующей схеме:

- название темы;
- цели и задачи изучения темы;
- основные вопросы темы;
- характеристика основных понятий и определений, необходимых для усвоения данной темы;
- краткие выводы, ориентирующие на определенную совокупность сведений, основных идей, ключевых положений, систему доказательств, которые необходимо усвоить.

При работе над конспектом обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении консультаций, либо в индивидуальном порядке.

Подготовка к лабораторным работам

Подготовку к каждой лабораторной работе каждый студент должен начать с изучения теоретического материала и ознакомления с планом, который отражает содержание предложенной темы. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы по теме задания, правильном выполнении лабораторной работы.

В процессе выполнения лабораторной работы студент должен создать требуемый документ с помощью предлагаемого программного средства и выполнить требуемые в задании операции. Задание по лабораторной работе содержит методические указания по подготовке документа, который должен быть получен в результате выполнения работы. При подготовке к лабораторной работе следует их внимательно прочесть.

Критерии оценки отчетов по лабораторным работам (проектов)

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано знание и владение навыками подготовки документа по теме. Допущено не более 2 ошибок, связанных с пониманием структуры и содержания задания.

– 60-50 баллов - если структура и содержание задания не соответствуют требуемым.

Шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 балов	зачтено	отлично

IX. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
8.	Основы WWW, HTML, HTTP.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 1-8
			умеет владеет	ПР-6 Лабораторная работа 1	
9.	Веб-сервер. Установка, настройка, принцип работы.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 9-15
			умеет владеет	ПР-6 Лабораторные работы 2-3	
10.	PHP.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 16-58
			умеет владеет	ПР-6 Лабораторные работы 4-8	
11.	Работа с СУБД в PHP.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Зачет вопросы № 1-17
			умеет владеет	ПР-6 Лабораторные работы 4-9	
12.	Веб-фреймворки	ОПК-3 ПК-9 ПК-10	знает	УО-1	экзамен вопросы № 59-64
			умеет владеет	ПР-6 Лабораторные работы 10-12	

13.	Безопасность интернет ПО.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен, вопросы 65-70
			умеет владеет	ПР-6 Лабораторные работы 10-12	
14.	Оптимизация нагруженных интернет проектов.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен, вопросы 71-73
			умеет владеет	ПР-6 Лабораторные работы 10-12	
	Жизненный цикл современного интернет проекта	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен вопросы №74-78
			умеет владеет	ПР-6 Лабораторные работы 10-12	
	Перспективы развития веб-технологий.	ОПК-3 ПК-9 ПК-10	знает	УО-1	Экзамен, вопрос 79

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	**Этапы формирования компетенции**		**критерии**	**показатели**
ОПК-3. Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	знает (пороговый уровень)	Современные средства автоматизации разработки интернет приложений	Знание средств автоматизации разработки интернет систем	Способность отвечать на вопросы о средствах автоматизации разработки интернет приложений
	умеет (продвинутый)	Использовать средства автоматизации разработки интернет приложений	Умение применить фреймворки при выполнении лабораторных работ	Наличие выполненных лабораторных работ
	владеет (высокий)	Методами разработки и оценки качества интернет приложений	Владение методами создания тестов и проверки качества созданных систем	Наличие подготовленных тестов
ПК-9. Владение навыками использования операционных	знает (пороговый уровень)	Особенности проектирования интернет приложений	Знание особенностей верстки интернет	Способность отвечать на вопросы об особенностях

систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных			приложений для разных платформ	верстки интернет приложений
	умеет (продвинутый)	Использовать существующие средства создания интернет систем	Умение создавать интернет системы, учитывая особенности платформ	Наличие выполненных лабораторных работ
	владеет (высокий)	Методами выбора подходящих инструментальных средств для разработки интернет приложения	Владение выбором инструментальных средств и обоснованием выбора	Способность обосновать выбор использованных средств разработки
ПК-10. Владение навыками использования различных технологий разработки программного обеспечения	знает (пороговый уровень)	Особенности создания интернет приложений для разных классов операционных систем	Знание особенностей платформ и методов учета этих особенностей при создании систем	Способность отвечать на вопросы о особенностях платформ
	умеет (продвинутый)	Программировать настройку интерфейса интернет приложений для разных классов операционных систем	Умение создать системы с настраиваемым отображением интерфейса на разных платформах	Наличие выполненных лабораторных работ
	владеет (высокий)	Методами проверки работоспособности создаваемых интернет систем для разных классов операционных систем	Владение методами проверки правильности работы интернет приложений на разных платформах	Способность продемонстрировать правильность работы приложений на разных платформах

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Текущая аттестация студентов. Текущая аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и

является обязательной.

