

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Смоленский государственный университет»

Кафедра аналитических и цифровых технологий

«УТВЕРЖДАЮ»
Проректор по учебно-методической работе
_____ Ю.А. Устименко
«30» июня 2022 г.

**Рабочая программа дисциплины
Б1.В.06 Основы web-разработки и web-дизайна**

Направление подготовки: 38.03.02 Менеджмент
Направленность (профиль): Цифровой маркетинг и бренд-менеджмент
Форма обучения: очная
Курс – 3
Семестр – 6
Всего зачетных единиц – 5, часов – 180.

Форма отчетности: экзамен – 6 семестр, курсовой проект – 6 семестр.

Программу разработал
кандидат физико-математических наук, доцент Д.С. Букачев

Одобрена на заседании кафедры аналитических и цифровых технологий
«23» июня 2022 года, протокол № 10

Заведующий кафедрой _____ Д.С. Букачев

Смоленск
2022

1. Место дисциплины в структуре ОП

Дисциплина Б1.В.06 Основы web-разработки и web-дизайна относится к обязательным дисциплинам вариативной части образовательной программы подготовки 38.03.02 Менеджмент, направленность (профиль): Цифровой маркетинг и бренд-менеджмент.

Изучение дисциплины предполагает сочетание фундаментальной подготовки с освоением технологии применения специализированных программных продуктов и систем, ориентированных на защиту экономической и служебной информации, базируется на компетенциях, сформированных при изучении дисциплины «Информатика», «Информационные технологии в менеджменте».

Изучение дисциплины подразумевает формирование у студентов целостного представления о современных информационных технологиях, анализе современных процессов проектирования и разработки программных продуктов, принципах и методах разработки программных решений, функционирующих на базе web-технологий, а также обеспечение подготовки бакалавра, способного осуществлять анализ требований, разработку, оценку качества и тестирование программных web-решений и решать следующие профессиональные задачи:

- программирование, тестирование и документирование приложений;
- автоматизированное решение прикладных задач операционного и аналитического характера;
- анализ и выбор методов и средств автоматизации и информатизации прикладных процессов на основе современных информационно-коммуникационных технологий.

2. Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
ПК-3. Способен проводить анализ, разработку и осуществление маркетинговой стратегии, стратегии развития бренда и медийной стратегии продвижения в информационно-коммуникационной сети «Интернет», направленной на обеспечение конкурентоспособности	Знать: основные базовые понятия стратегического менеджмента и основы осуществления маркетинговых стратегий; основные направления представления компании в сети Интернет, теоретические основы интернет-рекламы, ключевые аспекты поисковой оптимизации сайта, основы SMM-маркетинга и контекстной рекламы; основы веб-программирования и веб-дизайна; международное и российское законодательство в сфере защиты информации; основные способы безопасного хранения данных. Уметь: проводить анализ и разрабатывать маркетинговые стратегии, стратегии развития бренда и медийной стратегии продвижения в информационно-коммуникационной сети «Интернет»; использовать процессы внедрения информационных технологий для представления компании в интернет-среде, реализовывать организацию веб-представительства компании и его сопровождение в сети; разрабатывать и внедрять рекламные стратегии в сети Интернет, анализировать эффективность маркетинговых инструментов в Интернет-среде; разрабатывать динамические веб-страницы, проектировать дизайн веб-страниц, управлять веб-контентом;

	<p>оценивать ущерб от угроз информационной безопасности и осуществлять мероприятия, направленные на профилактику правонарушений в сфере информационной безопасности.</p> <p>Владеть: навыками стратегического мышления и практического осуществления маркетинговой стратегии, стратегии развития бренда и медийной стратегии продвижения в информационно-коммуникационной сети «Интернет»; навыками работы с информационными средствами для управления различными направлениями деятельности организации в сети Интернет с целью повышения её конкурентоспособности; навыками программирования, разметки и дизайна веб-страниц с использованием специализированных программных средств; навыками обеспечения локальной и сетевой информационной безопасности, программными средствами журналирования и анализа событий безопасности.</p>
<p>ПК-5. Способен управлять продуктовыми исследованиями, сопровождать дизайн, разработку и выпуск продукта; использовать инструменты проектного управления в продуктивном менеджменте и брендинге</p>	<p>Знать: виды и современные методы проведения маркетингового исследования; специфику сбора, измерения и анализа маркетинговой информации; основные механизмы управления продуктовыми исследованиями.</p> <p>Уметь: обрабатывать и анализировать результаты проведенных маркетинговых исследований; определять виды, методы, формы проведения маркетинговых исследований, применяемые в продуктивном менеджменте и брендинге; использовать инструменты проектного управления в продуктивном менеджменте и брендинге.</p> <p>Владеть: методологией маркетинговых исследований; современными инструментами маркетинговых исследований, применяемыми для разработки управленческих решений в продуктивном менеджменте и брендинге; навыками разработки и поддержки маркетинговых и брендинговых продуктов с использованием технологий баз данных, специализированных программных продуктов подготовки и управления контентом.</p>

3. Содержание дисциплины

Тема 1. Технологии стороны клиента. Сценарии и обработка событий. JavaScript.

Сценарий и обработка события.

События в динамическом HTML. Связывание кода с событиями. Создание сценария. Внедрение сценария в HTML.

JavaScript как основной язык сценариев для Web. Сферы использования JavaScript.

Основные идеи JavaScript. Структура JavaScript программы. Типовые примеры использования JavaScript-сценариев.

JavaScript. Базовые элементы языка. Основные объекты языка.

Синтаксис JavaScript. Переменные. Операции. Управляющие структуры и организация циклов. Функции. Объектная модель JavaScript. Обработка событий.

Объектная модель браузера и документа. Иерархия объектов браузера. Объект window. Свойства, методы и события объекта window. Объект document. Свойства, методы и события объекта document. Объект screen. Свойства, методы и события объекта screen.

Тема 2. Серверные технологии. PHP.

Web-серверы: назначение, принцип работы, виды серверов. Web-сервер Apache. Установка, настройка файлов конфигурации.

Динамические web-технологии. Синтаксис языка PHP. Формы. Компоновка и дизайн форм. Назначение формы. Создание формы. Текстовые поля. Текстовые области. Переключатели, Флажки. Раскрывающиеся списки. Отправка данных формы на сервер. Организация ветвлений. Применение ветвлений для обработки форм.

Определение массива. Численно индексированные массивы. Ассоциативные массивы. Многомерные массивы. Сортировка массивов. Некоторые другие операции с массивами. Понятие наследования. Реализация наследования в PHP. Перекрытие. Многоуровневое наследование. Множественное наследование.

Тема 3. Технологии создания web-сайта. Базы данных MySQL.

Функции для работы с базами данных. Получение данных из базы данных. Сохранение данных в базе данных.

