

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Щёкина Вера Викторовна
Должность: Ректор
Дата подписания: 15.09.2024 19:49
Уникальный программный идентификатор:
a2232a55157e576551a8993b1190891af58987d7b47017b0b171a514e17789



**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное
учреждение высшего образования**

«Благовещенский государственный педагогический университет»

**ОСНОВНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
Рабочая программа дисциплины**

УТВЕРЖДАЮ

**декан факультета физико-
математического образования и тех-
нологии ФГБОУ ВО БГПУ**

Н.В. Слесаренко

«03» сентября 2024 г.

**Рабочая программа дисциплины
РАЗРАБОТКА КОРПОРАТИВНЫХ САЙТОВ**

Направление подготовки

**02.03.03 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И
АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

Профиль

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

**Уровень высшего образования
БАКАЛАВРИАТ**

**Принята
на заседании кафедры информатики
и методики преподавания информатики
(протокол № 8 от «25» мая 2024 г.)**

Благовещенск 2024

СОДЕРЖАНИЕ

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ	4
3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)	5
4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ	6
5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ	9
6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА.....	35
7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ	37
В ПРОЦЕССЕ ОБУЧЕНИЯ	38
8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	39
9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ	39
10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА	40
11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ	41

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Цель дисциплины: формирование у студентов компетентности в области разработки корпоративных сайтов, современных методов программирования web-приложений на основе представлений о современных языках программирования высокого уровня, приобретение знаний и навыков при разработке современных web-сайтов.

1.2 Место дисциплины в структуре ООП: Дисциплина «Разработка корпоративных сайтов» относится к дисциплинам обязательной части блока Б1 (Б1.О.34). Данной дисциплине предшествует изучение дисциплины «Основы сайтостроения».

1.3 Дисциплина направлена на формирование следующих компетенций: ОПК-7, ПК-5.

- **ОПК-3.** Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения, **индикаторами** достижения которой является:

- ОПК-3.1 – **знает** основные положения и концепции в прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов;

- ОПК-3.1 – **умеет** использовать их в профессиональной деятельности;

- ОПК-3.1 – **имеет практические навыки** разработки программного обеспечения.

- **ПК-5.** Способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов, **индикаторами** достижения которой является:

- ПК -5.1 – **знает** современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования.

- ПК -5.1 – **умеет** использовать подобные инструментальные средства в практической деятельности;

- ПК -5.1 – **имеет практический опыт** применения подобных инструментальных средств.

1.4 Перечень планируемых результатов обучения. В результате изучения дисциплины студент должен **знать:**

- основные понятия компьютерных сетей и систем телекоммуникации;
- функционирование глобальной сети Интернет;
- процессы и архитектуру технологии «клиент-сервер»;
- основы web-дизайна;
- технологию создания гипертекстовых документов;
- приемы создания и оптимизации графических элементов сайта;
- клиентские технологии web-программирования;
- технологии создания web-приложений;
- средства управления HTML -документами;
- технологии PHP, JScript, MySQL.

уметь:

- настраивать программное обеспечение для работы в сети Интернет;
- применять языки гипертекстовой разметки и CSS к созданию web-документов;
- создавать макет сайта; распределять информацию по разделам сайта;
- разрабатывать навигацию;
- использовать шаблоны;
- макетировать сайт с учетом эргономики (web-usability);
- разрабатывать динамические элементы;
- оценивать и тестировать сайт;
- создавать интерактивные web-приложения на основе технологии PHP, JScript.

владеть:

- технологией создания web-документов;
- технологией программирования PHP, JScript.

1.5 Общая трудоемкость дисциплины «Разработка корпоративных сайтов» составляет 3 зачетные единицы (далее – ЗЕ) (108 часов).

Программа предусматривает изучение материала на лекциях и практических занятиях. Предусмотрена самостоятельная работа студентов по темам и разделам. Проверка знаний осуществляется фронтально, индивидуально.

1.6 Объем дисциплины и виды учебной деятельности**Объем дисциплины и виды учебной деятельности (очная форма обучения)**

Вид учебной работы	Всего часов	Семестр 6
Общая трудоемкость	108	108
Аудиторные занятия	54	54
Лекции	26	26
Лабораторные работы	28	28
Самостоятельная работа	54	48
Вид итогового контроля		зачет

2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ**2.1 Очная форма обучения****Учебно-тематический план**

№	Наименование тем (разделов)	Всего часов	Аудиторные занятия		Самостоятельная работа
			Лекции	Лабораторные работы	
1.	Современные технологии разработки web-приложений.	4	2	0	2
2.	Протоколы работы сети интернет.	6	2	0	4
3.	Средства создания web-страниц.	14	4	4	6

4.	Основы языка программирования JavaScript.	12	2	4	6
5.	Основы технологии PHP.	16	4	4	8
6.	Создание Web-приложений средствами PHP.	20	4	8	8
7.	Работа с базами данных в web-приложениях.	16	4	6	6
8.	Основы технологии MySQL	20	4	8	8
Зачет					
ИТОГО		108	26	34	48

Интерактивное обучение по дисциплине

№	Наименование тем (разделов)	Вид занятия	Форма интерактивного занятия	Кол-во часов
1.	Современные технологии разработки web-приложений.	ЛК	Лекция-дискуссия	4
2.	Протоколы работы сети интернет.	ЛК	Лекция-дискуссия	2
3.	Средства создания web-страниц.	ЛК	Лекция-дискуссия	2
4.	Основы языка программирования JavaScript.	ЛБ	Работа в малых группах	2
5.	Основы технологии PHP.	ЛБ	Работа в малых группах	2
6.	Создание Web-приложений средствами PHP.	ЛБ	Работа в малых группах	2
7.	Работа с базами данных в web-приложениях.	ЛБ	Работа в малых группах	2
8.	Основы технологии MySQL	ЛБ	Работа в малых группах	2
ИТОГО				18

3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)

Тема 1. Современные технологии разработки web-приложений

Основы разработки web-сайтов. Стратегии и направления развития web-индустрии. Подходы и популярные концепции разработки сайтов. Обзор современных технологий, преимущества и недостатки. Логическая и физическая структура web-сайта. Основные черты профессионально выполненного web-сайта. Динамическая и статическая компоновки сайта. Web-фреймворки, CMS, языки применяемые в разработке корпоративных сайтов.

Тема 2. Протоколы работы сети интернет.

Примеры сетевых протоколов: TCP/IP, HTTP, FTP, POP, SMTP, DNS, RTSP. Структура и свойства URL. Определение интернет и WWW.

Тема 3. Средства создания web-страниц.

Назначение языка HTML. Структура документа. Основные элементы языка. Форматирование. Понятие о тегах (метках). Непарные метки. Комментарии. Создание заголовков разных уровней. Цвета и спецсимволы. Фон web-страницы. Оформление текста: абзац, разрыв строки, выравнивание текста в абзаце, задание шрифта, увеличение и уменьшение размера шрифта. Связывание отдельных web-страниц. Гиперссылки. Закладки. Таблицы. Списки. Формы. Основные элементы ввода данных и управления формой. Многострочные текстовые поля. Фреймы. Графические элементы в HTML-документе. Основы создания каскадных таблиц стилей. Практическое использование возможностей CSS при разработке web-сайтов. Особенности отображения текста на web-странице. CSS-свойства, используемые для оформления текста. Блочная модель. Управление типами элементов.

Тема 4. Основы языка программирования JavaScript.

Возможности JavaScript. Размещение сценариев. Структура сценариев на JavaScript. Типы данных. Переменные и литералы. Операторы. Конструкции языка JavaScript. Функции пользователя. Проверка условий. Циклы.

Тема 5. Основы технологии PHP.

Краткий обзор PHP, обзор спецификаций PHP. Контекст и внедрение зависимостей, валидация компонентов. Объектно-реляционное отображение; управление постоянными объектами. Понятие корпоративных PHP-компонентов. Обзор спецификации PHP.