Текущая аттестация проводится в форме собеседования (устного опроса) для проверки теоретических знаний, а также в форме защиты проекта, выполняемого в рамках самостоятельной работы параллельно с лабораторными работами и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

– степень усвоения теоретических знаний - оценивается в форме собеседования и контрольных работ;

– уровень овладения практическими умениями и навыками – оценивается в форме защиты индивидуального заданий (проектов), выполняемого в рамках лабораторных.

Критерии оценки устного ответа

– **100-85 баллов** - если ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

– **85-76 баллов** - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

– **75-61 балл** - оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

– **60- 0** баллов - ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы;

незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области

Критерии оценки программы по лабораторным работам (проектов)

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками разработки, тестирования программ на языке программирования. Программа правильно работает на всех наборах входных данных. Текст программы содержит комментарии.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками разработки программ на языке программирования. Программа правильно работает не на всех наборах входных данных (90%). Текст программы содержит комментарии.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано знание методов разработки программ на языке программирования Программа правильно работает не на всех наборах входных данных (70%). В тексте программы комментарии отсуствуют.

60- 0 баллов - если структура и содержание задания не соответствуют требуемым

Шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 балов	зачтено	отлично

Текущий контроль состоит в проверке правильности выполнения лабораторных работ и решения задач для самостоятельного выполнения с использованием языков PHP и Java.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕЙ АТТЕСТАЦИИ

Контрольные работы

1. **Работа с формами. Программирование на Java Script.**

Вариант 1. Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 4-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	50% окна браузера
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, SQRT, C, ON/OFF
Цвет панели/кнопок	Голубой/Linen

Вариант 2. Написать скрипт, реализующий калькулятор, позволяющий вводить в тексто вое поле арифметич. выражение (напр. : 2+(3/4-7.5)*3) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 4-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	50% окна браузера
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, SQRT, C, ON/OFF
Цвет панели/кнопок	Голубой/Linen

Вариант 3. Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля,	30px

кнопок	
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, C, AC, M+, M-, M ON/OFF
Цвет панели	Желтый/голубой

Вариант 4. Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. : 2+(3/4-7.5)*3) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 4-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
Кнопки	Цифры, арифм. операции, C, AC, M+, M-, M, ON/OFF
Цвет панели/кнопок	Голубой/ ….

Вариант 5. Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
кнопки	Цифры, арифм. операции, C, ON/OFF, F0, F2, F4, где Fn – округление (ост. n цифр после запятой)

Вариант 6. Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. : 2+(3/4-7.5)*3) и вычисляющего его после нажатия на =. Вид: верхн.

строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	500px
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, серый
кнопки	Цифры, арифм. операции, C, ON/OFF, F0, F2, где Fn – округление (ост. n цифр после запятой)

Вариант 7. Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =, функция-аргумент- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	50%
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, красный
кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF

Вариант 8. Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. : SIN(2)+(3/4-7.5)*3) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Параметры калькулятора:

Ширина кальк.	50%
Высота текст. поля/кнопок	30px/25px
Шрифт на кнопках	18pt, bold, черный (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, синий

кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF

Вариант 9. Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =, функция-аргумент- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, черный
кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF

Вариант 10. Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. : SIN(2)+(3/4-7.5)*3) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, черный (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, синий
кнопки	Цифры, арифм. операции, C, SQRT, ^(степень), LOG2, LOG10, LN, ON/OFF

Вариант 11. Написать скрипт, реализующий калькулятор в “обычном” режиме (число-операция-число.- =, функция-аргумент- =). Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, синий (цифры), зеленый (др.)

