



**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ИНСТИТУТ МЕЖДУНАРОДНЫХ ЭКОНОМИЧЕСКИХ СВЯЗЕЙ»**

**INSTITUTE OF INTERNATIONAL ECONOMIC RELATIONS**

Принята на заседании  
Учёного совета ИМЭС  
(протокол от 26 января 2022 г. № 6)

**УТВЕРЖДАЮ**  
Ректор ИМЭС Ю.И. Богомолова  
26 января 2022 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**РАЗРАБОТКА СЕРВЕРНЫХ ПРИЛОЖЕНИЙ ДЛЯ WEB**

по направлению подготовки  
09.03.02 Информационные системы и технологии

Направленность (профиль)  
«Информационные системы и сетевые технологии»

Москва – 2022

## 1. АННОТАЦИЯ К ДИСЦИПЛИНЕ

Рабочая программа дисциплины «Разработка серверных приложений для WEB» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденным приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 926.

Дисциплина «Разработка серверных приложений для WEB» формирует общую систему теоретических и концептуальных представлений о разработке интернет-приложений. В процессе изучения дисциплины рассматриваются современные подходы к разработке интернет-приложений. Предметом изучения является язык HTML, объектно-ориентированный язык JavaScript и серверный язык сценариев PHP. Объектом изучения выступают как сами языки, так и основополагающие концепции объектно-ориентированного и предметно-ориентированного программирования.

### **Место дисциплины в структуре образовательной программы**

Настоящая дисциплина включена в часть, формируемую участниками образовательных отношений, Блока 1 учебных планов по программам подготовки бакалавров по направлению подготовки бакалавров по направлению 09.03.02 Информационные системы и технологии.

Дисциплина изучается на 3 курсе в 5 семестре.

### **Цели и задачи дисциплины.**

*Целью изучения дисциплины* является формирование у обучающихся навыков работы со средствами разработки Web-приложений.

### **Задачи изучения дисциплины:**

- формирование знаний о правовых нормах при проектировании в сетевой среде;
- формирование умения планирования деятельности исходя из имеющихся ресурсов и ограничений, накладываемых технологиями и условиями эксплуатации;
- формирование знаний о структурах программного кода и баз данных при использовании в сетевой среде;
- формирование практического опыта сопровождения и модификации серверных приложений;
- освоение практического опыта программирования Web-серверов;
- изучение ведущих языков программирования для обеспечения динамики на Web-страницах на стороне сервера;
- формирование практического опыта по разработке программ с использованием методов объектно-ориентированного программирования.

## 2. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины направлен на формирование компетенций, предусмотренных образовательной программой.

Результаты освоения ООП (содержание компетенций)	Код компетенции	Код и наименование индикатора достижения компетенций	Перечень планируемых результатов обучения по дисциплине			Формы образовательной деятельности
			выпускник должен знать	выпускник должен уметь	выпускник должен иметь практический опыт	
Способность к проектированию, отладке, проверке работоспособности, созданию (модификации) и сопровождению информационных систем (ИС), автоматизирующих задачи организационного управления и бизнес-процессы с целью повышения эффективности деятельности организаций - пользователей ИС	ПК-2	<b>ПК-2.1</b> Разрабатывает и верифицирует структуру программного кода и баз данных ИС, автоматизирующих задачи организационного управления и бизнес-процессы организаций	структуры программного кода и баз данных ИС; методы и средства разработки программного кода и баз данных при веб-программировании	проводить анализ соответствия кода и процесса кодирования принятым в организации или проекте стандартам и технологиям; согласовывать необходимость внесения изменений в процесс разработки, кодирования и сопровождения программных систем	разработки пользовательских интерфейсов для веб; анализа и верификации пользовательского интерфейса и программного кода на его эффективность	Контактная работа: Лекции Лабораторные практикумы Самостоятельная работа
		<b>ПК-2.2.</b> Согласовывает необходимость внесения изменений, обеспечивает и контролирует соответствие разработанного кода и процесса кодирования на языках программирования принятым в организации или проекте стандартам и технологиям	основные автоматизирующие задачи организационного управления и бизнес-процессы организаций; методы проектирования, отладки и проверки работоспособности серверных приложений	организовать обеспечение и контроль за разработкой кода на принятых языках программирования; разрабатывать рабочий проект по заданным требованиям	модификации пользовательского интерфейса и программного кода с целью повышения эффективности деятельности организаций – пользователей	
		<b>ПК-2.3.</b> Разрабатывает, верифицирует и модифицирует пользовательские интерфейсы с целью повышения эффективности деятельности организаций -	основные методы и средства рабочего проектирования серверных приложений для веб	применять приложения в заданных условиях функционирования, изменять настройки и исполняемый код приложения в зависимости	работы со средствами и средой программирования и адаптации серверных приложений для веб	