Тема 6. Создание Web-приложений средствами PHP.

Развертывание средств разработки, процесс установки и настройки PHPMyAdmin, MySQL Server. Создание каркаса приложения, написание дескриптора web.xml. Уменьшение количества повторяемого кода в jsp страницах путем использования jsp сегментов. Web-консоль Web-сервера. Обработка данных, полученных от пользователя путем GET или POST запросов.

Тема 7. Работа с базами данных в web-приложениях.

Создание соединения с БД, настройка pool соединений с базой данных. Создание класса сущностей из таблиц базы данных.

Тема 8. Основы технологии MySQL.

Клиент-серверные системы. Системы обработки распределенных баз данных. Базы данных с использованием интернет-технологий. Объектно-ориентированные СУБД. Язык структурированных запросов SQL. Стандарты SQL. Функции и основные возможности. Основные конструкции языка.

4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

4.1 Общие методические рекомендации

В процессе изучения дисциплины студенты знакомятся с основами разработки web-приложений, процессами и протоколами работы компьютерной сети.

Дисциплина включает 34 часа лабораторных занятий. Перед выполнением работы следует изучить теоретический материал и ответить на контрольные вопросы. Отчет по лабораторной работе производится лично преподавателю с демонстрацией выполненного задания.

Список литературы – расширенный и позволяет использовать материалы не только для подготовки к аудиторным занятиям, но и для организации самостоятельной работы, а также для расширения собственных представлений по отдельным аспектам изучаемой дисциплины.

4.2 Методические рекомендации по подготовке к лекциям

Приступая к изучению курса «Разработка корпоративных сайтов», студент должен иметь общие представления об объекте, предмете, методах, и структуре данной дисциплины; о ее месте в системе общественных наук и ее соотношении с другими науками; о ее практическом применении в профессиональной деятельности; о характере научной и учебной литературы, которую предстоит изучить. Продуманная и целенаправленная подготовка к лекции закладывает необходимые основы для глубокого восприятия лекционного материала.

Самостоятельная работа начинается до прихода студента на лекцию. Многие студенты активно используют «систему опережающего чтения», то есть предварительно прочитывают лекционный материал, содержащийся в учебниках и учебных пособиях, закладывают базу для более, глубокого восприятия лекции.

Другой формой самостоятельной работы студента является посещение лекции, внимательное слушание выступления лектора и конспектирование основных теоретических положений лекции. Внимательное слушание лекции, уяснение основного её содержания, краткая, но разборчивая запись лекции - непереносимое условие успешной самостоятельной работы каждого студента. Поэтому студенты, присутствующие на лекциях, обязаны не только внимательно слушать преподавателя, но и конспектировать излагаемый им материал. При этом конспектирование материала представляет собой запись основных теоретических положений, рассуждений, излагаемых лектором. Нужно помнить, что конспектирование лекций дает студенту не только возможность пользоваться записями лекций при самостоятельной подготовке к практическим занятиям и зачету, но и глубже и основательней вникнуть в существо излагаемых в лекции вопросов, лучше усвоить и запомнить материал.

Важной формой самостоятельной работы студента является систематическая и планомерная подготовка к лекционному занятию. Наличие разборчивого, краткого конспекта лекции, позволит студенту задуматься над прочитанным лекционным материалом, изучить специальную литературу по теме лекции, приобщиться к работе с правовыми данными, интересоваться использованием юридических фактов для объяснения событий, явлений, процессов, уметь толковать их.

После лекции студент должен приступить к самостоятельной подготовке по соответствующей теме занятия по программе дисциплины. Он уясняет обязательную и дополнительную литературу, которую необходимо прочитать, изучить и законспектировать. Обычно разъяснение по этим вопросам студенты получают в конце предыдущего лекционного занятия, когда преподаватель объявляет очередную тему занятия и кратко рассказывает, как к нему готовиться.

Заключительным этапом в самостоятельной работе студента является повторение материала по конспекту, которое способствует ясному пониманию и глубокому овладению материалом. Но эта работа может быть проделана непосредственно накануне лекционного занятия.

4.3 Методические рекомендации по подготовке к практическим занятиям

Целью практических занятий является формирование у студентов системы знаний по организации учебного процесса в целом по выбранному направлению подготовки бакалавриата, выработка у них начальных навыков организации самостоятельной работы, написания рефератов. Целью практических занятий является закрепление теоретического материала лекций и выработка умения использования информационных и других ресурсов, предоставляемых университетом.

Для подготовки к занятиям необходимо познакомиться с ресурсами внешнего сайта ФГБОУ ВО БГПУ: <http://www.bgpu.ru/index.jsp>, с ресурсами внутреннего сайта ФГБОУ ВПО БГПУ: <http://iskander.bgpu.ru>, с системой электронного обучения вуза <http://moodle.bgpu.ru>.

Для проведения практических занятий используются компьютеры, оснащенные ОС Windows 2000 и выше. Возможно использование проектора или интерактивной доски.

Одной из форм интерактивных форм проведения занятия является дискуссия.

Важно предварительно определить правила ведения дискуссии, а также критерии оценки выступлений ее участников. Лучше всего это сделать предметом обсуждения в группе, а не предлагать преподавателем в готовом виде. Например, обсудить и принять следующие или похожие правила ведения дискуссии:

- не допускать выпадов против личности;
- не допускать излишнюю эмоциональность;
- высказываться четко, кратко и по теме обсуждения;
- выбрать ведущего дискуссии, ответственного за время и правила ее проведения;
- дать высказаться всем желающим, уважительно относиться к любой точке зрения;
- внимательно слушать друг друга, не говорить одновременно;
- постараться проанализировать разные точки зрения;
- не повторяться, продвигать дискуссию дальше через движение новых идей, сообщение новой информации.

Обсудить и принять в группе следующие критерии оценки участия в дискуссии:

- точность аргументов (причинно-следственные связи);
- четкость и понятность аргументации;
- точность контраргументов (причинно-следственные связи);
- четкость понятность контраргументов;
- логичность;
- удачная подача;
- умение выделить главное;
- отделение фактов от субъективного мнения;
- использование ярких поддерживающих фактов;
- видение сути проблемы;
- ориентация меняющейся ситуации.

4.5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

Фонд оценочных средств.

Вопросы к зачёту.

Список литературы и информационных ресурсов.

**Учебно-методическое обеспечение самостоятельной работы
студентов по дисциплине**

№	Наименование раздела (темы)	Формы/виды самостоятельной работы	Количество часов, в соответствии с учебно-тематическим планом
1.	Современные технологии разработки web-приложений.	Поиск в Internet	2
2.	Протоколы работы сети интернет.	Поиск в Internet	4
3.	Средства создания web-страниц.	Поиск в Internet, Работа по созданию приложения с использованием технологии HTML, CSS.	6
4.	Основы языка программирования JavaScript.	Поиск в Internet, Работа по созданию приложения с использованием технологии JavaScript.	6
5.	Основы технологии PHP.	Работа по созданию приложения с использованием технологии PHP.	8
6.	Создание Web-приложений средствами PHP.	Работа по созданию приложения с использованием технологии PHP.	8
7.	Работа с базами данных в web-приложениях.	Проектирование и создание базы данных, формирование к ней запросов.	6
8.	Основы технологии MySQL	Проектирование и создание базы данных, формирование к ней запросов.	8
	ИТОГО		48

5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

Тема 3. Средства создания web-страниц

Лабораторная работа № 1 (HTML / CSS).

"Разработка HTML-страниц (форматирование текста с помощью стилей, гиперссылки, списки, таблицы, изображения)."