4. Тематический план

№ п/п	Темы	Всего часов	Формы занятий			
			Лекции	Практич. занятия	Лаборатор. занятия	Самостоятельная работа
1.	Технологии стороны клиента. Сценарии и обработка событий. JavaScript.	39	8	0	8	23
2.	Серверные технологии. PHP.	71	14	0	30	27
3.	Серверные технологии. Базы данных MySQL.	43	10	0	10	23
	Подготовка к экзамену	27	0	0	0	27
	Всего за семестр	180	32	0	48	100

5. Виды образовательной деятельности

Занятия лекционного типа

Тема 1. Технологии стороны клиента. Сценарии и обработка событий. JavaScript.

Лекция 1. Сценарий и обработка события. События в динамическом HTML. Связывание кода с событиями. Создание сценария. Внедрение сценария в HTML.

Лекция 2. JavaScript как основной язык сценариев для Web. Сферы использования JavaScript.

Основные идеи JavaScript. Структура JavaScript программы. Типовые примеры использования JavaScript-сценариев.

Лекция 3. JavaScript. Базовые элементы языка. Основные объекты языка. Синтаксис JavaScript. Переменные. Операции. Управляющие структуры и организация циклов. Функции. Объектная модель JavaScript. Обработка событий.

Лекция 4. Объектная модель браузера и документа. Иерархия объектов браузера. Объект window. Свойства, методы и события объекта window. Объект document. Свойства, методы и события объекта document. Объект screen. Свойства, методы и события объекта screen.

Тема 2. Серверные технологии. PHP.

Лекция 5. Web-серверы: назначение, принцип работы, виды серверов. Web-сервер Apache. Установка, настройка файлов конфигурации.

Лекция 6. Динамические web-технологии. Синтаксис языка PHP. Формы. Компоновка и дизайн форм. Назначение формы. Создание формы. Текстовые поля. Текстовые области.

Лекция 7. Переключатели, Флажки. Раскрывающиеся списки. Отправка данных формы на сервер.

Лекция 8. Организация ветвлений. Применение ветвлений для обработки форм.

Лекция 9. Определение массива. Численно индексированные массивы. Ассоциативные массивы. Многомерные массивы.

Лекция 10. Сортировка массивов. Некоторые другие операции с массивами.

Лекция 11. Понятие наследования. Реализация наследования в PHP. Перекрытие. Многоуровневое наследование. Множественное наследование.

Тема 3. Технологии создания web-сайта. Базы данных MySQL.

Лекция 12. Базы данных MySQL. Особенности настройки и работы сервера.

Лекции 13-14. Функции для работы с базами данных.

Лекция 15. Получение данных из базы данных.

Лекция 16. Сохранение данных в базе данных.

Занятия семинарского типа (лабораторные занятия)

1. Программирование на стороне клиента. JavaScript. Концепция управления событиями (2 часа).

Цель работы: сформировать представление о программировании на стороне клиента, рассмотреть синтаксис основных конструкций JavaScript, основные событийные процедуры.

Основные задачи: сформировать HTML-документ с поддержкой CSS и обработкой событий внутри HTML-кода.

2. Программирование на стороне клиента. JavaScript. Размещение сценария. Функции в JavaScript (4 часа).

Цель работы: рассмотреть возможности разделения скриптов JavaScript и HTML-кода, реализовать обработку событий во внешних файлах.

Основные задачи: реализовать обработку событий во внешних файлах, решить несколько вычислительных задач (решение квадратного уравнения, получение простого числа по его номеру, получение числа Фибоначчи по его номеру).

3. Программирование на стороне клиента. JavaScript. Обработка форм (2 часа).

Цель работы: сформировать представление о web-формах, рассмотреть события формы, ознакомиться с методикой проверки корректности вводимых данных в виде JavaScript-скриптов.

Основные задачи: сформировать HTML-документ, содержащий web-форму, реализовать проверку корректности данных формы при нажатии на кнопку.

4. Серверный язык PHP. Синтаксис. Включение PHP-сценария в HTML-документ (4 часа).

Цель работы: сформировать представление о программировании на стороне сервера, ознакомиться с синтаксисом PHP.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP.

5. Серверный язык PHP. Организация ветвлений (4 часа).

Цель работы: рассмотреть синтаксис условного оператора PHP.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, содержащими ветвление.

6. Серверный язык PHP. Понятие класса. Основные компоненты класса (4 часа).

Цель работы: рассмотреть понятия «класс» и «объект» и техническую реализацию.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, демонстрирующими возможности ООП.

7. Серверный язык PHP. Абстрагирование, инкапсуляция, модульность и иерархия (4 часа).

Цель работы: рассмотреть понятия «инкапсуляция» и «наследование» в рамках концепции PHP.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, демонстрирующими работу с иерархической системой классов.

8. Серверный язык PHP. Хранение и использование данных пользователя. Способы хранения. Хранение данных в файлах (2 часа).

Цель работы: сформировать представление о работе с данными пользователя, передаче данных между страницами.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, обеспечивающими межстраничное взаимодействие на уровне клиента.

9. Серверный язык PHP. Хранение данных в файлах. Открытие файла. Функция fopen(). Режимы файла. Чтение файла. Запись в файл (2 часа).

Цель работы: сформировать представление о работе с данными пользователя, сохранении и загрузке данных с использованием файлов пользователя.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, обеспечивающими загрузку и выгрузку данных в файлы.

10. Серверный язык PHP. Организация счетчика посещений (2 часа).

Цель работы: сформировать представление о счетчике посещений.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, обеспечивающими реализацию счетчика посещений.

11. Серверный язык PHP. Обработка форм (4 часа).

Цель работы: сформировать представление о работе с формами средствами PHP.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, обеспечивающими обработку данных форм на стороне сервера.

12. Серверный язык PHP. Массивы (4 часа).

Цель работы: сформировать представление о работе с массивами средствами PHP.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, демонстрирующими возможности PHP по работе с массивами.

13. Базы данных MySQL. Базовая архитектура баз данных для web. Транзакция базы данных для web (2 часа).

Цель работы: сформировать представление о работе с базами данных из web-приложений.

Основные задачи: рассмотреть особенности механизма транзакций, способы разрешения коллизий.

14. Базы данных MySQL. Доступ к базе данных из web. Выбор базы данных (2 часа).

Цель работы: сформировать представление о возможностях современных СУБД и их адаптированности к использованию совместно с PHP.

Основные задачи: провести анализ рынка программных средств, обосновать выбор СУБД.

15. Базы данных MySQL. Доступ к базе данных из web. Структура процедуры доступа. Установка соединения (2 часа).

Цель работы: сформировать представление о работе с базами данных MySQL из web-приложений.

Основные задачи: изучить механизмы доступа к данным, находящимся в базе MySQL, установить соединение, прочитать данные.

16. Базы данных MySQL. Доступ к базе данных из web. Фильтрация входных данных (42 часа).