Результаты освоения ООП (содержание компетенций)	Код компетенции	Код и наименование индикатора достижения компетенций	Перечень планируемых результатов обучения по дисциплине			Формы образовательной деятельности
			выпускник должен знать	выпускник должен уметь	выпускник должен иметь практический опыт	
		пользователей		от принятых в организации или проекте стандартов и технологий		

### 3. ТЕМАТИЧЕСКИЙ ПЛАН

Наименование тем	Контактная работа обучающихся с преподавателем (по видам учебных занятий)									Самостоятельная работа обучающихся	ТКУ / балл Форма ПА
	Лекции	Семинары	Практикум по решению задач	Ситуационный практикум	Мастер-класс	Лабораторный практикум	Тренинг	Дидактическая игра	Из них в форме практической подготовки		
<b>Очная форма</b>											
<i>Тема 1. Введение в предмет. Обзор средств создания серверных приложений.</i>	4					12				24	Отчет по лабораторному практикуму /25
<i>Тема 2. Основы программирования на PHP</i>	16					14				25	Отчет по лабораторному практикуму /25
<i>Тема 3. Функции. Программирование взаимодействия с сервером на PHP</i>	12					14				25	Отчет по лабораторному практикуму /25
<i>Тема 4. Основы работы с базами данных в web-приложениях</i>	4					14				25	Отчет по лабораторному практикуму /25
<b>Всего:</b>	<b>36</b>					<b>54</b>				<b>99</b>	<b>100</b>
<b>Контроль, час</b>	<b>27</b>									<b>Экзамен</b>	
<b>Объем дисциплины (в академических часах)</b>	<b>216</b>										
<b>Объем дисциплины (в зачетных единицах)</b>	<b>6</b>										

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### ***Тема 1. Введение в предмет. Обзор средств создания серверных приложений.***

Программирование на стороне клиента и сервера. Различия в программировании серверной и клиентской части. Инструменты и технологии программирования. Принципы взаимодействия web-сервера и браузера. Этапы создания веб-приложений. Выбор языка программирования для серверной части. Популярные языки программирования бекэнда. PHP и JavaScript: достоинства и недостатки. Node.js. Интерфейсные среды разработки веб-приложений. Применение языка ASP.NET. Локальные сервера XAMPP и OpenServer.

### ***Тема 2. Основы программирования на PHP***

Особенности применения языка PHP. Фреймворки веб-приложений Laravel. Встраивание скрипта на PHP в HTML код. Переменные и константы: особенности и различия. «Волшебные константы». Область видимости переменных. Типы данных. Обработка строковых переменных. Операторы, приоритеты и ассоциативность операторов. Массивы. Условные операторы и циклы. Объекты. Управляющие конструкции. Команды ветвления. Циклы, циклы обработки массивов. Функции пользовательские. Аргументы функций. Возврат значений функцией. Встроенные функции. Функции даты и времени.

### ***Тема 3 Функции. Программирование взаимодействия с сервером на PHP.***

Функции работы с файловой системой. Формы. Методы передачи данных форм на сервер. Способы передачи параметров сценарию. Особенности использования html-форм. Функции обработки данных форм на сервере. PHP и Upload. Multipart-формы. Обработка multipart-форм. Сценарии загрузки файлов на сервер. Cookie: применение и ограничения. Программирование Cookie. Чтение значений. Массив значений Cookie. Удаление Cookie. Буферизация. HTTP-Аутентификация. Сессии. Отличие Сессий и Cookie. Использование сессий. Создание, удаление сессий. Менеджеры изменений и различий; командные инструменты для обмена кодом.

### ***Тема 4. Основы работы с базами данных в web-приложениях.***

Программирование Базы Данных MySQL. Определение кода непосредственно в теле web-страницы. Подключение внешних скриптов. Функции и переменные для обработки запросов. Взаимодействие PHP и СУБД. Выбор СУБД. Алгоритм работы с базой данных. Соединение с сервером MySQL, обработка запроса, извлечение и хранение данных. Классическое и современное подключение к серверу. Использование шаблонов. Использование транзакций. Последующее использование

полученных данных в теле страницы. Ввод данных в базу данных с помощью html-форм.

Инструменты для совместной работы. Инструменты тестирования скорости сайта. Сообщества веб-разработки.

## **5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

В процессе изучения данной дисциплины используются такие виды учебной работы, как лекции, лабораторные работы, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков использования профессиональной лексики, закрепление практических профессиональных компетенций, поощрение интеллектуальных инициатив.

### ***Методические указания для обучающихся при работе над конспектом лекций во время проведения лекции***

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект, что позволит впоследствии вспомнить изученный учебный материал, дополнить содержание при самостоятельной работе с литературой, подготовиться к экзамену.