Вопросы допуска:

- 1) Определение HTML, Общее представление.
- 2) Структура HTML-документа.
- 3) Версии

Задание:

Разработать не менее 4 HTML-страниц, объединенных общей темой. Тема содержания страниц выбирается самостоятельно. При затруднениях в выборе темы создайте

«сайт» о самом себе (биографические сведения, интересы, путешествия, домашние животные и т.д., и т. п.)

Страницы должны включать:

- Форматированный текст (абзацы, заголовки, различные шрифты, выделенные части текста);
- Гиперссылки (на другие страницы и внутри страниц). Все страницы должны быть размещены в разных каталогах, размещенных на разных уровнях относительно друг друга;
- Как минимум одно изображение.
- Как минимум один список.
- Вложенные таблицы (различный цвет фона и текста в ячейках, ячейки с различными правилами выравнивания текста внутри них).

Указание: Вся текстовая информация и гиперссылки должны форматироваться только с помощью стилей. Запрещено использовать следующие элементы:

B	Полужирный шрифт	STRIKE	Зачеркнутый текст
BIG	Увеличенный шрифт	SUB	Нижний индекс
BLINK	Мигающий текст	SUP	Верхний индекс
FONT	Тип, размер, цвет шрифта	TT	Буквы фиксированной ширины
I	Курсив	U	Подчеркнутый текст
S	Зачеркнутый текст	SMALL	Уменьшенный шрифт

Определение стилей должно быть вынесено в отдельный (.css) файл.

Назначение языка HTML. Структура документа. Основные элементы языка. Форматирование. Понятие о тегах (метках). Непарные метки. Комментарии. Создание заголовков разных уровней. Цвета и спецсимволы. Фон web-страницы. Оформление текста: абзац, разрыв строки, выравнивание текста в абзаце, задание шрифта, увеличение и уменьшение размера шрифта. Связывание отдельных web-страниц. Гиперссылки. Закладки. Таблицы. Списки. Формы. Основные элементы ввода данных и управления формой. Многострочные текстовые поля. Фреймы. Графические элементы в HTML-документе. Основы создания каскадных таблиц стилей. Практическое использование возможностей CSS при разработке web-сайтов. Особенности отображения текста на web-странице. CSS-свойства, используемые для оформления текста. Блочная модель. Управление типами элементов.

Лабораторная работа № 2 (HTML / CSS /Фреймы)

"Фреймы в HTML документах и карты сегментации изображений. Организация связей между фреймами и контекстно-зависимые пояснительные надписи"

Вопросы допуска:

- 1) Цель создания CSS.
- 2) Способы подключения CSS к документу.
- 3) Правила построения CSS.
- 4) Классы и идентификаторы элементов.
- 5) Наследование. Каскадирование. Приоритеты стилей CSS.

CSS-вёрстка

Задание:

Создать HTML - страницу, состоящую из 3-х фреймов:

- Заголовок с названием страницы (не прокручиваемый).

- Основная часть с изображением, снабженным картой сегментации.
- Описательная часть (прокручиваемый фрейм).

Основная часть должна быть построена на основе карты сегментации изображения. Для каждого сегмента (фрагмента), выделяемого картой на изображении, должна быть контекстно-чувствительная подпись. В описательной части должна отображаться информация о фрагментах изображения.

Примерные темы для задания:

- 1) Выращивание кактусов в домашних условиях.
- 2) Столицы Европы.
- 3) Столицы крупнейших городов Азии.
- 4) Подводная фотосъемка.
- 5) Аквариумные рыбы.
- 6) Блюда из ананасов.
- 7) Технологии вязания подарочных бантов.
- 8) Освоение космоса.
- 9) Обитатель подводных глубин.
- 10) Ремонт велосипедов.
- 11) Компьютерные игры.
- 12) Техники росписи деревянной посуды.
- 13) Интересные факты из жизни комедийных актеров.
- 14) Как устроен дизельный двигатель.
- 15) Любая выбранная тема...

Тема 4. Основы языка программирования JavaScript

Лабораторная работа № 3 (JavaScript)

"JavaScript -код в тексте HTML -страниц. Интерпретация кода на стороне клиента для поддержки диалога с пользователем".

Вопросы допуска:

- 1) Язык программирования JavaScript.
- 2) JavaScript интерпретаторы, создание клиентской части приложений.
- 3) Node.js.
- 4) JavaScript библиотеки.

Задания по вариантам:

1. Напишите скрипт, который запрашивает таблицу чисел размерностью 10x10 с проверкой правильности ввода и выводит в окне браузера наибольшее, наименьшее и среднеарифметическое значение введенных чисел.
2. Напишите скрипт, который запрашивает дату (число/месяц/год) и выводит следующую информацию: время года, високосный или нет год, какому зодиакальному знаку соответствует данная дата, какой это год по восточному календарю и сколько дней, недель, месяцев и лет разделает нас от текущей даты.
3. Напишите скрипт, который отображает диалоговое окно с кнопками «калькулятора» и при выполнении простейших математических операций скрипт выводит в окно браузера результат выполнения операции.

4. Напишите скрипт, запрашивающий количество учащихся студенческой группы по следующим критериям: дата рождения, успеваемость по различным предметам, место рождения, место проживания и в соответствии с выбранными параметрами выводит их Ф.И.О.. Ф.И.О. необходимо отобразить в окне браузера. Необходимо предусмотреть возможность просмотра всей хранимой таблицы.
5. Напишите скрипт, который выводит в окно браузера таблицу умножения чисел размерностью 10x10 в диапазоне чисел задаваемом пользователем.
6. Напишите скрипт, который выводит цитату дня на экран на основе введенного пользователем слова или словосочетания (за основу берутся однокоренные слова, встречающиеся в том или ином высказывании).
7. Подготовьте веб странички на нескольких иностранных языках (не менее 5). Напишите скрипт, позволяющий пользователям выбирать для просмотра страницу на одном из языков.
8. Напишите скрипт, который проверяет правильность ввода в поля формы. Пусть у нас есть некоторая анкета, в которую надо ввести фамилию, возраст, мастерство и e-mail и т.д. пользователя (количество полей должно быть не менее 30). При этом необходимо обозначить обязательные и необязательные поля (количество не менее половины). Должна быть выполнена проверка на корректность ввода информации.
9. Написать скрипт, содержащий текущее время, дату, часовой пояс, название административного субъекта РФ в виде работающих электронных часов с картой РФ
10. Задан текст (объемом 0,5 стр. формата А4). Написать скрипт, который определяет количество каждого используемого в тексте символа и выводит на экран таблицу частоты появления каждого символа с возможностью сортировки по алфавиту и убыванию или возрастанию числа повторений.
11. Написать скрипт, который выводит на экран бегущую строку, определяет и выводит текущее разрешение экрана, определяет браузер и его версию.
12. Написать скрипт, который организует работу выпадающего меню и динамической смены слоев текста и графики на странице html документа (смена слоев происходит при наведении нажатии клавиши мыши: левая клавиша переключение слоев в прямом порядке, правая клавиша – обратный порядок).
13. Написать скрипт, который реализует метод Крамера с размерностью матрицы 4x4, ввод первоначальных данных осуществляется пользователем, в случае, если решения нет или данные введены не корректно, выводится сообщение с соответствующим текстом.
14. Написать скрипт, который реализует метод обратной матрицы с размерностью матрицы 5x5, ввод первоначальных данных осуществляется пользователем, в случае, если решения нет или данные введены не корректно, выводится сообщение с соответствующим текстом.
15. Написать скрипт, который реализует метод Гаусса с размерностью матрицы 6x6, ввод первоначальных данных осуществляется пользователем, в случае, если решения нет или данные введены не корректно, выводится сообщение с соответствующим текстом.
16. Написать скрипт, который создает тест (не менее 15 вопросов) на основе заложенных в него вопросов с обработкой результатов (вывод: времени, затраченного на выполнение, количества верно, неверно или частично правильно полученных ответов, среднего балла, оценки).