Шрифт в текстовом поле	18pt, bold, черный
кнопки	Цифры, арифм. операции, C, SIN, COS, TG, CTG, PI (конст.), ON/OFF
Радиокнопки (в отд. строке)	F0, F2 (Fn – округление (ост. n цифр после запятой)), F – с обычной точностью

Вариант 12. Написать скрипт, реализующий калькулятор, позволяющий вводить в текстовое поле арифметич. выражение (напр. : SIN(2)+(3/4-7.5)*3) и вычисляющего его после нажатия на =. Вид: верхн. строка (во всю ширину) – текст. поле; далее кнопки в 5-х рядах (во всю ширину). Требования:

Ширина кальк.	500px
Высота текст. поля, кнопок	30px
Шрифт на кнопках	18pt, bold, черный (цифры), зеленый (др.)
Шрифт в текстовом поле	18pt, bold, синий
кнопки	Цифры, арифм. операции, C, SQRT, ^(степень), LOG2, LOG10, LN, ON/OFF
Радиокнопки (в отд. строке)	F0, F2 (Fn – округление (ост. n цифр после запятой)), F – с обычной точностью

2. Программирование на Java Script. Регулярные выражения. Проверка корректности ввода текстовых данных

Создать web-страницу, содержащую текстовые поля для ввода персональных данных с ограничениями согласно таблице:

№	Текстовое поле	Ограничения
1.	Фамилия	Буквы русского алфавита. Первая - заглавная
2.	Имя	
3.	Отчество	
4.	Число, месяц и год рождения	Шаблон: dd-mm-yyyy, например 18-10-1986. Ограничения на вводимые числа
5.	e-mail	Латинский шрифт, цифры. Первый символ – буква. Разделители: @ и .
6.	Логин	Первый символ – буква. Не менее 6 символов
7.	Пароль	Не может содержать одни цифры.

Разбивку страницы и выравнивание текстовых полей осуществить с помощью таблицы. Разместить на странице кнопку, по нажатию на которую будет происходить проверка введенных данных и вывод соответствующего сообщения об их корректности.

Приложение

1. Некоторые спец. символы, используемые в регулярных выражениях:

^	начало строки
$	конец строки
.	любой символ
x | *y*	один из символов *x* или *y*
{*n*}	*n* повторений предыдущего символа
{*n*,}	*n* или более повторений предыдущего символа
{*n*,*m*}	от *n* до *m* повторений предыдущего символа
+	предыдущий символ встречается 1 или более
[*xyz*]	раз
[^*xyz*]	один из перечисленных символов
\d	любой символ кроме перечисленных
\D	цифровой символ (эквивалентно [0-9])
\w	нецифровой символ (эквивалентно [^0-9])
	любой буквенно-цифровой символ или знак _
\W	(эквивалентно [A-z0-9_])
	любой не буквенно-цифровой символ или знак _
	(эквивалентно [^A-z0-9_])

2. Метод exec(*строка*) осуществляет поиск регулярного выражения в заданной строке. Возвращает подстроку, соответствующую первому совпадению.

3. Некоторые свойства объекта RegExp :
input – возвращает строку, в которой был осуществлен результативный поиск
index – возвращает позицию первого успешного поиска (если нет совпадения, то -1)

4. Примеры регулярных выражений:
/\d+/ - одна или более цифр
/[A-z]{3,}/ - последовательность из не менее трех латинских букв
/^.{4}$/ - любая строка из 4-х символов
/^\D+$/ - строка из одного или более нецифровых символов

5. Пример странички, осуществляющей поиск регулярного выражения в задаваемой строке:

```
<html>

<script>
function fun(str,regexp)
{
var re = eval(regexp);
var a = re.exec(str);
alert(' Строка поиска: ' + RegExp.input + '\n Совпадение: ' + a + '\n Позиция первого совпадения: ' +
RegExp.index);
}
</script>

<body>
<p align="center">
<font face="Verdana,Arial" size="2">
Поиск регулярного выражения в заданной строке
</p>
<form name="fname">
Строка: <br>
<input type="text" name="str" size="30">
<br>
<br>
Регулярное выражение:<br>
<input type="text" name="regexp" size="30">
<br>
<br>
<input type="button" value="Поиск" onclick="fun(document.fname.str.value, document.fname.regexp.value);">
</form>
</font>
</body>
</html>
```

3. Программирование на Java Script. Использование таймера.

Создать web-страницу, реализующую электронные часы в формате ЧЧ : ММ : СС. Вывод текущего времени осуществить в текстовое поле <input>. Используя таблицы стилей CSS, задать размер текстового поля, фон,

параметры шрифта (размер, цвет, …). Включить в функцию электронных часов будильник. Время для сигнала будильника задавать в двух текстовых полях (часы, минуты). Сигнал будильника осуществить в виде периодической смены цвета фона текстового поля часов. Разбивку страницы и выравнивание текстовых полей осуществить с помощью таблицы.