Цель работы: сформировать представление о механизме фильтрации.

Основные задачи: изучить механизмы фильтрации данных, сформировать HTML-документ с поддержкой CSS и скриптами PHP, демонстрирующими подключение к базе, чтение и фильтрацию данных.

17. Базы данных MySQL. Доступ к базе данных из web. Выполнение запроса к базе данных. Получение результатов запроса (2 часа).

Цель работы: сформировать представление о запросах к базе данных из web-приложения.

Основные задачи: сформировать HTML-документ с поддержкой CSS и скриптами PHP, демонстрирующими подключение к базе, чтение данных, выполнение запросов на выборку, добавление, обновление, удаление.

Самостоятельная работа

Самостоятельная работа студентов направлена на углубление и закрепление знаний, а также развитие практических умений и заключается в:

- работе с лекционным материалом, поиске и анализе литературы и электронных источников информации;
- выполнении домашних заданий (домашние задания представляют из себя перечень задач, с которыми студенты не справились в ходе выполнения лабораторных работ), заданий для самостоятельного выполнения к каждой лабораторной работе, подготовке ответов на контрольные вопросы к лабораторным работам;
- изучении теоретического материала к лабораторным занятиям.

Самостоятельная работа студента по настоящему курсу является гармоничным продолжением выполнения заданий, обозначенных в рамках лабораторных работ, а также работы с лекционным материалом по его расширению при поиске ответов на вопросы для самостоятельного изучения.

Проверка качества самостоятельной работы студентов проводится во время защиты лабораторных работ. Студент должен ориентироваться в теоретической базе, необходимой для выполнения текущей работы, выполнить все задания из лабораторной и самостоятельной частей, уметь отвечать на контрольные вопросы по направлению данной работы.

Вопросы для самостоятельного изучения

1. История возникновения Интернет, WorldWideWeb (WWW), и "стандартов Web".
2. стек протоколов TCP/IP.

3. Принципы создания Web-страниц.
4. Элементы Web-страниц.
5. Структура HTML-документа и элементы разметки заголовка документа
6. Взаимосвязи документов: элемент LINK.
7. Принципы применения таблиц в HTML – разметке
8. Основные понятия CSS, их назначение, определение и использование при форматировании HTML- документа.
9. Применение Web-страниц с таблицами.
10. Структура HTML-кода.
11. Применение списков. Маркированные списки. Нумерованные списки. Списки определений.
12. Использование гиперссылок. Внешние гиперссылки. Внутренние гиперссылки. Гиперссылки на адрес электронной почты.
13. Структура HTML-документа, содержащего фреймы. Описание фреймовой структуры.
14. Формы. Создание формы для регистрации сайта. Структура документа с формами.
15. Основные понятия JavaScript.
16. Операторы JavaScript.
17. Расположение функций внутри HTML-документа.
18. Встроенные классы JavaScript.
19. Основные понятия событий JavaScript.
20. Основные понятия PHP.
21. Методы встраивания PHP-кода.
22. Операторы PHP.
23. Константы в PHP.
24. Функции для работы со строками.
25. Выбор доменного имени.
26. Архитектуры распределённой обработки информации.
27. Безопасность в Интернете.
28. Скрытие IP-адреса, веб – анонимайзеры.
29. Анонимная отправка электронной почты».
30. Схема сайта.
31. Именованые страниц.
32. Монохроматические цветовые схемы.
33. Дополнительные цветовые схемы.
34. Триадиические цветовые схемы.
35. Определение гарнитуры шрифта для заголовков, подзаголовков и основного текста. Некоторые секреты выравнивания.
36. Базовые типы данных HTML.
37. Ссылки – мнемоники в HTML.
38. Формальное определение HTML, и проверка.
39. Формат и назначение элементов разметки заголовка.
40. Форматирование текста.
41. Организованная в таблицы графика.
42. Интерактивное меню навигации средствами CSS.

43. Схемы css-позиционирования. Способы позиционирования элементов.
44. Абсолютное и относительное позиционирование.
45. Способы отправки данных на сервер и их обработке с помощью PHP.
46. Основы клиент- серверных технологий. HTML-формы и отправка данных с ее помощью.
47. Краткая характеристика методов Post и Get.
48. Взаимодействие PHP и MySQL.
49. Способы взаимодействия PHP и СУБД MySQL.

Задачи для самостоятельного выполнения

1. Написать HTML-документ, содержащий абзацы. Каждый новый абзац должен отделяться от предыдущего текста пустой строкой.
2. Написать HTML-документ, содержащий перевод строки (разрыв строки), используя элемент
.
3. Написать HTML-документ с оформлением текста с различными способами выравнивания.
4. Написать HTML-документ содержащей заголовки всех шести уровней.
5. Написать HTML-документ, содержащий линии с различными значениями атрибутов.
6. Написать HTML-документ, в котором текст отображается шрифтами различных относительных размеров.
7. Написать HTML-документ, содержащий текст, отображаемый различными гарнитурами шрифтов.
8. Написать HTML-документ отображающий текст, выделенного элементами TT, BIG, SMALL, U, STRIKE.
9. Написать HTML-документ с внутренними ссылками.
10. Написать HTML-документ с выделением ячеек таблицы.
11. Написать HTML-документ содержащий панель навигации.
12. Написать HTML-документ описывающую фреймовую структуру.
13. Написать HTML-документ с применением плавающих фреймов.
14. Написать HTML-документ с применением шести фреймов.
15. Написать HTML-документ с использованием формы.
16. Написать HTML-документ формы регистрации пользователя.
17. Написать HTML-документ позиционирования блоков.
18. Написать HTML-документ содержащий список определений.
19. Написать HTML-документ содержащий блочные и строчные элементы с различными значениями параметров форматирования.
20. Написать HTML-документ содержащий «объемный текст».
21. Написать HTML-документ использования стилевых свойств при оформлении маркированного списка.
22. Написать HTML-документ с применением фиксированного позиционирования.
23. Написать HTML-документ содержащий три различных списка.
24. Написать HTML-документ содержащий группу флажков.
25. Написать HTML-документ содержащий пример формы книжного магазина (заполнение формы – фамилия, имя, образование посетителя).

6. Критерии оценивания результатов освоения дисциплины (модуля)

6.1. Оценочные средства и критерии оценивания для текущей аттестации

Задания для самостоятельного выполнения

Задания для самостоятельного выполнения разбиты в соответствии с тематическим планированием курса и являются гармоничным дополнением к лабораторным работам (см. пункт «Виды учебной деятельности. Самостоятельная работа»).

Критерии оценивания заданий для самостоятельного выполнения.