Написать скрипт, который реализует автоматическое управление перемещением шариков по полю с бардюрами (предусмотреть случай столкновения, если шариков на поле несколько).

Тема 5. Основы технологии PHP

Лабораторная работа № 4 (PHP. Часть 1) «Основы PHP»

Вопросы допуска:

- 1) Определение, Общее представление.
- 2) Основы PHP

Работа состоит из двух частей.

Часть 1. Написать HTML-документ и PHP-сценарий для решения задачи вашего варианта. HTML-документ служит для ввода исходных данных и вызывает PHP-сценарий. PHP-сценарий выполняет обработку данных и формирует новую HTML-страницу с результатами решения задачи.

Часть 2. Написать PHP-сценарий для решения задачи вашего варианта. При первом обращении PHP-сценарий формирует HTML-страницу для ввода исходных данных, и эта страница вызывает вновь PHP-сценарий. При втором вызове PHP-сценарий выполняет обработку введенных пользователем данных и формирует HTML-страницу с результатами решения задачи.

Задания по вариантам:

1. Отредактировать заданное предложение, удаляя из него все слова, целиком составленные из вхождений не более чем двух букв.
2. В предложении слова отделяются пробелами. Вывести слово с заданным номером.
3. Даны два предложения. Найти самое короткое из слов первого предложения, которого нет во втором предложении.
4. Найти множество всех слов, которые встречаются в каждом из двух заданных предложений.
5. В предложении все слова начинаются с различных букв. Напечатать (если это возможно) слова предложения в таком порядке, чтобы последняя буква каждого слова совпадала с первой буквой следующего слова.
6. Из заданного текста выбрать и напечатать те символы, которые встречаются в нем ровно один раз (в том порядке, как они встречаются в тексте).
7. Отредактировать заданное предложение, удаляя из него те слова, которые уже встречались в предложении раньше.
8. Расстояние между двумя словами равной длины - это количество позиций, в которых различаются эти слова. В заданном предложении найти пару наиболее далеко удаленных слов заданной длины.

9. Найти самое длинное симметричное слово заданного предложения.
10. Для каждого из слов заданного предложения указать, сколько раз оно встречается в предложении.
11. В заданном предложении найти пару слов, из которых одно является обращением другого.
12. Перечислить все слова заданного предложения, которые состоят из тех же букв, что и первое слово предложения.
13. В предложении слова отделяются пробелами. Сформировать и вывести "слово", выполнив конкатенацию первого и последнего слов предложения.
14. Для заданного текста определить длину содержащейся в нем максимальной серии символов, отличных от букв.
15. Для встречающихся в заданном тексте пар рядом расположенных символов указать, сколько раз встречается в тексте каждое из таких двухбуквенных сочетаний.
16. Пусть последовательность символов в строке представляет собой алгебраическую сумму вида: <знак><число><знак><число><знак> ... <знак><число> где "число" - целое натуральное число, "знак" - аддитивная операция (плюс или минус), либо для первого слагаемого его знак (+, -). Вычислить и вывести значение этой суммы. Пример: строка: "-4+22-11", результат: 7
17. Пусть последовательность символов в строке представляет собой алгебраическую сумму вида: <знак><цифра><знак><цифра><знак> ... <знак><цифра> где "цифра" - изображение десятичной цифры от 0 до 9, "знак" - аддитивная операция (плюс или минус), либо знак для первого слагаемого (+, -). Вычислить и вывести значение этой суммы. Пример: строка: "-2-8+4+2", результат: -4
18. Пусть в строке находится символьное изображение десятичного числа (возможно со знаком, возможно с десятичной точкой, возможно с порядком). Сформировать по изображению числа его значение и присвоить его целой N или вещественной X переменной (в зависимости от типа числа). В изображении вещественного числа должна быть точка и/или признак (разделитель) для обозначения порядка: 'e' или 'E'. Вывести тип числа и его значение. Примеры: строка: "0.124E1", результат: Вещественное число X=1.24 строка: "124", результат: Целое число N=124
19. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Выбрать из предложения все слова, начинающиеся и заканчивающиеся на одну и ту же букву. Вывести выбранные слова, либо сообщение об их отсутствии.
20. Из последовательности символов удалить все лишние пробелы, т.е. все пробелы в начале, все пробелы в конце, а внутри последовательности вместо повторяющихся подряд пробелов оставить по одному. Вывести полученную строку.
21. Проверить, является ли последовательность символов правильным представлением десятичного вещественного числа (возможно со знаком). Вывести сообщение о результате проверки. (Целое число считать частным случаем вещественного.)

22. Проверить баланс скобок в строке, т.е. убедиться, что каждой открывающей скобке "(", "{", "[", "<", "/" соответствует закрывающая ")", "}", "]", ">", "*"/. При нарушении баланса скобок выдать сообщение с изображением недостающей скобки и указанием количества "разбалансов" ("несоответствий"). Например, для строки: "{дорога { без [начала]] и] конца }" результат будет таким: "} - 1", "[- 2"

23. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Упорядочить слова в предложении по возрастанию длин слов. Вывести измененное предложение.

24. В строке найти все вхождения заданной подстроки (их не должно быть больше девяти) и заменить их на другую подстроку, дополнив ее номером очередного вхождения. Пример: исходная строка: "полили лилию" искомая подстрока: "ли" заменяющая подстрока: "сто" Результат: "посто1сто2 сто3сто4ю"

25. Упорядочить последовательность символов в строке таким образом, чтобы вначале шли все цифры, затем все пробелы, затем прочие символы. Вывести полученную строку.

26. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Упорядочить символы внутри отдельных слов по возрастанию кодов символов. Вывести измененное предложение.

27. Пусть строка содержит предложение на английском языке, а затем, после символа "-" - его транскрипцию русскими буквами. Все слова разделены пробелами. Например: "I don't know - Ай донт ноу" Сформировать массив строк, каждая из которых содержит пару: "английское слово - транскрипция". Вывести в столбик слова с транскрипциями. Для приведенного примера результат будет таким: I - Ай don't - донт know - ноу

28. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Вывести все слова предложения в столбик, т.е. каждое слово напечатать на новой строке.

29. Проанализировать последовательность символов в строке и определить количество различающихся символов, отличных от пробела. Вывести все различающиеся символы с указанием количеств вхождений каждого из них в анализируемую последовательность. Например, введя последовательность "12B1R21", получим: "1 - 3", "2 - 2", "B - 1", "R - 1".

30. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Введя с клавиатуры произвольное слово, проверить, входит ли оно в предложение. Вывести номер его последнего вхождения в предложение (порядковый номер слова от начала предложения), либо предупреждающее сообщение.

31. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Переставить в обратном порядке (поменять местами) слова предложения. Например, из предложения "кризис жанра." сформировать "жанра кризис.". Вывести преобразованное предложение.

32. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Инвертировать все слова предложения, представленного строкой, оставляя их на прежних местах. Например, строку "воз и ныне там." заменить строкой "зов и нын мат.". Вывести преобразованное предложение.

33. Пусть строка представляет предложение, в конце которого стоит точка, а слова отделены друг от друга пробелами. Введя произвольный номер слова в предложении, выбрать из строки и вывести соответствующее слово. Если слово с заданным номером в предложении отсутствует, вывести соответствующее предупреждающее сообщение.

34. Проверить символы строки и если встретится символ, отличный от буквы русского алфавита (в выбранной вами кодировке), пробела или знака препинания, то вывести его порядковый номер в строке и изображение символа.

35. Найти в строке середину и зеркально отобразить половинки строки относительно их средин.

36. Определить частоту вхождения каждой буквы алфавита в текст и вывести список частот (буква - цифра) в алфавитном порядке. Буквы с нулевой частотой не выводить. Пример: "Привет, мир". Результат: в-1, е-1, и-2, м-1, П-1, р-2, т-1.