Следует также обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Любая лекция должна иметь логическое завершение, роль которого выполняет заключение. Выводы по лекции подытоживают размышления преподавателя по учебным вопросам. Формулируются они кратко и лаконично, их целесообразно записывать. В конце лекции, обучающиеся имеют возможность задать вопросы преподавателю по теме лекции.

### ***Методические рекомендации для обучающихся по выполнению лабораторных практикумов***

Лабораторные практикумы выполняются в соответствии с рабочим учебным планом при последовательном изучении тем дисциплины.

*Порядок проведения практикума.*

1. Получение задания и рекомендаций к выполнению практикума.
2. Настройка инструментальных средств, необходимых для

выполнения практикума.

3. Выполнение заданий практикума.
4. Подготовка отчета в соответствии с требованиями.
5. Сдача отчета преподавателю.

В ходе выполнения практикума необходимо следовать технологическим инструкциям, использовать материал лекций, рекомендованных учебников, источников интернета, активно использовать помощь преподавателя на занятии.

*Требования к оформлению результатов практикумов.*

При подготовке отчета: изложение материала должно идти в логической последовательности, отсутствие грамматических и синтаксических ошибок, шрифт Times New Roman, размер – 14, выравнивание по ширине, отступ первой строки – 1,25, междустрочный интервал – 1,5, правильное оформление рисунков (подпись, ссылка на рисунок в тексте).

При подготовке презентации: строгий дизайн, минимум текстовых элементов, четкость формулировок, отсутствие грамматических и синтаксических ошибок, воспринимаемая графика, умеренная анимация.

### ***Методические указания для обучающихся по организации самостоятельной работы***

Самостоятельная работа обучающихся направлена на самостоятельное изучение отдельных тем/вопросов учебной дисциплины.

Самостоятельная работа является обязательной для каждого обучающегося, ее объем по дисциплине определяется учебным планом.

При самостоятельной работе обучающиеся взаимодействуют с рекомендованными материалами при минимальном участии преподавателя.

*Работа с литературой (конспектирование)*

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 обучающихся свое отношение к конкретной проблеме.

Изучая материал по учебной книге (учебнику, учебному пособию, монографии, и др.), следует переходить к следующему вопросу только после полного уяснения предыдущего, фиксируя выводы и вычисления (конспектируя), в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода.

Особое внимание обучающийся должен обратить на определение основных понятий курса. Надо подробно разбирать примеры, которые поясняют определения. Полезно составлять опорные конспекты.



Выводы, полученные в результате изучения учебной литературы, рекомендуется в конспекте выделять, чтобы при перечитывании материала они лучше запоминались.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса.

Вопросы, которые вызывают у обучающегося затруднение при подготовке, должны быть заранее сформулированы и озвучены во время занятий в аудитории для дополнительного разъяснения преподавателем.

### ***Навигация для обучающихся по самостоятельной работе в рамках изучения дисциплины***

<b>Наименование темы</b>	<b>Вопросы, вынесенные на самостоятельное изучение</b>	<b>Формы самост. работы</b>	<b>Форма текущего контроля</b>
<b><i>Тема 1. Введение в предмет. Обзор средств создания серверных приложений.</i></b>	Инструменты и технологии клиент-серверного программирования. интерфейсные среды разработки веб-приложений Принципы взаимодействия web-сервера и браузера. Локальные сервера XAMPP и OpenServer. Основы конфигурирования web-сервера Apache.	Работа с литературой, включая ЭБС, источниками в сети Internet Подготовка к лабораторному практикуму, подготовка отчета по практикуму	Отчет по лабораторному практикуму
<b><i>Тема 2. Основы программирования на PHP</i></b>	Фреймворки веб-приложений Laravel Создание динамического веб-сайта с контекстной поддержкой пользователя. Основы аутентификации пользователей. «Волшебные константы». Приоритеты и ассоциативность операторов.	Работа с литературой, включая ЭБС, источниками в сети Internet Подготовка к лабораторному практикуму, подготовка отчета по практикуму	Отчет по лабораторному практикуму
<b><i>Тема 3 Функции. Программирование взаимодействия с сервером на PHP.</i></b>	PHP и Upload. Multipart-формы. Обработка multipart-форм. Сценарии загрузки файлов на сервер.	Работа с литературой, включая ЭБС, источниками в сети Internet Подготовка к	Отчет по лабораторному практикуму