Уровень выполнения	Оценка
Задача решена в полном объеме, алгоритмические и вычислительные ошибки отсутствуют, проведен анализ полученного решения.	5 (отлично)
Задача решена в полном объеме с незначительными техническими ошибками или отсутствует анализ результатов решения.	4 (хорошо)
Задача решена не полностью или в решении присутствуют ошибки алгоритмического характера, незначительно влияющие на ход решения.	3 (удовлетворительно)
Задача не решена или в решении присутствует значительное количество ошибок алгоритмического характера, существенно влияющих на ход решения.	2 (неудовлетворительно)

Вопросы для самостоятельного изучения

Вопросы для самостоятельного изучения указаны в пункте «Виды учебной деятельности. Самостоятельная работа».

Критерии оценивания ответов на вопросы для самостоятельного изучения

Ответ по каждому вопросу оценивается по пятибалльной шкале в зависимости от содержательности ответа и логики изложения материала.

Уровень ответа	Оценка
Полно и аргументировано отвечает по содержанию темы; может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из лекции, но и самостоятельно составленные; излагает материал последовательно и корректно.	5 (отлично)
Дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.	4 (хорошо)
Излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки.	3 (удовлетворительно)
Не знает ответ на вопрос, допускает существенные ошибки в формулировке определений и алгоритмов, искажающие их смысл, беспорядочно и неуверенно излагает материал.	2 (неудовлетворительно)

6.2. Оценочные средства и критерии оценивания для промежуточной аттестации

Форма промежуточной аттестации – экзамен.

Для определения уровня сформированности компетенций применяется процедура независимой оценки.

Экзаменационный тест (типовой)

Какие парадигмы программирования поддерживает PHP?

- логическую
- **(Правильный ответ)** процедурную
- **(Правильный ответ)** объектно-ориентированную

Написать программу, которая выводит «жирными» буквами (тег) строку «Добро пожаловать!» с использованием языка PHP.

- **(Правильный ответ)**
<?php echo « Добро пожаловать! «»;?>
- <?phpecho «Добро пожаловать!»?>
- <?echo «<bold>Добро пожаловать!</bold>»?>

Известно, что настройки PHP можно сохранять/изменять не только в php.ini, но и в самих PHP скриптах, в файлах .htaccess и в файле настроек сервера httpd.conf. Где можно установить значение опции session.auto_start?

- **(Правильный ответ)** в php.ini или httpd.conf
- **(Правильный ответ)** в скрипте пользователя
- **(Правильный ответ)** в файле .htaccess

Какой оператор обозначает равенство значений в языке PHP?

- :=
- =
- =:
- **(Правильный ответ)** ==

Как можно задать массив в языке PHP?

- \$arr[«a»,»b»,»c»] = «q»;
- \$arr («0"=> «a»);
- **(Правильный ответ)** \$arr = array(«a»,»b»,»c»);
- **(Правильный ответ)** \$arr[0] = «a»;

Какие из утверждений относительно оператора require верны?

- require выполняет код указанного в нем файла только один раз
- **(Правильный ответ)** require используется для включения в программу содержимого другого файла
- при использовании внутри условных блоков require не нужно заключать в фигурные скобки

Каким будет результат выполнения следующей программы

```
<?phpfor ($j=1; ? { $i = round (9/$j); switch ($i) { case 5: echo «»+»»; break; case 9: echo «»-«»; break; case 3: echo «»!»»; break 2; default: echo $i; break; } $j++;}?>
```

- +-!
- ошибка синтаксиса
- бесконечный цикл
- **(Правильный ответ) -+!**

Дана команда: `if(!$var) echo «Hello»;` В каком случае на экран будет выведено слово «Hello»?

- если `$var == true`
- **(Правильный ответ) если `$var = 0`**
- **(Правильный ответ) если `$var` преобразуется к логическому `false`**

Какие из операторов `switch` записаны правильно с точки зрения синтаксиса?

- `switch ($par){«»1"»: echo «»1"»»;»»2"»: echo 2;}`
- **(Правильный ответ)**
`switch ($par){case «»1"»: echo «»1"»»;case 2: echo 2;default: echo 3;}`
- **(Правильный ответ)**
`switch ($par):case «»1"»: echo «»1"»»;break;case 2: echo 2;break;endswitch;`

Файл `vars.php`:

```
<?php$a = 1;?>
```

Файл `index.php`:

```
<?phpfor ($I=0; $I < 3; $I++){ include(«»vars.php»»); echo $a++;}?>
```

Что мы получим в результате обработки интерпретатором файла `index.php`?

- 12
- 1234
- 123
- **(Правильный ответ) 111**

В каком случае выполняется блок действий цикла

```
for (expr1; expr2; expr3) { // блок действий }
```

- **(Правильный ответ) если второе выражение (`expr2`) вычисляется как `true`**
- если третье выражение (`expr3`) вычисляется как `true`
- если первое выражение (`expr1`) вычисляется как `true`

Какое из утверждений относительно оператора `include` верно?

- файл, включаемый с помощью include, должен быть правильным HTML-файлом
- **(Правильный ответ)** include используется для включения в программу содержимого другого файла
- файл, включаемый с помощью include, может быть любым файлом

Как можно узнать виртуальный путь до выполняющегося в данный момент скрипта?

- **(Правильный ответ)** getenv('SCRIPT_NAME')
- **(Правильный ответ)** \$_SERVER['SCRIPT_NAME']
- **(Правильный ответ)** \$_SERVER['PHP_SELF']
- с помощью константы SCRIPT_NAME

Чем отличается клиент от сервера?

- **(Правильный ответ)** клиент посылает запросы, а сервер обрабатывает их
- **(Правильный ответ)** клиент отображает данные на экране компьютера пользователя, а сервер предоставляет данные
- **(Правильный ответ)** сервер выполняет запросы клиента, специализируясь на эффективном решении задач определенного класса
- сервер есть часть компьютерной архитектуры сервер-сервер, а клиент – архитектуры клиент-клиент
- сервер создает соединение, а клиент, пользуясь им, передает свой запрос

Как можно получить адрес страницы, с которой был послан запрос?

- **(Правильный ответ)** \$_SERVER['HTTP_REFERER']
- getenv('REMOTE_HOST')
- **(Правильный ответ)** getenv('HTTP_REFERER')
- с помощью константы HTTP_REFERER

В чем состоят функции сервера?

- **(Правильный ответ)** обрабатывать запросы клиента
- **(Правильный ответ)** запускать процессы, запрошенные клиентом, и возвращать клиенту результаты
- отображать данные на экране компьютера пользователя
- инициировать соединение с клиентом

Дана следующая html форма:

```
<form name="my_form" method="get">Field 1 <input type="text" name="f1"><br>Select field <select name="f2"> <option value="1">first <option value="2">second</select><br> <input type="submit"></form>
```

Как можно получить переданные клиентом значения из этой формы?