37. Зашифровать введенный текст шифром Цезаря и вернуть результат. Шифр Цезаря заключается в увеличении кода символа на 2 по модулю количества символов в алфавите. Например: КОНЧИЛ ДЕЛО, ГУЛЯЙ СМЕЛО -> МРПЩКН ЁЖНР ЕХНБЛ УОЖНР

38. Переписать все слова предложения (слово - последовательность букв русского алфавита) задом наперед.

39. Найти последнее слово во введенной строке (слово - последовательность букв латинского алфавита) и зеркально отобразить его относительно середины слова.

Лабораторная работа № 5 (РНР. Часть 2)

«Работа с web-формами»

Задания по вариантам:

ВАРИАНТ № 1

Определить HTML-документ для ввода сведений о студенте:

- ·фамилия и инициалы;
- ·номер группы;
- ·успеваемость (поля для пяти дисциплин).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка конкретной группы в алфавитном порядке (по фамилиям) с указанием названий дисциплин и оценок;
- для вывода списка неуспевающих студентов (с указанием дисциплин, по которым нет аттестации или получена оценка 2).

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем студенте.

ВАРИАНТ № 2

Определить HTML-документ для ввода сведений о студенте:

- фамилия и инициалы;
- номер группы;
- успеваемость (поля для пяти дисциплин).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка фамилий и номеров групп студентов, имеющих оценки 4 и 5;
- для вывода списка студентов, упорядоченного по возрасту среднего (для каждого студента) балла по пяти дисциплинам;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем студенте.

ВАРИАНТ № 3

Определить HTML-документ для ввода сведений о студенте:

- фамилия и инициалы;
- номер группы;
- успеваемость (поля для пяти дисциплин).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода упорядоченного по фамилиям списка всех студентов с указанием номеров групп и названий дисциплин с оценками.
- для вывода списка студентов, имеющих только отличные оценки.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем студенте.

ВАРИАНТ № 4

Определить HTML-документ для ввода сведений о авиарейсе:

- пункт назначения авиарейса;
- номер рейса;
- время вылета;
- тип самолёта.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода упорядоченного по времени вылета списка авиарейсов, вылетающих в пункт с указанным названием;
- для вывода упорядоченного по типам самолета списка всех авиарейсов.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем авиарейсе.

ВАРИАНТ № 5

Определить HTML-документ для ввода сведений о авиарейсе:

- пункт назначения авиарейса;
- номер рейса;
- время вылета;
- тип самолёта.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех авиарейсов, упорядоченных по названиям пунктов назначения;
- для вывода списка авиарейсов, вылетающих позже указанного времени;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем авиарейсе.

ВАРИАНТ № 6

Определить HTML-документ для ввода сведений о сотруднике:

- фамилия, имя, отчество работника;
- название занимаемой должности;
- год поступления на работу.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода сведений обо всех сотрудниках, упорядоченных в алфавитном порядке по фамилиям сотрудников;
- для вывода сведений обо всех сотрудниках, чей стаж работы превышает введенное значение.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем сотруднике.

ВАРИАНТ № 7

Определить HTML-документ для ввода сведений о поезде:

- название пункта назначения;
- номер поезда;
- время отправления.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода сведений обо всех поездах, упорядоченных в алфавитном порядке по пунктам назначения;
- для вывода сведений обо всех поездах, отправляющихся позже указанного времени в названный пункт назначения.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем поезде.

ВАРИАНТ № 8

Определить HTML-документ для ввода сведений о поезде:

- ·название пункта назначения;
- ·номер поезда;
- ·время отправления.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех поездов, упорядоченный по времени отправления;
- для вывода сведений о поезде с заданным номером.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем поезде.

ВАРИАНТ № 9

Определить HTML-документ для ввода сведений о поезде:

- ·название пункта назначения;
- ·номер поезда;
- ·время отправления.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех поездов, упорядоченный по номерам поездов;
- для вывода сведений обо всех поездах, отправляющихся в названный пункт назначения.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем поезде.

ВАРИАНТ № 10

Определить HTML-документ для ввода сведений о туристическом маршруте:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех маршрутов, упорядоченный по номерам;
- для вывода сведений обо всех маршрутах, начинающихся в названном пункте.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем маршруте.

ВАРИАНТ № 11

Определить HTML-документ для ввода сведений о туристическом маршруте:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех маршрутов, упорядоченный по названиям начальных пунктов;
- для вывода двух списков: маршруты отправляемые - начинающиеся в названном пункте, и маршруты принимаемые - заканчивающиеся в том же пункте.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем маршруте.

ВАРИАНТ № 12

Определить HTML-документ для ввода сведений о человеке:

- фамилия, имя;
- номер телефона;
- день рождения (три числа).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех людей, упорядоченный по датам их рождения;
- для вывода сведений о человеке, номер телефона которого задан.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем человеке.

ВАРИАНТ № 13

Определить HTML-документ для ввода сведений о человеке:

- фамилия, имя;
- номер телефона;

- ·день рождения (три числа).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех людей, в алфавитном порядке их фамилий;
- для вывода сведений о людях, дни рождения которых приходятся на заданный месяц.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем человеке.

ВАРИАНТ № 14

Определить HTML-документ для ввода сведений о человеке:

- ·фамилия, имя;
- ·номер телефона;
- ·день рождения (три числа).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех людей, в порядке номеров их телефонов;
- для вывода сведений о человеке, фамилия которого задана.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем человеке.

ВАРИАНТ № 15

Определить HTML-документ для ввода сведений о человеке с учетом знака Зодиака:

- ·фамилия, имя;
- ·знак Зодиака;
- ·день рождения (массив из трёх чисел).

Овен (21 марта-20 апреля); Телец (21 апреля - 21 мая); Близнецы (22 мая - 22 июня); Рак (23 июня - 23 июля); Лев (24 июля - 23 августа); Дева (24 августа - 23 сентября); Весы (24 сентября - 23 октября); Скорпион (24 октября - 22 ноября); Стрелец (23 ноября - 21 декабря); Козерог (22 декабря -20 января); Водолей (21 января - 19 февраля); Рыбы (20 февраля - 20 марта).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка сведений о всех людях, упорядоченного по датам рождения;
- для вывода сведений о человеке, фамилия которого задана.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем человеке.

ВАРИАНТ № 16

Определить HTML-документ для ввода сведений о человеке с учетом знака Зодиака:

- фамилия, имя;
- знак Зодиака;
- день рождения (массив из трёх чисел).

Овен (21 марта-20 апреля); Телец (21 апреля - 21 мая); Близнецы (22 мая - 22 июня); Рак (23 июня - 23 июля); Лев (24 июля - 23 августа); Дева (24 августа - 23 сентября); Весы (24 сентября - 23 октября); Скорпион (24 октября - 22 ноября); Стрелец (23 ноября - 21 декабря); Козерог (22 декабря -20 января); Водолей (21 января - 19 февраля); Рыбы (20 февраля - 20 марта).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка сведений о всех людях, упорядоченного по их фамилиям (по алфавиту);
- для вывода списка сведений о людях, родившихся под указанным знаком Зодиака.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем человеке.

ВАРИАНТ № 17

Определить HTML-документ для ввода сведений о человеке с учетом знака Зодиака:

- фамилия, имя;
- знак Зодиака;
- день рождения (массив из трёх чисел).

Овен (21 марта-20 апреля); Телец (21 апреля - 21 мая); Близнецы (22 мая - 22 июня); Рак (23 июня - 23 июля); Лев (24 июля - 23 августа); Дева (24 августа - 23 сентября); Весы (24 сентября - 23 октября); Скорпион (24 октября - 22 ноября); Стрелец (23 ноября - 21 декабря); Козерог (22 декабря -20 января); Водолей (21 января - 19 февраля); Рыбы (20 февраля - 20 марта).