Наименование темы	Вопросы, вынесенные на самостоятельное изучение	Формы самост. работы	Форма текущего контроля
	Буферизация. HTTP- Аутентификация.	лабораторному практикум. подготовка отчета по практикуму	
<i>Тема 4. Основы работы с базами данных в web-приложениях.</i>	Выбор СУБД. Использование шаблонов. Использование транзакций. Сообщества веб-разработки Stack Overflow, Front-end Front, Hashnode.	Работа с литературой, включая ЭБС, источниками в сети Internet Подготовка к лабораторному практикуму, подготовка отчета по практикуму	Отчет по лабораторному практикуму

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 6.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

#### *Основная литература:*

1. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2021. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/466449>

2. Мейер, Б. Инструменты, алгоритмы и структуры данных / Б. Мейер. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 543 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429033>

3. Шабашов, В.Я. Организация доступа к данным из РНР приложений для различных СУБД – Москва ; Берлин : Директ-Медиа, 2019. – 121 с. <http://biblioclub.ru/index.php?page=book&id=499185>

#### *Дополнительная литература:*

1. Сысолетин, Е. Г. Разработка интернет-приложений : учебное пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под научной редакцией Л. Г. Доросинского. — Москва : Издательство Юрайт, 2021. — 90 с. — (Высшее образование). — ISBN 978-5-9916-9975-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472200>

2. Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 265 с. : ил. ; То же [Электронный

ресурс]. - URL: <http://biblioclub.ru/>

3. Гениатулина, Е.В. CMS – системы управления контентом : учебное пособие / Е.В. Гениатулина ; Министерство образования и науки Российской Федерации, Новосибирский государственный технический университет. - Новосибирск : НГТУ, 2015. - 63 с. : ил. - Библиогр. в кн. - ISBN 978-5-7782-2696-8 ; То же [Электронный ресурс].

- URL: <http://biblioclub.ru/>

4. Строганов, А.С. Ваш первый сайт с использованием PHP-скриптов – Москва : Диалог-МИФИ, 2015. – 288 с. : <http://biblioclub.ru/index.php?page=book&id=447998>

5. Вагин, Д.В. Современные технологии разработки веб-приложений. – Новосибирск : Новосибирский государственный технический университет, 2019. – 52 с. <http://biblioclub.ru/index.php?page=book&id=573960>

6. Малашкевич, В.Б. Интернет-программирование : лабораторный практикум / В.Б. Малашкевич ; Поволжский государственный технологический университет. – Йошкар-Ола : Поволжский государственный технологический университет, 2017. – 96 с. : ил. – Режим доступа: по подписке. –

URL: <http://biblioclub.ru/index.php?page=book&id=476400>

## 6.2. Перечень ресурсов информационно-коммуникационной сети «Интернет»

№	Наименование ресурса	Ссылка
1.	Язык программирования JavaScript.	<a href="https://learn.javascript.ru/">https://learn.javascript.ru/</a>
2.	Современные обновления языка PHP	<a href="http://php.net">Http://php.net</a>
3.	Описание языка PHP	<a href="http://www.php.ru">http://www.php.ru</a>
4.	Сервер Апач	<a href="http://www.apache.org">http://www.apache.org</a>
5.	Все для веб-разработки	<a href="https://techblog.sdstudio.top">https://techblog.sdstudio.top</a>
6.	Сообщество IT-специалистов	<a href="https://habr.com/ru/all/">https://habr.com/ru/all/</a>
7.	Телеграмм-канал веб-разработчиков	<a href="https://t.me/phpGeeksJunior">https://t.me/phpGeeksJunior</a>

## 6.3. Описание материально-технической базы

Материально-техническое обеспечение дисциплины включает в себя:

Учебная аудитория (Лаборатория информационно-коммуникационных технологий), оборудованная: комплекты специализированной учебной мебели, мультимедийный проектор, экран, доска классная, принтер, компьютер преподавателя и компьютеры обучающихся с выходом в сеть «Интернет», доступом в электронную информационно-образовательную среду.

Помещение для самостоятельной работы обучающихся – аудитория оборудованная:

комплекты специализированной учебной мебели, мультимедийный проектор, экран, доска классная, компьютеры с выходом в сеть «Интернет» и доступом в электронную информационно-образовательную среду.

**6.4. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, в том числе комплект лицензионного программного обеспечения, электронно-библиотечные системы, современные профессиональные базы данных и информационные справочные системы**

Обучающиеся обеспечены доступом к электронной информационно-образовательной среде из любой точки, в которой имеется доступ к сети «Интернет», как на территории организации, так и вне ее.