- \$f1 и \$f2, если register_globals=off
- \$_POST['f1'] и \$_POST['f2']

- **(Правильный ответ)** \$_GET[«f1»] и \$_GET[«f2»]
- \$POST[«f1»] и \$POST[«f2»]
- **(Правильный ответ)** \$_REQUEST[«f1»] и \$_REQUEST[«f2»]

Создать класс А, расширяющий класс В с помощью метода add() и переменной \$a.

- class B { var \$b; function B() {} } class B extends A { var \$a; function add(\$b) { \$this->a += \$b; } }
- **(Правильный ответ)** class B { var \$b; function B() {} } class A extends B { var \$a; function add(\$b) { \$this->a += \$b; } }
- class A extends B { var \$a; function add(\$b) { \$this->a += \$b; } }

Как можно получить и вывести на экран список всех методов класса, представителем которого является объект?

- get_methods(get_class(объект));
- **(Правильный ответ)** get_class_methods(get_class(объект));
- get_class_methods(get_class -> объект);

Класс MyClass задан следующим образом:

```
<?php class MyClass { var $a; function MyClass() { $this->a = «hello»; } function MyFunc($b) { return $b . «» . $this->a; } } ?>
```

Создать представителя класса MyClass. Получить и вывести значения всех свойств этого класса. Вызвать методы данного класса.

- **(Правильный ответ)** \$myObj = new MyClass(); echo \$myObj->a; \$str = \$myObj->MyFunc(«people»); Конструктор MyClass() вызывается автоматически при создании представителя класса.
- new MyClass(); echo \$MyClass->a; \$str = \$MyClass->MyFunc(«people»); Конструктор MyClass() вызывается автоматически при создании представителя класса.
- \$myObj = new MyClass(«test»); \$b = «a»; echo \$myObj->\$b; MyFunc(\$b);

Имеется объект (\$obj) какого-то класса. Требуется получить список всех свойств класса, которому принадлежит объект.

- \$get_class_vars(get_class->\$obj);
- get_class_var(get_class(\$obj));
- **(Правильный ответ)** \$vars = get_class_vars(get_class(\$obj));

Как можно программно узнать имя класса, представителем которого является объект?

- \$(get_class->объект);
- **(Правильный ответ)** get_class(объект);

- class(объект);

Имеется следующая форма:

```
<form action=task3.php>Запись номер 1 <input type=checkbox name=id[] value=10>
<br>Запись номер 2 <input type=checkbox name=id[] value=20> <br>Запись номер 3 <input
type=checkbox name=id[] value=30> <br><input type=submit value=»»Отправить»»></form>
```

Изменить значения переданных элементов, увеличив их в 10 раз. Использовать функцию array_walk(). Вывести значения элементов до и после применения array_walk

- **(Правильный ответ)**

```
<?print_r($_GET);array_walk($_GET,»»test»»);function test(&$val,$key){foreach ($val as
$k=> $v)$val[$k] = $v*10;}print_r($_GET);?>
```
- ```
<?print_r($_GET);array_walk($_GET,»»test»»);function test(&$val,$key){ $val =
$val*10;}print_r($_GET)?>
```
- ```
<?print_r($_POST);array_walk($_POST,»»test»»);function test(&$val,$key){foreach
($val as $k=> $v)$val[$k] = $v*10;}print_r($_POST);?>
```

Дана строка «<h2>Наука – двигатель прогресса!</h2>». Как можно вывести ее в браузер именно в таком виде, без форматирования согласно стандарту HTML?

- **(Правильный ответ)**

```
echo htmlspecialchars(«»<h2>Наука — <font color=red>двигатель</font>
прогресса!</h2>»»);
```
- ```
echo htmlentities(«»<h2>Наука – двигатель
прогресса!</h2>»»);
```
- **(Правильный ответ)**  

```
echo htmlspecialchars(«»<h2>Наука — двигатель
прогресса!</h2>»», ENT_COMPAT);
```

Дана строка текста: «PHP – очень простой язык. Я знаю PHP!!!». Заменить в ней все вхождения слова «PHP» на слово «Perl».

- ```
<?php$str = «»PHP – очень простой язык. Я знаю PHP!!!»»;echo str_replace($str,
«»Perl»»);?>
```
- **(Правильный ответ)**

```
<?php$str = «»PHP – очень простой язык. Я знаю PHP!!!»»;echo str_replace(«»PHP»»,
«»Perl»», $str);?>
```
- ```
<?php$str = «»PHP – очень простой язык. Я знаю PHP!!!»»;echo
substr_replace(«»PHP»», «»Perl»», $str);?>
```

Что делает функция move\_uploaded\_file (временное имя файла, место назначения)?

- удаляет загруженный файл
- **(Правильный ответ)** проверяет, загружен ли файл, и если да, то перемещает его из временной директории в указанную директорию для хранения
- перемещает файл из временной директории в указанную директорию для хранения

Имеется следующая html-форма

```
<form enctype="multipart/form-data" action="parse.php" method="post"><input type="hidden" name="MAX_FILE_SIZE" value="30000" />Загрузить файл: <input type="file" name="my" />
<input type="submit" value="Отправить файл" /></form>
```

Как узнать размер файла, который клиент хочет загрузить на сервер с помощью этой формы.

- **(Правильный ответ)** filesize(\$\_FILES['my']['tmp\_name'])
- filesize(\$\_FILES['my']['name'])
- \$\_FILES['size']
- filesize(\$\_FILES['name']['my'])

С помощью какой функции можно проверить, существует ли файл?

- **(Правильный ответ)** file\_exists()
- is\_writable()
- is\_readable()

Какие из перечисленных функций считывают данные из файла, ссылка на который установлена функцией fopen?

- **(Правильный ответ)** fgetc()
- **(Правильный ответ)** fgets()
- fread()
- **(Правильный ответ)** fread()

Имеется следующая html-форма

```
<form enctype="multipart/form-data" action="parse.php" method="post"><input type="hidden" name="MAX_FILE_SIZE" value="30000" />Загрузить файл: <input type="file" name="my" />
<input type="submit" value="Отправить файл" /></form>
```

Как узнать, что файл, загружаемый на сервер с помощью этой формы, был загружен успешно?

- if (isset(\$\_FILES['my'])) echo «файл загружен успешно»;
- **(Правильный ответ)** if (\$\_FILES['my']['error'] == 0) echo «файл загружен успешно»;
- if (\$\_FILES['my']['error'] == 1) echo «файл загружен успешно»;

Каков синтаксис оператора удаления таблицы?

- REMOVE TABLE [IF EXISTS] имя\_таблицы [, имя\_таблицы,...] [RESTRICT | CASCADE]



- **(Правильный ответ)** DROP TABLE [IF EXISTS] имя\_таблицы [, имя\_таблицы,...] [RESTRICT | CASCADE]
- DELETE TABLE [IF EXISTS] имя\_таблицы [, имя\_таблицы,...] [RESTRICT | CASCADE]

Составить запрос к таблице Articles на получение всех значений таких ее полей, как название статьи (title), автор (author) и краткое содержание (abstract). Упорядочить результат запроса по имени автора.