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка сведений о всех людях, упорядоченного по календарному расположению знаков Зодиака;
- для вывода списка сведений о людях, родившихся в указанный месяц.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем человеке.

ВАРИАНТ № 18

Определить HTML-документ для ввода сведений о товаре:

- ·название товара;
- ·название магазина, в котором продаётся товар;
- ·стоимость товара в рублях.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода "сведений для покупателя" - список всех товаров, в алфавитном порядке их названий;
- для вывода сведений о конкретном товаре (с ценами в разных магазинах).

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем товаре.

ВАРИАНТ № 19

Определить HTML-документ для ввода сведений о товаре:

- ·название товара;
- ·название магазина, в котором продаётся товар;
- ·стоимость товара в рублях.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех товаров, упорядоченных (и сгруппированных) по названиям магазинов;
- для вывода сведений о товарах, цены на которые не превышают заданной.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем товаре.

ВАРИАНТ № 20

Определить HTML-документ для ввода сведений о товаре:

- ·название товара;
- ·название магазина, в котором продаётся товар;
- ·стоимость товара в рублях.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех товаров, упорядоченных по ценам;
- для вывода сведений о товарах, которые имеются в указанном магазине.

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем товаре.

ВАРИАНТ № 21

Определить HTML-документ для ввода сведений о государстве:

- ·название страны;
- ·название столицы;
- ·численность населения;
- ·площадь территории.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех государств, упорядоченных по названиям стран;
- для вывода сведений о государствах, в которых плотность населения ниже заданной;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем государстве.

ВАРИАНТ № 22

Определить HTML-документ для ввода сведений о государстве:

- ·название страны;
- ·название столицы;
- ·численность населения;
- ·площадь территории.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех государств, упорядоченных по численности населения;
- для вывода сведений о государстве с указанной столицей;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем государстве.

ВАРИАНТ № 23

Определить HTML-документ для ввода сведений о государстве:

- ·название страны;
- ·название столицы;
- ·численность населения;
- ·площадь территории.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех государств, упорядоченных по названиям столиц;

- для вывода сведений о государстве, в котором плотность населения ближе всего к заданной;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем государстве.

ВАРИАНТ № 24

Определить HTML-документ для ввода сведений о государстве:

- ·название страны;
- ·название столицы;
- ·численность населения;
- ·площадь территории.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех государств, упорядоченных по площадям территорий;
- для вывода сведений о государстве с указанным названием;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем государстве.

ВАРИАНТ № 25

Определить HTML-документ для ввода сведений о государстве:

- ·название страны;
- ·название столицы;
- ·численность населения;
- ·площадь территории.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех государств, упорядоченных по плотности населения стран;
- для вывода сведений о государстве, в котором численность населения ближе всего к заданной;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем государстве.

ВАРИАНТ № 26

Определить HTML-документ для ввода сведений об авиарейсе:

- название пункта назначения авиарейса;
- номер рейса;
- тип самолёта.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех рейсов, упорядоченных (сгруппированных) по типам самолетов;
- для вывода сведений о рейсе по заданному номеру;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем государстве.

ВАРИАНТ № 27

Определить HTML-документ для ввода сведений об авиарейсе:

- название пункта назначения авиарейса;
- номер рейса;
- тип самолёта.

Данные вводятся в произвольном порядке. Предусмотреть три "выхода" со страницы:

- для передачи введенных данных на сервер, где они должны быть занесены в файл;
- для вывода списка всех рейсов, упорядоченных по номерам рейсов;
- для вывода сведений о всех рейсах, выполняемых самолетами заданного типа;

После передачи на сервер данных отображение в окне браузера должно быть подготовлено для ввода сведений о следующем государстве.

Тема 6. Создание Web-приложений средствами PHP.

Лабораторная работа № 6

«Язык PHP»

Теоретическая часть

1. Примеры объявления переменных:

`$a = 1;` - целочисленный тип

`$b = 2.45;` - вещественный тип

`$c = "string";` - строковый тип

Комментарии:

`//` однострочный комментарий

`#` это тоже однострочный комментарий

`/*` многострочный
комментарий `*/`

2. Основные команды и примеры:

`if (<условие>) <команда> [ else <команда> ]`

`if (<условие>) <команда> elseif (<условие>) <команда>`

`if (x > y) print "x>y";`

`elseif (x<y) print "x<y";`

`else print "x=y";`

`for(<инициализация>;<условие>;<приращение>) <команда>`

`for ($i=0; $i < 10; $i++) $fact = $fact * $i;`

continue; - продолжить цикл, пропустив нижележащие команды
 break; - прекратить цикл
 mt_rand (<минимум>, <максимум>); - сгенерировать случайное число
 \$num = mt_rand(10,50);

3. Основные функции вывода:

print <аргумент>; - вывести аргумент

Пример: print "Hello World!";

echo <аргумент>; - вывести группу аргументов

Пример: echo "Hello World!", "Hello Earth!", "Hello Sky!";

printf (<формат>[, <аргументы>,...]); - C-подобная функция

Пример: printf ("my course is %d%s\n", 4, "
");

4. Пример PHP-скрипта:

```
<html>
  <body>
    <?php
      print "Hello Programmer!<br>";
      for ($i = 0; $i < 10; $i++)
        printf ("Serial number is %d<br>", $i);
      printf ("Random number is %d<br>", mt_rand(0,100));
    ?>
  </body>
</html>
```

Задание:

1. Динамически сгенерировать таблицу данных 50 строк на 15 столбцов с помощью генератора случайных чисел. Подсчитать статистические данные по строкам и столбцам – среднее значение, максимальное значение и сумма. Результат показать в браузере. Примерный вид таблицы:

	1	2	...	15	<i>среднее</i>	<i>максимальное</i>	<i>сумма</i>
1	12	29	...	45	34	67	345
2	43	23	...	24	54	89	254
...	...	...	...	...	...	...	...
50	12	87	...	27	23	95	259
<i>среднее</i>	34	56	...	45			
<i>максимальное</i>	98	89	...	87			
<i>сумма</i>	345	674	...	345			

2. Сгенерировать строку из 100 символов с различным типом форматирования. Символы и форматирование получаются случайным образом. Для форматирования применяются тэги , <u></u>, <i></i> и их комбинации. Результат показать в браузере. Примерный вид строки:

djklsgjk~~lf~~mbdgcjdfgkqwxfdgzdfaglj

Лабораторная работа № 7

«Обмен информацией между WEB-сервером и клиентом»

Теоретическая часть

Примеры создания полей ввода и кнопок можно посмотреть в лекции или спецификации языка HTML. Используемые тэги:

<FORM> - создание формы

<INPUT> - создание полей редактирования, кнопок различных видов

<SELECT> - создание списка выбора и комбинированного списка

Практическая часть

Программа «Получение анкетных данных от пользователя»

1. Разработать HTML-файл для возможности получения от пользователя автобиографической информации. Для этого необходимо использовать следующие виды объектов экранной формы: поле редактирования, радио-кнопки, комбинированный список, список с множественным выбором, кнопки-переключатели.

Пример, вводимых данных:

- ФИО – поля редактирования
- Пол (мужской, женский) – радио-кнопки
- семейное положение (холост, женат/замужем) – радио-кнопки
- образование (среднее, высшее и т.п.) – комбинированный список
- хобби (спорт, охота, рыбалка, компьютеры, пиво, ТВ и т.п.) – список с множественным выбором
- есть ли компьютер – кнопка-переключатель

2. Разработать PHP-скрипт, анализирующий введенные данные и генерирующий введенную информацию в виде списка автобиографических данных по каждому из введенных лиц.