***лицензионное программное обеспечение:***

- Windows (зарубежное, возмездное);
- MS Office (зарубежное, возмездное);
- Adobe Acrobat Reader (зарубежное, свободно распространяемое);
- КонсультантПлюс: «КонсультантПлюс: Студент» (российское, свободно распространяемое);
- 7-zip – архиватор (зарубежное, свободно распространяемое);
- Comodo Internet Security (зарубежное, свободно распространяемое);
- MySQL for Windows – реляционная система управления базами данных (зарубежное, свободно распространяемое);
- Apache NetBeans – свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и ряда других (зарубежное, свободно распространяемое);
- Android Studio – разработка мобильных приложений (зарубежное, свободно распространяемое)

***электронно-библиотечная система:***

- Электронная библиотечная система (ЭБС) «Университетская библиотека ONLINE» <http://biblioclub.ru/>.
- Образовательная платформа «Юрайт». Для вузов и ссузов. Электронная библиотечная система (ЭБС) <https://urait.ru/>

***современные профессиональные баз данных:***

- Официальный интернет-портал базы данных правовой информации <http://pravo.gov.ru>.

• Портал Единое окно доступа к образовательным ресурсам <http://window.edu.ru/>

**информационные справочные системы:**

• Портал Федеральных государственных образовательных стандартов высшего образования <http://fgosvo.ru>.

• Компьютерная справочная правовая система «КонсультантПлюс» (<http://www.consultant.ru/>).

## **7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

### **7.1. Описание оценочных средств для проведения текущего контроля успеваемости в процессе освоения дисциплины**

<b>№ п/п</b>	<b>Наименование оценочного средства</b>	<b>Шкала и критерии оценки, балл</b>
1.	Лабораторный практикум	25-20 – работа и отчет выполнены в срок, самостоятельно, правильно поняты и использованы соответствующие формулы, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, правильно выбраны совместимые комплектующие, сделаны необходимые выводы, хорошо аргументированы, даны исчерпывающие ответы на все поставленные вопросы; 19-14 – работа и отчет выполнены в срок, самостоятельно, правильно поняты и использованы соответствующие формулы, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, правильно выбраны совместимые комплектующие, необходимые выводы сделаны частично, хорошо аргументированы, даны ответы на все поставленные вопросы; 13-5 – работа и отчет выполнены в срок, в основном самостоятельно, использованы соответствующие формулы; определены соответствующие спецификации, имеются ошибки в расчетах; выбраны совместимые комплектующие необходимые, выводы сделаны частично, слабо аргументированы, даны ответы не на все вопросы; 4-1 – обучающийся подготовил работу и отчет несамостоятельно или не завершил в срок, описание спецификации содержит незначительные ошибки, выводы и ответы на вопросы отсутствуют.

***Типовые контрольные задания или иные материалы в рамках  
текущего контроля успеваемости***

**Типовые задания к лабораторным практикумам:**

***Лабораторный практикум № 1.***

**Задание № 1.** *Создание динамических страниц с помощью JavaScript.*

1. Напишите программу для создания на странице таблицы. С помощью prompt пользователь вводит количество строк, столбцов и заголовков таблицы. Ячейки таблицы оставьте пустыми.

2. Напишите программу для создания списка на странице. Для каждого пункта запрашивайте содержимое пункта у пользователя с помощью prompt. Созданный пункт списка добавляйте к UL.

3. Напишите программу для создания на странице таблицы. С помощью любого элемента управления на странице (чек-бокса, радиокнопки, выпадающего меню) пользователь выбирает что он будет вводить заголовки столбцов или строк. А затем с помощью вводит содержимое первой строки или первого столбца. Остальные ячейки оставьте пустыми. Пустых строк/столбцов достаточно 2.

4. Напишите код для проверки, пуст ли элемент elem. «Пустой» – значит нет дочерних узлов, даже текстовых.

5. Напишите код, который выделит все четные строки в таблице.

***Лабораторный практикум № 2.***

**Задание № 1.** *Основы PHP.*

Написать скрипт на PHP.

1. Нарисовать таблицу умножения

а) Создайте две числовые переменные cols и rows

б) Присвойте созданным переменным произвольные значения в диапазоне от 1 до 10

в) Используя циклы, отрисуйте таблицу умножения в виде HTML-таблицы на следующих условиях:

- Число столбцов должно быть равно значению переменной cols
- Число строк должно быть равно значению переменной rows
- Ячейки на пересечении столбцов и строк должны содержать значения, являющиеся произведением порядковых номеров столбца и строки

г) Рекомендуется использовать циклы for

д) Значения в ячейках первой строки и первого столбца должны быть отрисованы полужирным шрифтом

е) Фонový цвет ячеек первой строки и первого столбца должен быть отличным от фонового цвета таблицы.

2. Нарисовать (добавить) навигационное меню на странице, типа: `<a href="contact.php">Contact</a>`

а) Создайте ассоциативный массив `$menu`

б) Заполните массив, соблюдая следующие условия:

- Индексом элемента массива является пункт меню, например:

Home, About, Contact...