- SELECT \* FROM Articles WHERE author=»»\$author»», title=»»\$title»», abstract=»»\$abstract»» ORDER BY author;
- SELECT (title, author, abstract) FROM Articles WHERE ORDER BY author;
- **(Правильный ответ)**  
SELECT title, author, abstract FROM Articles ORDER BY author;

Оператор update предназначен для обновления значений существующих столбцов таблицы в соответствии с введенными значениями. Каков синтаксис этого оператора?

- UPDATE имя\_таблицы SET имя\_столбца1=выражение1 [, имя\_столбца2=выражение2, ...]
- UPDATE имя\_таблицы SET имя\_столбца1=выражение1 [, имя\_столбца2=выражение2, ...]
- **(Правильный ответ)**  
UPDATE [LOW\_PRIORITY] [IGNORE] имя\_таблицы SET имя\_столбца1=выражение1 [, имя\_столбца2=выражение2, ...][WHERE where\_definition] [LIMIT число]

Составить запрос к таблице описаний статей (Articles): изменить название статьи (title), автор (author) которой «Петров» на название «Второе название».

- **(Правильный ответ)**  
UPDATE Articles SET title=»»Второе название»» WHERE author=»»Петров»»;
- UPDATE Articles SET title=»»Второе название»»;
- UPDATE Articles SET title=»»Второе название»» WHERE title=»» Первое название»»;

Составить запрос к таблице Articles на добавление описания статьи с названием (title) «Новая статья» и автором (author) «Сидоров С. С.».

- **(Правильный ответ)**  
INSERT INTO Articles SET title='Новая статья', author='Сидоров С. С.';
- INSERT INTO Articles title='Новая статья', author='Сидоров С. С.';
- **(Правильный ответ)**  
INSERT INTO Articles (title, author) VALUES('Новая статья', 'Сидоров С. С.);

Получить имена всех полей таблицы persons базы данных book

- ```
<?php$fld = mysql_list_fields(«»book»», «»persons»»);$n = mysql_num_fields($fld);for($i=0;$i<$n; $i++){ $name_f = mysql_field_name ($i); echo «»<br>Имя поля: «». $name_f;}?>
```

- **(Правильный ответ)**

```
<?php$conn = mysql_connect(«localhost», «nina», «123»);$fld =
mysql_list_fields(«book», «persons», $conn);$n =
mysql_num_fields($fld);for($i=0;$i<$n; $i++){ $name_f = mysql_field_name ($fld, $i);
echo «<br>Имя поля: «. $name_f;}>
```
- ```
<?php$conn = mysql_connect(«localhost», «nina», «123»);$fld =
mysql_list_fields(«book», «persons», $conn);for($i=0;$i<count($fld); $i++){ echo
«
Имя поля: «. $fld[$i];}>
```

Функция `mysql_connect`, устанавливающая соединение с базой данных MySQL, имеет следующий синтаксис:

```
mysql_connect (server, username, password, new_link, client_flags)
```

Какие значения будут установлены для параметров `server`, `username`, `password`, если они не были заданы при вызове функции `mysql_connect`?

- **(Правильный ответ)**  
`server` = 'localhost:3306'  
`username` = имя пользователя владельца процесса  
`serverpassword` = пустой пароль
- эти параметры обязательны для функции `mysql_connect`
- `server` = 'localhost:8080'  
`username` = имя пользователя владельца процесса  
`serverpassword` = пароль пользователя владельца процесса сервера

Каковы синтаксис и семантика функции `explode`?

- массив `explode` (строка `string`). Эта функция разбивает строку `string` на части с помощью разделителя » » и возвращает массив полученных строк
- эта функция без параметров. Ее семантика неизвестна
- **(Правильный ответ)** массив `explode`(строка `separator`, строка `string` [, `int limit`]). Эта функция разбивает строку `string` на части с помощью разделителя `separator` и возвращает массив полученных строк

Параметр `session.cookie_lifetime` задает длительность жизни cookies в секундах. Какое значение имеет этот параметр по умолчанию?

- **(Правильный ответ)** по умолчанию это «0», т.е. данные в cookies считаются правильными до закрытия окна браузера
- по умолчанию это «10», т.е. данные в cookies считаются правильными ровно 10 секунд
- по умолчанию это «60»

Уничтожить текущую сессию целиком можно командой `session_destroy()`; К чему приведет уничтожение сессии?

- после уничтожения сессии очищается массив `$_SESSION` и больше ничего не происходит

- **(Правильный ответ)** после уничтожения сессии уничтожается ее идентификатор, мы больше не можем ни регистрировать переменные, ни вообще производить какие-либо действия с сессией
- после уничтожения сессии массив `$_SESSION` заполняется нулями

С помощью какой функции можно получить идентификатор сессии?

- `id_session`
- `session_identificator`
- **(Правильный ответ)** `session_id`

В одном из скриптов программы имеется переменная `$user_name = «Иван Петров»`. Как сделать так, чтобы эта переменная была доступна во всех скриптах программы.

- **(Правильный ответ)** передавать эту переменную при переходе от скрипта к скрипту в качестве скрытого элемента HTML-формы
- **(Правильный ответ)** в этом скрипте нужно создать сессию (или восстановить текущую) с помощью команды `session_start()`. Потом нужно зарегистрировать переменную в качестве переменной сессии таким образом: `$_SESSION[«user_name»] = «Иван Петров»;`

Все остальные скрипты программы начинать с команды `session_start()`;

- в этом скрипте нужно создать сессию (или восстановить текущую) с помощью команды `session_start()`. Потом нужно зарегистрировать переменную в качестве переменной сессии таким образом: `$_SESSION[«user_name»] = «Иван Петров»;`
- нужно зарегистрировать переменную в качестве переменной сессии: `$_SESSION[«user_name»] = «Иван Петров»;`

Какова структура регулярного выражения?

- **(Правильный ответ)** общая структура регулярного выражения: шаблон, выделенный с помощью специального символа разделителя, модификатор, влияющий на способ обработки регулярного выражения
- общая структура регулярного выражения: шаблон, заключенный в круглые скобки, и функция для его обработки
- общая структура регулярного выражения: шаблон, выделенный с помощью специального символа разделителя

Назначение метасимвола «-«?