Пример:

Иванов Иван Иванович

Пол: *мужской*

Семейное положение: *холост*

Образование: *высшее*

Хобби: *охота, рыбалка*

Наличие ПК дома: *есть*

3. Использовать поочередно в п.1 методы GET и POST. Посмотреть в чем различия. Разобрать формат передачи параметров в методе GET.

Тема 7. Работа с базами данных в web-приложениях

Лабораторная работа № 8

«Разработка диалоговых систем с помощью Web-технологий»

Задачи:

1. Научиться переносить данные между страницами
2. Научиться переадресовывать запросы

Требования к системе. Разработать систему, выполняющую следующие функции:

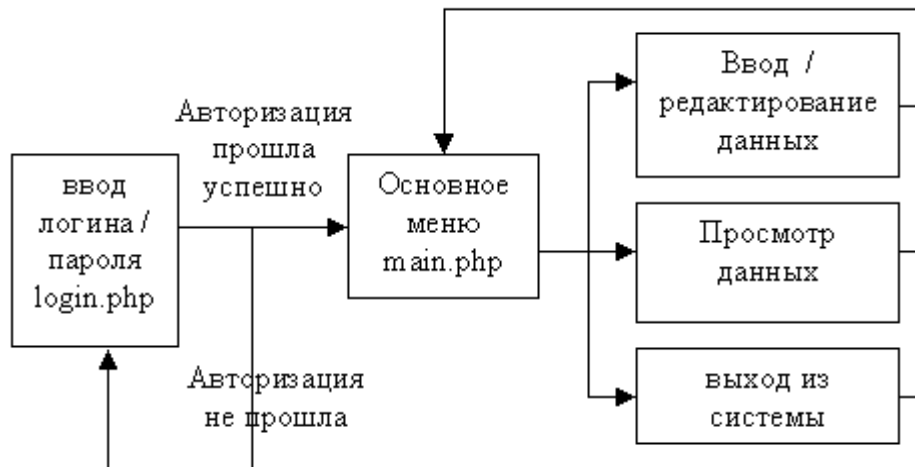
1. Авторизация введенных данных о пользователе. После авторизации каждая страничка должна содержать ФИО «залогинившегося» пользователя
2. Ввод данных от пользователя
3. Редактирование введенных данных
4. Просмотр введенных данных

Система состоит из 4-х модулей:

1. Страница ввода идентификационных данных (логин/пароль)

2. Главное меню
3. Страница ввода/редактирования данных
4. Просмотр введенных данных

Функциональная схема работы:



Практическая реализация

Данная лабораторная работа опирается на предыдущую лабораторную работу №2 «Обмен информацией между WEB-сервером и клиентом». В качестве примера для реализации можно взять систему работы с анкетными данными. Работа системы начинается со страницы «ввод логина/пароля». В случае удачной аутентификации пользователь попадает на «основное меню», из которого он может ввести/поменять/просмотреть данные о себе или выйти из системы. Переадресация на другую страницу выполняется с помощью команды протокола HTTP, реализуемую через:

```
print header ("Location: login.php");
exit;
```

Данная команда должна идти первой из выводимого на поток вывода (до тега <html>), иначе переадресации не случится.

Требование к безопасности: «неавторизованные» пользователи не должны видеть ни одной странички, кроме «ввод логина/пароля». В случае, если «неавторизованный» пользователь пытается вызвать любую страницу, система должна адресовать его на страницу «ввод логина/пароля».

Простейший пример реализации механизма аутентификации.

Создаются две функции:

get_user_id(\$login, \$password); - в случае успешной проверки возвращает идентификатор пользователя, в случае неудачи – возвращает -1.

get_correct_login(\$user_id); - проверяет корректность идентификации. В случае неудачи переадресует на страницу «ввод логина/пароля».

Файл «authlib.inc»

```
<?php
$passwd["Ivanov"]["id"] = 1289;
$passwd["Ivanov"]["password"] = "pass1";
$passwd["Petrov"]["id"] = 1423;
$passwd["Petrov"]["password"] = "table";
$passwd["Sidorov"]["id"] = 4598;
$passwd["Sidorov"]["password"] = "nothing";
```

```

function get_user_id($login, $password) {
    GLOBAL $passwd;
    if ($passwd[$login]["password"] == $password) {
        return $passwd[$login]["id"];
    } else {
        return -1;
    }
}
function get_correct_login($user_id) {
    GLOBAL $passwd;
    return in_array($user_id, $passwd);
}

```

?>

После чего можно использовать эту библиотеку в страницах командой
`include("authlib.inc");`

После чего нужно идентифицировать пользователя:

```

$id = get_user_id($login, $password);
if ($id == -1) {
    print header ("Location: login.php");
    exit;
}

```

Для передачи параметров между страницами в формах используется тэг
`<INPUT type="HIDDEN">`. Данный тег не отображается на форме визуально и предназначен для передачи скрытых параметров, принцип передачи – как у `<INPUT type="TEXT">`.
 Пример передачи идентификатора пользователя:
`<INPUT type="HIDDEN" name="user_id" value="<?php print $user_id; ?>">`

После чего для организации безопасности каждый файл можно начинать со строки:

```

$id = get_correct_login($user_id);
if ($id == -1) {
    print header ("Location: login.php");
    exit;
}

```

Тема 8. Основы технологии MySQL.

Лабораторная работа № 9

«Знакомство с СУБД MySQL»

Задание

Разработать информационно-справочную систему, например, «телефонный справочник».

База данных системы должна включать, как минимум, 10 типов строк по каждой записи и не менее 20 видов записей.

Например:

1. Фамилия.
2. Имя.
3. Отчество.
4. Телефон.

5. Адрес проживания.
6. И т.д.

Система должна выполнять следующие операции:

1. Ввод и корректировка данных администратором системы.
2. Поиск и вывод информации по различным критериям, не менее 5 видов (определить самостоятельно).

Система состоит из двух независимых блоков:

1. Блок ввода информации в базу администратором
2. Блок вывода информации по запросам пользователей

Блок вывода информации по запросам пользователей состоит из двух модулей. Первый модуль запрашивает информацию для поиска в базе (фамилию или телефон). Второй модуль, на основании критериев поиска, формирует запрос к базе, получает ответ и формирует вывод результатов поиска в виде таблицы

Блок ввода информации в базу позволяет вводить и изменять данные только администратору БД. Перед работой в этом блоке необходима проверка прав доступа

Теоретический материал

MySQL - Система Управления Базами Данных (СУБД). Основное отличие от всех остальных СУБД это то, что она является бесплатной. В силу того, что MySQL бесплатна, она поддерживается очень многими хостинг провайдерами.

Для связи PHP и MySQL нужно знать:

- Хост - адрес сервера баз данных MySQL
- Имя базы данных
- Имя пользователя
- Пароль - пароль для доступа к БД
- Некоторые SQL команды

Алгоритм:

1. Устанавливаем соединение с сервером
2. Выбираем нужную базу данных
3. Делаем запрос к серверу баз данных (SQL запрос)
4. Обрабатываем результат запроса, если это необходимо
5. Закрываем соединение (отключаемся от сервера БД)

Для части PHP, чтобы работать с БД необходимо знать несколько функций.

- `mysql_connect();`
- `mysql_select_db();`
- `mysql_query();`
- `mysql_fetch_array();`
- `mysql_close();`

Это тот минимум, который позволит устанавливать соединение с сервером MySQL, выбирать базы данных и делать запросы к серверу баз данных.

Рассмотрим эти функции в работе, т.е. на примере.

Допустим в базе данных храниться одна таблица со следующей структурой:

имя поля	тип(длина)	описание
Id	целый(2)	уникальный номер
name	символьный(100)	имя
Tel	символьный(20)	телефон

Таблица имеет имя: **customer**, а база данных называется **database**.