- Значением элемента массива является имя файла, на который будет указывать ссылка, например: `index.php`, `about.php`, `contact.html`...

в) Используя цикл `foreach` отрисуйте вертикальное меню, структура которого описана в массиве `$menu`

3. Дан числовой массив. Массив задается случайным образом.

Проверить, есть ли в заданном числовом массиве отрицательные числа (хотя бы одно). Если есть — выводить «да», если нет — выводить «нет». Выводить сообщение один раз. Для проверки вывести исходный массив.

4. Создать двумерный массив размерностью 3 x 3. Заполнить его значениями для первой строки: «1\_1», «1\_2», «1\_3»; для второй строки: «2\_1», «2\_2», «2\_3» и т.д.. Вывести на экран

5. Дано объявление и несколько разных людей, которым нужно это объявление отправить. Для этого сделать заготовку с содержанием объявления, внутри которого есть ряд изменяющихся параметров: массива имен людей и массив событий. Вывести один вариант итогового объявления на экран. Для задания массивов использовать и ассоциативный массив.

Используйте шаблон объявления и ориентируйтесь по цветам: красным – массивы, коричневым – числовая переменная, синим – константа.

*Например:*

*Уважаемый (ая), Иван Иванович!*

*Приглашаем Вас на день открытых дверей.*

*Дата события: 12 мая.*

*С уважением, Василий.*

### **Лабораторный практикум № 3.**

**Задание № 1. PHP. Использование функций.**

Написать скрипт на языке программирования PHP.

1. Создайте строковую переменные `now` и `birthday`. Присвойте переменной `now` значение метки времени актуальной даты (сегодня), переменной `birthday` значение метки времени Вашего дня рождения. Выведите фразу «До моего дня рождения осталось », количество секунд, оставшееся до Вашего дня рождения, закончите фразу » секунд».

2. Написать функцию с двумя аргументами, возвращающую результат возведения числа в степень. Вывести для примера результат выполнения функции в формате «А в степени В = N», для переменных заданных в программе. Примечание: степень в `php` вычисляется при

помощи функции row()

row (\$a, \$b); где \$a — число, \$b — степень

3. Создайте строковую переменную now, присвойте переменной now значение метки времени актуальной даты (сегодня). Создайте переменную \$H и присвойте ей значение часов, выделенных из значения переменной \$now. Создайте переменную \$i и присвойте ей значение минут, выделенных из значения переменной \$now. Показывать текущее время в виде: ... часов...мин . Изменять окончание слова *часов/часа/час* в зависимости от кол-ва часов и минут.

### **Задание № 2. PHP. Использование Cookie и сессий.**

1. Создать cookie, установить значение – ваше имя. Написать код с проверкой передачи и выводом данного cookie на экран. Добавить текстовое поле для вывода значения cookie в нем.

2. Выводите на экран количество посещений страницы, используя cookie. Предлагаемый алгоритм (возможен другой вариант выполнения):

а) Установите переменную для счетчика (\$counter), обнулив ее.

б) Проверьте, установлен ли уже cookie, если да — то присвойте переменной \$counter значение cookie (см. пункт 3).

в) Добавьте cookie для хранения количества посещений.

г) Приращивайте счетчик.

д) Проверьте, установлен ли уже cookie, если да — то выводите значение cookie.

3. Создать массив данных для хранения паролей. Значения паролей сохранить в cookie. Пароли вводить из формы по нажатию кнопки.

а) В html-коде предусмотреть кнопку, по нажатию которой формировать страницу, на которую добавить текстовые поля, выводить в них значения всех введенных паролей.

б) В html-коде предусмотреть кнопку, по нажатию которой удалить cookie, при этом повторно обновить браузер.

4. Создать файл cookie.php.

а) При первом запросе страницы cookie.php пользователем, выводить фразу «Добро пожаловать!»

б) При повторных запросах:

в) Выводить количество посещений. Например: «Вы зашли на страницу 5 раз»

г) Указывать дату и время последнего посещения. Например: «Последнее посещение: 20-08-2009 16:34:45»

5. Создать файл header.php с заголовком второго уровня «Здравствуй!». При помощи директивы header отображать данную страницу в текстовом формате. Использовать на этой же странице директиву для перехода через 5 секунд на какую-либо другую страницу

### **Лабораторный практикум № 4.**

#### **Задание № 4. Связь скриптов PHP с MySQL.**



Написать скрипт на языке программирования PHP.

1. Создать БД с тремя полями:

- адрес электронной почты (50 символов),
- текст сообщения (250 символов),
- дата и время отправления.

Написать программу, с помощью которой пользователи могут заполнить эту БД. Написать программу, выводящую информацию из БД

2. Написать систему хранения книг в БД. Хранить только названия книг и авторы. Предложите структуру таблиц. Учтите, что книга может быть написана несколькими соавторами.