Приложение

1. Некоторые методы объекта Date:
getHours(), getMinutes(), getSeconds()
– возвращает час суток (0-23), минуты(0-59), секунды(0-59)

2. Создание таймера осуществляется методом setInterval(), который выполняет код через указанный временной интервал. Синтаксис метода:
setInterval(*выполняемый код, временной интервал в милисекундах*).
По окончании работы, используйте метод clearInterval() для прекращения работы таймера и удаления его из памяти.

3. Пример реализации простейших часов:

```
<html>
<script>
var tmr;
function timer(){
tmr=setInterval("clock()",1000);
}
function clock(){
var d = new Date();
document.formname.textfield.value =
d.getHours()+':'+d.getMinutes()+':'+d.getSeconds();
}
function stop(){
clearInterval(tmr);
}
</script>
<body onload="timer();" onbeforeunload="stop();">
<form name="formname">
<input type="text" name="textfield" style="font-size: 14pt; font-weight: 700;
width: 100">
```

</form>
</body>
</html>

4. Программирование на Java Script. Использование Cookies.

Создать скрипт, который осуществляет сбор и вывод статистики посещений страницы конкретным клиентом. Идентификация компьютера клиента и хранение данных по статистике посещений осуществить с помощью Cookies. Выводимая статистика должна включать следующую информацию о посещениях:

Номера варианта	*Статистика*
1+3n	Посещения за последнюю неделю по дням
2+3n	Посещения за последний месяц, распределенные по времени суток (посещения в ночное, дневное и вечернее время)
3+3n	Количество посещений за последний месяц в выходные и рабочие дни

Предусмотреть в скрипте проверку разрешено ли использование Cookies на компьютере клиента (свойство cookieEnabled объекта naigator).

Приложение

1. Установка cookies осуществляется через свойство cookie объекта document: document.cookie = "<*имя cookie*>=<*значение cookie*>; expires=<*время действия cookie*>".
Время действия cookie задается в милисекундах в GMT формате.

2. При создании нескольких cookie (например, *cookie1, cookie2*, …) значение document.cookie представляет собой следующую строку:
" *cookie1*=<*значение cookie1*>; *cookie2*=<*значение cookie2*>; …."

3. Считывание значения cookie можно осуществлять через свойство length и метод substring( ) объекта String.

4. Для установки времени действия cookie требуется использовать методы объекта Date: getTime() – получение текущего времени (количество миллисекунд с 00:00 1.01.1970); toGMTString() – перевод времени в GMT-формат строкового типа.

5. Пример простейшего счетчика.

```
<html>
<script>
function f1(){
if(document.cookie){
var c = document.cookie;
var n = c.length;
counter = c.substring(8,n-1);
counter++;
var d = new Date();
d.setTime(d.getTime() + 10000000);
var exp = d.toGMTString();
document.cookie="visits="+counter+"; visits1="+counter+"; expires="+exp;
alert(counter);
}
else{
var d = new Date();
d.setTime(d.getTime() + 10000000);
var exp = d.toGMTString();
document.cookie="visits=1; expires="+exp;
alert(1);
}
}
</script>
<body onload="f1()">
</body>
</html>
```

5. Программирование на Java Script. Перемещение слоев.

Создать страницу, содержащую несколько графических элементов (например, елка и игрушки, либо продуктовая корзинка и продукты и др.). Все графические элементы необходимо спозицировать через стили. Написать скрипт, позволяющий перемещать графические элементы с помощью мыши. При перемещении какого-либо слоя он должен располагаться поверх остальных слоев. Предусмотреть наличие *главного* слоя, при перемещении которого все слои расположенные на нем перемещаются вместе с ним (например, при перемещении елки, игрушки перемещаются вместе с елкой).