- вычисляет символьный класс
- отрицание класса, но только если это первый символ
- **(Правильный ответ)** задает диапазон символов

Выделить из URL адреса ресурса имя хоста

- **(Правильный ответ)**  
<?php preg\_match («»/^(\http:\V\)?([\^V]+)/i»», «»http://www.php.net/test/index.html»», \$matches);echo «»Host:»», \$matches[2];?>
- <?php preg\_match («»/^(\http://)(\w+[\^V]+)/i»», «»http://www.php.net/test/index.html»», \$matches);echo «»Host:»», \$matches[2];?>
- <?php preg\_match («»/^(\http:\V\)?([\^V]+)/i»», «»http://www.php.net/test/index.html»», \$matches);echo «»Host:»», \$matches[0];?>

Что делает следующая функция?

```
function Test($str){$pattern = «»/\d{3}-\d{2}-\d{2}/m»»; $num_match = preg_match_all($pattern, $str, $result);return $num_match;}
```

- возвращает 0 или 1 в зависимости от того, встречена ли в строке семизначная комбинация цифр, записанных в виде: три цифры, тире, две цифры, тире, две цифры
- **(Правильный ответ)** возвращает число встреченных в строке семизначных комбинаций цифр, записанных в виде: три цифры, тире, две цифры, тире, две цифры
- возвращает число встреченных в строке семизначных комбинаций цифр, состоящих из цифр 3 и 2

Функция domxml\_new\_doc в качестве результата возвращает пустой XML-документ. Что передается этой функции в качестве параметра?

- в качестве параметра передается строка, в которой содержатся первоначальные данные XML-документа
- в качестве параметра передается строка, содержащая путь, где будет храниться XML-документ
- **(Правильный ответ)** в качестве параметра передается версия создаваемого XML-документа
- у этой функции параметров нет

Что возвращает метод document\_element класса XmlDocument?

- содержимое элемента
- **(Правильный ответ)** корневой элемент
- значение атрибута

Какая комбинация технологий позволяет получить HTML-документы?

- **(Правильный ответ)** XML + XSLT
- XML + DTD
- XSLT + DTD

Что представляет собой шаблон Smarty?

- это набор переменных Smarty и html-тегов
- это набор html-тегов

- **(Правильный ответ)** это набор переменных, циклов, условных операторов, операторов вставки файлов и т.д

Для чего используется метод parse() при работе с шаблонами FastTemplate?

- обрабатывает шаблон и выводит его содержимое на экран
- **(Правильный ответ)** этот метод инициализирует обработку шаблона и сохраняет обработанный шаблон в переменную
- этот метод присваивает переменной указанное значение

Как задается переменная в шаблоне Smarty?

- {имя\_переменной}
- **(Правильный ответ)** {\$имя\_переменной}
- **(Правильный ответ)** {#bodyBgColor#}

Дан массив \$b = array(«23aaa»,»4",»qww»,»с», 3). Найти в массиве число 3, не перебирая все элементы массива. Если элемент найден, вывести значение его ключа.

- `<?php$b = array(«23aaa»,»4",»qww»,»с», 3);$index = array_search($b,»3");if ($index === false) echo»Нет такого числа в массиве»;else { echo»Число найдено с ключом «»; echo $index;}?>`
- `<?php$b = array(«23aaa»,»4",»qww»,»с», 3);if (!array_search(«3", $b)) echo»Нет такого числа в массиве»;else { echo»Число найдено с ключом «»; echo array_search(«3", $b);}?>`
- **(Правильный ответ)**  
`<?php$b = array(«23aaa»,»4",»qww»,»с», 3);$index = array_search(3,$b,true);if ($index === false) echo»Нет такого числа в массиве»;else { echo»Число найдено с ключом «»; echo $index;}?>`

Что делает приведенная ниже программа.

```
<?php $f = fopen(«file.html», «r»);echo fread($f, 1024);fclose($f);?>
```

- открывает файл file.html только для записи. Дописывает в конец файла строку «1024». Если файла с именем file.html не существует в директории скрипта, то он будет создан и в него запишется слово «1024». Закрывает соединение с файлом
- **(Правильный ответ)** открывает файл file.html только для чтения. Считывает 1024 байт текста из файла и выводит его на экран. Закрывает соединение с файлом
- открывает файл file.html для записи и чтения. Считывает 1024 символа из файла и выводит его на экран. Закрывает соединение с файлом

Функция domxml\_open\_file в качестве результата возвращает объект класса DOMDocument. Что передается этой функции в качестве параметра?

- в качестве параметра передается строка, содержащая XML-документ
- **(Правильный ответ)** в качестве параметра передается строка, содержащая путь к XML-документу

- в качестве параметра передается объект класса DOMNode

Каким будет результат работы следующей программы

```
<?phpfunction Test($a=»q», $b){ echo $a; return $a . $b;}Test(1);?>
```

- q1
- qq1
- **(Правильный ответ)** предупреждение, что не хватает второго аргумента для функции, и значение 1

Данные некоторой формы отправлены на сервер методом POST. Проверить, была ли передана в качестве значения какого-либо элемента формы строка «hello»

- <?phpif (array(\$\_POST, «hello»), true) echo «Element found»;?>
- <?phpif (array\_search(«hello», \$\_GET)) echo «Element found»;?>
- **(Правильный ответ)**  
<?phpif (array\_keys(\$\_POST, «hello»)) echo «Element found»;?>

С какими параметрами может вызываться функция explode()?

- **(Правильный ответ)** максимальное количество возвращаемых строк
- максимальная длина возвращаемых строк
- **(Правильный ответ)** разделитель в виде строки
- **(Правильный ответ)** строка для разделения

Какие из циклов while записаны правильно с точки зрения синтаксиса?

- **(Правильный ответ)**  
while (\$a < \$b):echo \$a;\$a++;endwhile;
- **(Правильный ответ)**  
while (\$a < \$b){echo \$a;\$a++;}
- while (\$a < \$b){echo \$a;\$a++;endwhile;}

Что такое пользовательская директория сервера и где она находится по умолчанию (для windows систем)?

- это директория, выделенная пользователю для личных нужд, файлы которой обрабатываются сервером. Находится в «C:/Program Files/Apache Group/Apache/htdocs/users/»
- **(Правильный ответ)** это директория, выделенная пользователю для личных нужд, файлы которой обрабатываются сервером, если он получает запрос типа ~user. Находится в «C:/Program Files/Apache Group/Apache/users/»
- это директория, выделенная пользователю для личных нужд, файлы которой никогда не обрабатываются сервером. Находится в «C:/Program Files/Apache Group/Apache/users/»

- это директория, выделенная пользователю для личных нужд, файлы которой обрабатываются сервером, если он получает запрос типа ~user. Находится в «C:/Program Files/Apache Group/Apache/htdocs/users/»

### Критерии оценивания результатов тестирования

Ответ на каждый вопрос экзаменационного теста оценивается от 0 до 1 балла.

#### *Критерии оценивания уровня овладения студентами компетенциями на экзамене*

Оценка по дисциплине складывается из трех составляющих:

- успешность текущей работы в течение семестра ( $A$ );
- бонусных баллов ( $B$ );
- успешность прохождения экзаменационного теста ( $C$ ).