Написать программу, выводящую информацию из БД

Получить список книг, которые написаны 3-мя соавторами. То есть получить отчет «книга — количество соавторов» и отфильтровать те, у которых соавторов меньше 3х.

3. Построить базу: фильмы и страны, у одного фильма может быть много стран производителей.

Написать программу, выводящую информацию из БД

Нерадивый программист удалял фильмы, но не подчистил таблицу связи. Надо найти и удалить все мусорные записи, которые остались в таблице связи.

## 7.2. Описание оценочных средств для проведения промежуточной аттестации

Промежуточная аттестация по дисциплине «Разработка серверных приложений для Web» проводится в форме экзамена.

Процедура оценивания	Шкала и критерии оценки, балл
Экзамен представляет собой выполнение обучающимся заданий билета, включающего в себя:  Задание №1 – теоретический вопрос на знание базовых понятий предметной области дисциплины, а также позволяющий оценить степень владения обучающимся принципами предметной области дисциплины, понимание их особенностей и взаимосвязи между ними: Задание №2 – задание на анализ ситуации из предметной области дисциплины и выявление способности обучающегося выбирать и применять соответствующие принципы и методы решения практических проблем, близких к	Выполнение обучающимся заданий оценивается по следующей балльной шкале: Задание 1: 0-30 баллов Задание 2: 0-30 баллов Задание 3: 0-40 баллов  <b>-90 и более (отлично)</b> – ответ правильный, логически выстроен, приведены необходимые формулы, использована профессиональная лексика. Задача решена правильно. Обучающийся правильно интерпретирует полученный результат. <b>-70 и более (хорошо)</b> – ответ в целом правильный, логически выстроен, приведены необходимые формулы, использована профессиональная лексика. Ход решения задачи правильный, ответ неверный. Обучающийся в целом правильно интерпретирует полученный результат. <b>-50 и более (удовлетворительно)</b> – ответ в основном правильный, логически выстроен, приведены не все необходимые формулы, использована профессиональная лексика. Задача решена частично. <b>-Менее 50 (неудовлетворительно)</b> – ответы на теоретическую часть неправильные или неполные.

Процедура оценивания	Шкала и критерии оценки, балл
профессиональной деятельности; Задание №3 – задание на проверку умений и опыта, полученных в результате освоения дисциплины	Задача не решена

### ***Типовые задания для проведения промежуточной аттестации обучающихся***

#### ***Задания на знания***

1. Программирование серверной части сайта. Статические и динамические сайты. Структура динамического запроса.
2. Типичные применения и выгоды бэкэнда
3. Этапы создания веб-приложения.
4. Структура веб-приложения.
5. Средства создания активных веб-страниц.
6. Схема работы Web-приложения, архитектура клиент-сервер.

#### ***Протокол HTTP***

7. Выбор метода: GET или POST. Кодирование в URL
8. О взаимодействии PHP и HTML
9. Режимы работы PHP-приложений. Выбор режима работы.
10. Запуск приложения с правами пользователя wwwrun
11. Авторизация на сайте с использованием сессий, способы, защита одной страницы, защита нескольких страниц
12. Cookie, назначение, время жизни.
13. Особенности использования сессий и cookie в современных стандартах веб-приложений.
14. Перспективы сохранения контекста пользователя.
15. Обращения к СУБД MySQL через функции PHP
16. Обработка ошибок MySQL
17. Область использования .htaccess.
18. Выполнение запросов к базе данных.
19. Включение файлов, принцип работы include
20. Загрузка файлов на сервер (upload)
21. Общие сведения о PHP. Основы синтаксиса.
22. Обработка ошибок MySQL. Выполнение запросов к базе данных. Получение результата.
23. Поддержка и оптимизация. Поисковая оптимизация (SEO), её автоматизация
24. Создание гостевой книги в PHP с хранением информации в файлах.
25. CMS и их использование в web-технологиях
26. Многофайловые сценарии PHP
27. Фреймворки веб-приложений
28. Менеджеры изменений и различий Diffchecker и Beyond Compare

29. Командные инструменты для обмена кодом JS Bin и CodeShare
30. Инструменты для совместной работы
31. Инструменты тестирования скорости сайта
32. Сообщества веб-разработки