Приложение

Пример скрипта реализующего перемещение слоев:

```
<html>
<head>
<title> Перемещение слоев</title>
<head>
<script language=javascript>
 var draggedEl=null;
 var shiftX, shiftY;
 function moveMouse(){
 if((event.button == 1) && (draggedEl != null)){
 draggedEl.style.pixelTop = event.clientY - shiftY;
 draggedEl.style.pixelLeft = event.clientX - shiftX;
 }
 }
 function isDragEnabled(el){
 if(el != null){
 if(el.getAttribute("dragEnabled") != null) return el;
 }
 return null;
 }
 function pressMouseDown(){
 var el = isDragEnabled(event.srcElement);
 if(el != null){
 draggedEl = el;
 shiftX = event.offsetX;
 shiftY = event.offsetY;
 }
 }
</script>
<body onmousedown = "pressMouseDown();"
onmousemove = "moveMouse();"
onmouseup = "draggedEl = null;"  >
<h1 style="position: absolute; left: 100; top: 100;" dragEnabled>
Привет участникам конференции!
</h1>
<h1 style="position: absolute; left: 100; top: 300;" dragEnabled>
Hello!!!
</h1>
</body>
</html>
```

Задачи для самостоятельного выполнения

1. **Программирование на PHP. Файловый ввод/вывод. Счетчик обращения к странице.**

Написать сценарий, реализующий счетчик обращения к странице. Требования: традиционный дизайн (либо путем включения графических элементов, либо путем использования таблицы стилей), наличие количества посещений за текущий день и общего количества посещений. Хранение всей информации осуществлять в текстовом файле. Считывание и запись данных в файл осуществить с помощью функций fwrite, fread, либо их аналогов.

Приложение

1. Проверка существования файла:
   file_exists ($filename) → TRUE/FALSE

2. Открытие и закрытие файлов:
   fopen($filename, режим) → файловый манипулятор (file handle), целое
   fclose(манипулятор)

   режимы:
   r только чтение. Указатель текущей позиции в начало.
   r+ чтение и запись…..
   w запись. Указатель в начало. Содержимое уничтожается. w+ чтение и запись…..
   a только запись. Указатель в конец Если файла нет – попытка создать….
   a+ чтение и запись…..

3. Запись в файл:
   fwrite(манипулятор, строка [, длина в символах])

4. Чтение из файла:
   fread(манипулятор, длина в байтах) читает из файла заданное количество байт
   fgets(манипулятор, длина в байтах)

5. Перевод файла в массив:
file ($filename) → массив (элементы – построчно)

6. Пример – Счетчик обращения к странице

```
<?
$cfile = "test1.txt";
$afile = file($cfile);              // Записываем содержимое файла в массив
$n_vis = $afile[0];                 // Извлекаем первый элемент
++$n_vis;
$fc = fopen($cfile, "w");       // Открываем файл, текущая позиция в начале
fwrite($fc, $n_vis);                // Записываем новое значение счетчика
fclose($fc);
print $n_vis;
?>
```

2. Файловый ввод/вывод. Работа с формами. Строковые функции.

Доработайте предлагаемый ниже сценарий, реализующий гостевую книгу. Чтение из файла осуществить с помощью функции fgets или fread. Сообщения должны быть отсортированы по времени отправки: более новые располагаются выше старых. При отображении сообщений нужно организовать автоматические разрывы, так чтобы длина строк не превосходила 50 символов. Сценарий должен реагировать на незаполненные текстовую область и поле.

```
<html>
<head><title>Гостевая книга</title></head>
<body>
Сообщение:<br>
<form action="guestbook.php" method="post">
<textarea name="txt" cols="40" rows="8">
</textarea>
<br><br>
Имя:<br>
<input type="text" name="login">
<br>
<input type="submit" name="a" value="Отослать">
<br><br>
</form>
```

```
<?
if(isset($a)){
$fm = fopen("guestbook.txt","a");
$str = "<i>".date("H:i:s, d F Y")."\n$login</i>\n$txt<br>\n";
fwrite($fm,$str);
fclose($fm);
}
$m = file("guestbook.txt");
$i=0;
while(isset($m[$i])){
print $m[$i]."<br>";
$i++;
}
?>
</body>
</html>
```

3. Программирование на PHP. Работа с таблицами базы данных. Регулярные выражения.

Переделайте гостевую книгу из лаб. работы 7. Хранение всех данных должно осуществляться в таблице базы данных. Добавить режим регистрации посетителей. Указываемые при регистрации данные: логин, пароль, фамилия, имя, е-мэйл, дата рождения. Сценарий должен посредством регулярных выражений осуществлять проверку корректности ввода персональной информации по каждой позиции (требования корректности – на усмотрение разработчика, исходя из здравого смысла).