Пусть  $A_{max}$  – максимальный суммарный балл за текущую работу,  $C_{max}$  – максимальный балл за прохождение экзаменационного теста. Тогда оценка по 100-балльной шкале определяется следующим образом:

$$R_{100} = \text{Min}(100, 50(A + B) / A_{max} + 50C / C_{max}).$$

Оценка на экзамене определяется по таблице:

Оценка по 5-балльной шкале	Оценка по 10-балльной шкале	Оценка по 100-балльной шкале
5 – «отлично»	10	95-100 баллов
	9	91-95 баллов
	8	86-90 баллов
4 – «хорошо»	7	79-85 баллов
	6	71-78 баллов
3 – «удовлетворительно»	5	63-70 баллов
	4	56-62 баллов
2 – «неудовлетворительно»	3	38-55 баллов
	2	19-37 баллов
	1	0-18 баллов

### Курсовой проект

«Разработка web-сайта»

**Постановка задачи.** Выполните задания в соответствии с методическими рекомендациями. В качестве результата приложите отчёт по курсовому проекту.

Требуется создать сайт из нескольких web-страниц. Минимальный набор языковых средств HTML + JS + PHP, который должен быть использован в документах:

- разделение экрана на фреймы;
- разные приемы форматирования текста;
- оформление списков;
- оформление таблиц;

- гиперссылки, нацеленные на свой и чужой фреймы;
- использование изображений как независимых объектов и в качестве фона;
- элементы Java Script;
- PHP.

Возможные темы сайтов: любимая вокальная группа, театральный коллектив, спортивная команда, автор любимых книг, Ваш город, вид спорта, отдыха, реклама фирмы и т. п. Тема согласуется с преподавателем.

Критерии оценивания заданий курсового проекта.

Критерии оценки	Баллы
Соответствие содержания теме	0-5
Выполнение целей и задач	0-5
Оригинальность	0-5
Обоснованность применяемых методов/инструментов	0-5
Глубина проработки материала	0-5
Последовательность изложения материала	0-5
Грамотность изложения и точность формулировок	0-5
Аккуратность ссылок, библиографического описания	0-5
Аккуратность оформления текста (в т.ч. графиков, рисунков, таблиц и т.д.)	0-5
Использован обязательный минимальный набор языковых средств HTML + JS + PHP	0-5

Максимальный балл – 50.

При использовании 10-балльной шкалы итоговые баллы за проект переводятся в оценку по формуле  $K = \text{баллы} / 5$ . Оценка округляется до целых.

#### Итоговая оценка

Оценка	
10	отлично
9	
8	
7	хорошо
6	
5	удовлетворительно
4	
3	неудовлетворительно
2	
1	

## 7. Перечень основной и дополнительной учебной литературы

### 7.1. Основная литература

1. *Полуэктова, Н. Р.* Разработка веб-приложений: учебное пособие для вузов / Н. Р. Полуэктова. — Москва: Издательство Юрайт, 2022. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496682>.
2. *Тузовский, А. Ф.* Проектирование и разработка веб-приложений: учебное пособие для вузов / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2022. — 218 с. — (Высшее образование). — ISBN 978-5-534-00515-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490128>.

### 7.2. 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, 2022. — 90 с. — (Высшее образование). — ISBN 978-5-9916-9975-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492224>.

### **7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. Свободно доступные курсы Интернет-университета информационных технологий (ИНТУИТ) <http://www.intuit.ru/>;
2. Официальное руководство по PHP. <https://www.php.net/manual/ru/>
3. Программирование на PHP и основы MySQL. <https://addphp.ru/>
4. PHP: Правильный Путь. <https://getjump.github.io/ru-php-the-right-way/>

## **8. Материально-техническое обеспечение**

**Учебная аудитория для проведения занятий лекционного типа.** Аудитория 124 уч.к. № 2.

Стандартная учебная мебель (40 учебных посадочных мест), стол и стул для преподавателя – по 1 шт., кафедра для лектора – 1 шт.

Компьютерные студенческие столы (17 шт.), компьютерный стол для преподавателя – 1 шт., мониторы Acer – 18 шт., системные блоки Kraftway – 18 шт., колонки Genius – 18 шт., мультимедиапроектор BenQ – 1 шт., интерактивная доска Interwrite – 1 шт. Обеспечен выход в Интернет.

Программное обеспечение: Microsoft Open License (Windows XP, 7, 8, 10, Server, Office 2003-2016), лицензия 66975477 от 03.06.2016 (бессрочно).

**Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации – компьютерный класс.** Аудитория 124 уч.к. №2.

Стандартная учебная мебель (40 учебных посадочных мест), стол и стул для преподавателя – по 1 шт., кафедра для лектора – 1 шт.

Компьютерные студенческие столы (17 шт.), компьютерный стол для преподавателя – 1 шт., мониторы Acer – 18 шт., системные блоки Kraftway – 16 шт., колонки Genius – 16 шт., мультимедиапроектор BenQ – 1 шт., интерактивная доска Interwrite – 1 шт. Обеспечен выход в Интернет.

Программное обеспечение: Microsoft Open License (Windows XP, 7, 8, 10, Server, Office 2003-2016), лицензия 66975477 от 03.06.2016 (бессрочно).

**Помещение для самостоятельной работы – компьютерный класс с доступом к сети «Интернет» и ЭИОС СмолГУ.** Аудитория 124 уч.к. №2.

Стандартная учебная мебель (40 учебных посадочных мест), стол и стул для преподавателя – по 1 шт., кафедра для лектора – 1 шт.

Компьютерные студенческие столы (17 шт.), компьютерный стол для преподавателя – 1 шт., мониторы Acer – 18 шт., системные блоки Kraftway – 18 шт., колонки Genius – 18 шт., мультимедиапроектор BenQ – 1 шт., интерактивная доска Interwrite – 1 шт. Обеспечен выход в Интернет.

Программное обеспечение: Microsoft Open License (Windows XP, 7, 8, 10, Server, Office 2003-2016), лицензия 66975477 от 03.06.2016 (бессрочно).

## **9. Программное обеспечение**

Microsoft Open License (Windows XP, 7, 8, 10, Server, Office 2003-2016), Лицензия 66920993 от 24.05.2016

Microsoft Open License (Windows XP, 7, 8, 10, Server, Office 2003-2016), Лицензия 66975477 от 03.06.2016

Dr. Web Server/Desktop Security Suite (Антивирус) Лицензия EE4E-QN5S-6FG2-N76B (Ежегодное обновление)

Kaspersky Endpoint Security для бизнеса – Стандартный, Лицензия 1FB6151216081242, ежегодное обновление.

- MS Visual Studio 2019;

- MySQL 8;

- PHP 8.

ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 03B6A3C600B7ADA9B742A1E041DE7D81B0  
Владелец: Артеменков Михаил Николаевич  
Действителен: с 04.10.2021 до 07.10.2022