### ***Задания на умения***

1. Для чего предназначен плагин (plug-in): функции и назначение.
2. Какую принципиальную проблему решило создание FTP?
3. Охарактеризуйте документ из серии пронумерованных информационных документов Интернета, содержащих технические спецификации и Стандарты
4. Что такое иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети
5. Перечислите варианты, когда протокол работает независимо от операционных систем
6. Перечислите варианты, когда протокол позволяет обмениваться файлами и выполнять операции над ними через TCP-сети
7. Перечислите варианты, когда протокол работает в соответствии с принципами архитектуры «клиент-сервер»
8. Перечислите варианты, когда поставщик услуг Интернета выполняет транспортную функцию для конечных пользователей
9. Программы, выполняющиеся на сервере это программы, в которых\_\_\_?
10. Перечислите варианты, когда код программы не передается клиенту, а выполняется на веб-сервере
11. Перечислите варианты, когда код программы не передается клиенту, а выполняется на веб-сервере
12. Перечислите варианты, когда результаты работы программа в виде HTML-документа передаются клиент-машине
13. Перечислите варианты, когда исходный текст программы представляет собой часть веб-страницы
14. Что такое IP-адрес?
15. За счет чего в сети Интернет достигается надежность передачи данных?
16. Какую роль при передаче данных играет понятие «доменное имя»?
17. Перечислите принципиальные проблемы, решаемые созданием FTP?
18. Назовите принципиальные отличия понятий «тонкий» и «толстый» клиент?
19. Чем отличаются URI и URL?
20. Перечислите отличия файл-серверной и клиент-серверной архитектуры?
21. Перечислите способы встраивания php-скриптов на страницу html?

22. Перечислите способы описания строковых данных на php
23. Назовите отличия способов описания строковых данных на php
24. Перечислите способы объявления массивов на php? Чем они отличаются?
25. Назовите общие черты и отличия объявления строк и массивов на php?
26. Перечислите суперглобальные массивы, в которых хранятся данные при передаче данных на сервер через html-формы?
27. Перечислите выгоды использования cookie.
28. Назовите принципиальные отличия использования cookie и сессий?
29. В чем выгода применения фреймворка веб-приложений Laravel?
30. Для чего используются менеджеры изменений и различий Diffchecker и Beyond Compare?
31. Для чего используются командные инструменты для обмена кодом JS Bin и CodeShare
32. Какие функции у инструментов для совместной работы?
33. Зачем применяются инструменты тестирования скорости сайта?
34. Какие параметры измеряются при тестировании скорости сайта?
35. Какие вопросы решаются с помощью сообществ веб-разработки?

### ***Задания на навыки***

Написать код программы на PHP, составить описание интерфейса программы и описание объектных моделей кода

1. Опишите функцию `getTable()`. Задайте для функции три аргумента: `cols`, `rows`, `color`. Скопируйте код, отрисовывающий таблицу умножения из предыдущей лабораторной работы, вставьте скопированный код в тело функции `getTable()`. Измените код таким образом, чтобы таблица отрисовывалась в зависимости от входящих параметров `cols`, `rows` и `color`. Отрисуйте таблицу умножения вызывая функцию `getTable()` с различными параметрами.

2. Создайте строковую переменные `now` и `birthday`. Присвойте переменной `now` значение метки времени актуальной даты (сегодня), переменной `birthday` значение метки времени Вашего дня рождения. Выведите фразу «До моего дня рождения осталось », количество секунд, оставшееся до Вашего дня рождения, закончите фразу » секунд».

3. Написать функцию с двумя аргументами, возвращающую результат возведения числа в степень.

4. Примечание: степень в php вычисляется при помощи функции `pow()` `pow ($a, $b)`; где `$a` — число, `$b` — степень

5. Создайте строковую переменную `now`, присвойте переменной `now` значение метки времени актуальной даты (сегодня). Создайте переменную `$H` и присвойте ей значение часов, выделенных из значения переменной `$now`. Создайте переменную `$i` и присвойте ей значение минут, выделенных из значения переменной `$now`. Показывать текущее время в виде: ... часов...мин . Изменять окончание слова *часов/часа/час* в зависимости от кол-ва часов и минут.

6. Найти значение переменной  $z$ , заданной суммой функций. Для нахождения значения функции  $f(u,t)$  написать пользовательскую функцию. Числа  $a$  и  $b$  – случайные.

7. Создать функцию, которая выводит текст с разным размером шрифта. Функция имеет два аргумента: текстовая строка и размер шрифта.

8. Написать программу-календарь для любого месяца между 2000 и 2020 годом. Пользователь выбирает месяц и день из выпадающих списков, а даты для выбранного месяца выводятся в виде таблицы-календаря.

9. Создать функцию, которая принимает четыре текстовые строки, а выводит html-код таблицы, где каждая из исходных строк расположена в отдельной ячейке.

10. Написать программу, которая читает данные из файла и выводит на экран, разделив на пронумерованные строки.

11. Создать файл, каждая запись которого имеет структуру: название товара, количество товара, цена за единицу. Выдать ведомость товара, количество которого меньше заданного значения  $A$