Приложение

1. Работа с регулярными выражениями

Примеры шаблонов:

/ph+/ совпадение шаблона: phhhh, php4, phh

/ph{2,4}/ совпадение шаблона: phh, phhhhp4

Метасимволы:

\d – любая цифра. Пример: /(\d+)000\b/ - цифры, заканчив. 000.

\w – любой алф-цифровой символ

\W – любой символ не являющ., алф-цифр. Пример: /(\W+)/
\b – граница слова. Пример: /sa\b/ - строки, заканчив. на ‘sa’.
\B – не граница. Пример: /sa\B/ - строки, не заканчив. на ‘sa’.

Функции:
preg_match (шаблон, строка [, массив совпадений]) – совпадение шаблона в строке
preg_grep(шаблон, массив) → массив
- перебирает все элементы и возвращает те, где совпадает шаблон.

Пример. Поиск шаблона в строке.

```
<html>
<head><title>Поиск шаблона </title></head>
<body>
Строка:<br>
<form action="regexp.php" method="post">
<input type="text" name="str" value=<?if(isset($a)) print $str;?>>
<br><br>
Регулярное выражение:<br><br>
<input type="text" name="re" value=<?if(isset($a)) print $re;?>>
<br><br>
<input type="submit" name="a" value="Отослать"><br><br>
</form>
<?
if(isset($a)){
if(preg_match($re,$str)) print "Совпало!";
else print "No";
}
?>
</body>
</html>
```

2. Функции PHP для работы с MySQL

2.1**.** Подключение к серверу MySQL
[идентификатор соединения =]

@mysql_connect (хост, имя пользов, пароль) or die (“соединение не может быть уст-но”);

2.2. Выбор базы данных
[идентификатор базы =] @mysql_select_db (имя бд) or die (“соединение не может быть уст-но”);

2.3. Завершение работы с MySQL
@mysql_close (идентификатор соединения);

2.4. Организация запросов к БД
[идентификатор результата =] @mysql_query (запрос [, идентиф соедин]);

2.5. Получение набора данных после запроса mysql_query
mysql_result (идентификатор рез-та, номер записи [, поле]);

2.6. Количество записей, возвращаемых командой SELECT
mysql_num_rows (идентификатор рез-та);

Пример. Дана таблица products с полями: id, name, price. Требуется организовать выборку всех записей и напечатать их в виде таблицы.

```
$x=0 ;
$query = “ SELECT * FROM products” ;
$result = mysql_query ($query) ;
while ( $x <  mysql_num_rows ($result)) :
        $id = mysql_result ( $result, $x, ‘id’) ;
        $name = mysql_result ( $result, $x, ‘name’) ;
        $price = mysql_result ( $result, $x, ‘price’) ;
        print “ <tr> \n” ;
        print “ <td> $id </td> \n <td> $name </td> \n <td> $price </td> \n” ;
        print “ </tr> \n” ;
        $x++ ;
endwhile;
```

2.7. Передача набора данных после запроса mysql_query в массив
mysql_fetch_row (идентификатор рез-та) ; либо

mysql_fetch_array (идентификатор рез-та) ;

Пример. То же, что и в примере 2.1, но с использованием функции mysql_fetch_row .

```
…
while ($row = mysql_fetch_row ($result)) :
        print " <tr> \n" ;
        print " <td>".$row[0]." </td> \n <td>".$row[2]."</td> \n
<td>".$row[2]."</td> \n" ;
        print " </tr> \n" ;
endwhile;
```

3. Некоторые команды SQL

3.1. Создание БД, таблиц

CREATE DATABASE [IF NOT EXISTS] database
CREATE TABLE [IF NOT EXISTS] table (col1 CHAR(20), col2 INTEGER, col3 CHAR(20))

3.2. Выбор записей из таблицы

SELECT * FROM table - выбор всех записей

SELECT * FROM table WHERE col3 = 'Майор'

SELECT * FROM table WHERE col1 = 'И%' - выбор записей с фамилией на 'И'

SELECT col1, col2 FROM table WHERE col3 = 'Майор'

SELECT * FROM table Limit 10 - выбор первых 10 результирующих записей

3.3. Изменение записей

UPDATE table SET col3 = ‘Капитан’ WHERE col3 = ‘Майор’ - понижение звания всем майорам

3.4. Вставка записей

INSERT INTO table (col1, col2, col3) VALUES (‘Пупкин К.К.’, 1977, ‘Сержант’)

3.5. Уничтожение записей

DELETE FROM table LIMIT 5 - уничтожение первых 5 записей.

4. Программирование на PHP. Работа с таблицами базы данных. Cookie и сеансовые переменные.

Создать гостевую книгу, включающую в себя:
1. Наличие регистрации и идентификации пользователя.
2. Возможность изменять личные установки (e-mail, адрес, пол ;) и др.)
3. Проверка корректности ввода e-mail, даты рождения, Ф.И.О. посредством использования регулярных выражений.
4. Счетчик посещения страницы, не увеличивающий число посещений идущих с одного ip-адреса, либо с одного компьютера ранее чем за 3 часа.

Требования:
1. Хранение всех данных осуществляется в таблицах базы данных.
2. Предусмотреть использование сеансовых переменных и cookie.

5. Программирование на PHP. Обработка текстовых данных. Работа с формами.

1. Написать программу, которая подсчитывает процентное содержание слов различной длины в данном текстовом файле. Результат работы программы представить в виде таблицы и диаграммы.

Примечание.
Словом является слитная последовательность букв. Знаки препинания частью слова не являются, даже, если они написаны с ним слитно.

10.2. Разработать текстовый редактор с возможностями: отображение содержимого некоторого текстового файла в текстовой области; редактирование и сохранение, поиск заданной символьной последовательности, поиск и замена.

Приложение

Некоторые полезные функции:

substr(строка, начальная позиция [, длина]) –
возвращает часть строки начиная с зад позиции
strlen(стр) – длина строки
str_replace(подстрока, замена, строка) –
ищет в строке все входящие подстроки и делает замены
explode(разделитель, строка) – возвращает массив (переводит строку в массив)
implode(разделитель, массив) – возвращает строку (переводит массив в строку)

Промежуточная аттестация студентов. Промежуточная аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине предусмотрены зачет и экзамен, которые проводятся в устной форме.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Вопросы к зачету (5 семестр)

1. Основы HTML.
2. Каскадные таблицы стилей.
3. Протокол HTTP.
4. Динамический HTML.
5. DOM и клиентские скрипты.
6. Общий шлюзовый интерфейс (CGI).
7. Расширяемый язык разметки XML.

8. Технологии на основе XML.
9. Модули веб-сервера.
10. ISAPI и apache modules.
11. Установка Apache 1.3.29 под Windows XP.
12. Установка PHP 4.3.4 под Windows.
13. Установка PHP под Linux.
14. Дистрибутив Денвер.
15. Дистрибутив XAMPP
16. Введение в PHP. История PHP. Возможности PHP. У
17. становка и настройка ПО.
18. Первая PHP-программа. Основной синтаксис. Разделение инструкций. Комментарии.
19. Переменные, константы и операторы. Переменные. Константы. Операторы.
20. Типы данных. Тип boolean (булев или логический тип). Тип integer (целые). Тип float (числа с плавающей точкой). Тип string (строки). Одинарные кавычки. Двойные кавычки. Heredoc.
21. Тип array (массив). Определение при помощи array(). Определение с помощью синтаксиса квадратных скобок.
22. Тип object (объекты).
23. Тип resource (ресурсы).
24. Тип Null.
25. Функции в PHP.
26. Объекты и классы в PHP.
27. Управляющие конструкции.
28. Условные операторы. Оператор if. Оператор else. Оператор elseif. Альтернативный синтаксис.
29. Оператор switch.
30. Циклы: While, do... while, For, foreach.
31. Операторы передачи управления: Break, continue.
32. Операторы включения: include, require.
33. Обработка запросов с помощью PHP.
34. Основы клиент-серверных технологий.
35. Протокол HTTP и способы передачи данных на сервер.
36. Форма запроса клиента. Методы.
37. Использование HTML-форм для передачи данных на сервер.
38. Для метода GET. Для метода POST.

39. Обработка запросов с помощью PHP. Пример обработки запроса с помощью PHP
40. Массивы. Операции с массивами.
41. Функция count. Функция in_array. Функция array_search. Функция array_keys. Функция array_unique.
42. Сортировка массивов. Сортировка массива по ключам.
43. Сортировка с помощью функции, заданной пользователем. Применение функции ко всем элементам массива. В
44. Выделение подмассива. Функция array_slice. Функция array_chunk.
45. Строки. Поиск элемента в строке. Выделение подстроки.
46. Функция strstr. Функция substr.
47. Замена вхождения подстроки. Функция str_replace. Функция substr_replace.
48. Разделение и соединение строки. Строки, содержащие html-код
49. Работа с файловой системой. Создание файла.
50. Функция fopen. Закрытие соединения с файлом.
51. Запись данных в файл. Функция fwrite.
52. Чтение данных из файла. Функция fread. Функция fgets.
53. Функция fgetss. Функция fgetc. Функция readfile.
54. Функция file. Функция file_get_contents.
55. Проверка существования файла. Функция file_exists.
56. Функция is_writable. Функция is_readable.
57. Удаление файла. Загрузка файла на сервер
58. Обратная ссылка. Утверждения. Условные подвыражения

Вопросы к экзамену (7 семестр)

59. Назначение Веб-фреймворков
60. основные принципы устройства популярных фреймворков.
61. DRY.
62. Паттерны MVC, ORM.
63. URL dispatching.
64. Шаблонирование и кэширование.
65. Безопасность интернет ПО
66. Интернет-специфика безопасности.
67. Виды атак.
68. [D]DoS, Code/SQL инъекции,

69. XSS-уязвимости, phishing.
70. Типичные ошибки при написании кода, методы защиты. Упаковка стилей и библиотек.
71. Оптимизация работы с БД.
72. Кэширование, виды и стратегии кэширования.
73. Nginx, его преимущества.
74. Жизненный цикл современного интернет проекта
75. Разделение frontend/backend разработки, их специфика.
76. Команда проекта.
77. Идеология «вечной беты».
78. Итеративный подход к разработке.
79. Перспективы развития веб-технологий.

Вопросы к зачету (6 семестр)

1. Работа с СУБД в PHP.
2. Взаимодействие PHP и MySQL.
3. Построение интерфейса для добавления информации.
4. Установка соединения.
5. Выбор базы данных. Получение списка полей таблицы.
6. Отображение списка полей в html-форму.
7. Запись данных в базу данных.
8. Отображение данных, хранящихся в MySQL
9. Авторизация доступа с помощью сессий.
10. Авторизация доступа. Механизм сессий. Настройка сессий. Работа с сессиями.
11. Создание сессии. Регистрация переменных сессии. Удаление переменных сессии.
12. Регулярные выражения. Понятие регулярного выражения. Р
13. Регулярные выражения в PHP. Синтаксис регулярных выражений.
14. Подвыражения (подшаблоны).
15. Повторения (квантификаторы).
16. Модификаторы PCRE.
17. Регулярные выражения для "продвинутых".

Образец экзаменационного билета

Структура экзаменационного билета по курсу «Сетевые и интернет технологии»

1. Теоретический вопрос (вопрос № 1-29, 59-68 из списка вопросов к экзамену).
2. Теоретический вопрос (вопрос № 30-58, 69-79 из списка вопросов к экзамену).

Пример экзаменационного билета

1. . Оптимизация нагруженных интернет проектов
2. Команда проекта

Критерии оценивание по дисциплине

Оценивание происходит по формуле:

*Оитоговая = 0,2 * Онакопленная + 0,3 * Опроектная + 0,5 * Оитогового контроля*

- *Онакопленная* - накопленная оценка – среднее арифметическое из оценок проставленных за активность обучающегося на занятиях, прохождение текущего контроля и выполнение самостоятельной работы.
- *Опроектная* - проектная оценка - среднее арифметическое из оценок проставленных за защиту лабораторных работ по курсу.
- *Оитогового* контроля - оценка итогового контроля проставляется за прохождение контрольного испытания по курсу в формате, определенным рабочим учебным планом.

Оценки ставятся по 100-балльной шкале. Округление оценки производится в пользу студента.

Итоговая оценка выставляется в ведомость согласно следующему правилу:

Критерии выставления оценки студенту на зачете (экзамене)

Баллы	**Оценка зачета/ экзамена**	**Требования к сформированным компетенциям**

86-100	«зачтено»/ «отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	«зачтено»/ «хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«зачтено»/ «удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	«не зачтено»/ «неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.