Допустим, что данные для подключения к серверу MySQL такие:

- Хост: **localhost**

- Имя пользователя: **Guest**
- Пароль: **mypassword**

Для начала необходимо создать нашу таблицу **customer**. Напишем скрипт, который создаст в базе данных **database** таблицу **customer**. Назовём файл как **install.php**.

Файл **install.php**

```
<?php
```

```
// Данные для MySQL сервера
$DBHost = "localhost";           // Хост
$DBUser = "Guest";               // Имя пользователя
$DBPassword = "mypassword";      // Пароль
$DBName = "database";           // Имя базы данных

// Подключаемся к MySQL серверу
$link = mysql_connect($DBHost, $DBUser, $DBPassword);

// Выбираем нашу базу данных
mysql_select_db($DBName, $link);

// Создаём таблицу customer
// т.е. делаем SQL запрос
$query = "CREATE TABLE customer (id INT(2) PRIMARY KEY
AUTO_INCREMENT, name VARCHAR(100), tel VARCHAR(20))";
mysql_query($query, $link);

// Закрываем соединение
mysql_close($link)
```

```
?>
```

Функция `mysql_connect()` возвращает идентификатор соединения. Этот идентификатор необходимо указывать во всех MySQL функциях. Можно провести аналогию с указателем на файл (file pointer), который используется функциями по работе с файлами. В параметрах функции указали хост, имя пользователя и пароль к базе данных.

На сервере MySQL храниться не только ваша база данных. Сервер может обслуживать тысячи таких баз данных. Поэтому, чтобы получить доступ к своей базе данных, необходимо выбрать её. Выбор базы данных осуществляется функцией `mysql_select_db()`. В параметрах указывается: имя требуемой базы данных `$DBName` и идентификатор соединения `$link`, который мы получили с помощью функции `mysql_connect()`.

После того как мы подключились к серверу MySQL и выбрали нашу базу данных, мы делаем SQL запрос. В запросе указываем, что мы хотим создать таблицу с именем **customer** и со структурой, приведённой в таблице выше. Запрос к серверу MySQL осуществляется с помощью функции `mysql_query()`. В параметрах указывается сам запрос `$query` и идентификатор соединения `$link`.

И, наконец, отключаемся от сервера (закрываем соединение) функцией `mysql_close()`. В параметрах указываем идентификатор того соединения, которое нужно закрыть.

Если вы всё сделали правильно, то на сервере MySQL в вашей базе данных, появится таблица **customer**.

После того как создали таблицу, её надо заполнить т.е. записать в неё какие-нибудь данные (записи). Напишем скрипт, который будет добавлять запись в нашу таблицу. Назовём файл скрипта как **insert.php**.

Файл **insert.php**

```
<?php
```



```

// Данные для MySQL сервера
$DBHost = "localhost";           // Хост
$DBUser = "Guest";               // Имя пользователя
$DBPassword = "mypassword";      // Пароль
$DBName = "database";           // Имя базы данных

// Подключаемся к MySQL серверу
$link = mysql_connect($DBHost, $DBUser, $DBPassword);

// Выбираем нашу базу данных
mysql_select_db($DBName, $link);

// Добавляем запись в нашу таблицу customer
// т.е. делаем SQL запрос
$query = "INSERT INTO customer VALUES(0, 'Иванов Иван
Иванович',
      '(095) 555-55-55')";

mysql_query($query, $link);

// Закрываем соединение
mysql_close($link);

```

?>

Посмотрев этот скрипт, вы заметите, что он отличается от предыдущего только строкой запроса \$query.

Теперь когда мы умеем записывать данные в базу данных, перейдём к процедуре запроса данных из базы MySQL. Для этого напишем скрипт и назовём файл select.php.

Файл select.php

```
<?php
```

```

// Данные для MySQL сервера
$DBHost = "localhost";           // Хост
$DBUser = "Guest";               // Имя пользователя
$DBPassword = "mypassword";      // Пароль
$DBName = "database";           // Имя базы данных

// Подключаемся к MySQL серверу
$link = mysql_connect($DBHost, $DBUser, $DBPassword);

// Выбираем нашу базу данных
mysql_select_db($DBName, $link);

// Добавляем запись в нашу таблицу customer
// т.е. делаем SQL запрос
$query = "SELECT * FROM customer";

// Запрашиваем
$result = mysql_query($query, $link);

while($rows = mysql_fetch_array($result, MYSQL_ASSOC))
{

```

```

        printf("id:%d, name:%s, tel:%s",
               Rows['id'], Rows['name'], Rows['tel']);
    }

    // Закрываем соединение
    mysql_close($Link);

```

?>

Естественно изменилась строка запроса `$Query`, которая теперь содержит SQL-оператор `SELECT`. Так же добавился небольшой код `while() { .... }`. Скажу только, что функция `mysql_fetch_array()` обрабатывает результат запроса и возвращает массив полей текущей! Внимание! текущей строки результата.

Константа `MYSQL_ASSOC` указывает на то, что функция должна вернуть ассоциативный массив полей. После того как функция `mysql_fetch_array()` обработает все строки результата, она вернёт значение `FALSE` и тогда цикл `while()` - не выполниться.

Тут в цикле используем функцию `printf()`.

Лабораторная работа № 10

«Разработка информационной системы»

Задание

Разработать программное обеспечение для работы в заданной предметной области. Программное обеспечение является независимым от типа SQL-сервера. Работоспособность системы будет проверяться на MySQL.

На выбор предлагаются следующие предметные области:

- деканат института
- школа
- больница
- троллейбусное депо
- магазин
- склад
- и далее по аналогии (самостоятельно предложить свой вариант)

Программное обеспечение состоит из следующих частей:

1. **Блок инициализации** – создает начальные структуры таблиц на SQL-сервере и производит заполнение базовых справочников. Возможны два варианта реализации:
 - о Текстовый файл с набранными командами на SQL. Файл записывается на сервер, затем запускаются с консолей соответствующих SQL-серверов (mysql).
 - о Скрипт на PHP, который присоединяется к БД и создает все сам через метод `exec`.
2. **Блок корректировки** - заносит новую информацию предметной области в БД и справочники, корректирует имеющиеся данные.
3. **Блок генерации отчетов** – предоставляет возможность генерации отчетов по предметной области (естественно, в виде HTML).

Предметная область должна содержать (как минимум):

1. 3 таблицы (1 таблица + 2 ассоциированных справочника)
2. механизм работы с данными в этих таблицах
3. 2 любых отчета по этим таблицам на тематику предметной области

Примерная реализация предметной области «Деканат»

1. Необходимы следующие справочники и таблицы:
 - о Справочник «Учебные курсы» имеет поля (код предмета, наименование)

- о Справочник «Студенты» имеет поля (код студента, ФИО, группа, номер зачетки)
- о Таблица «Ведомость» имеет поля (код студента, код предмета, оценка)
- 2. Необходимо реализовать следующие механизмы доступа и работы с данными:
 - о Добавление/изменение/удаление информации из справочника «Учебные курсы»
 - о Добавление/изменение/удаление информации из справочника «Студенты»
 - о Добавление/изменение/удаление информации из таблицы «Ведомость»
- 3. Необходимо реализовать подготовку и вывод следующих отчетов:
 - о «Зачетная книжка» студента – по выбранной фамилии выводятся все предметы и оценки по ним для конкретного студента
 - о «Зачетный лист» группы – для выбранной группы и предмета формируется список студентов группы и оценок по предмету

Литература:

Список основной литературы

1. Зудилова, Т.В. Web-программирование JavaScript [Электронный ресурс] : учебно-методическое пособие / Т.В. Зудилова, М.Л. Буркова. – Электрон. дан. – Спб. : НИУ ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики), 2012. – 68 с. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=43561.

Список дополнительной литературы

1. Кнут, Дональд Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol. 1. Fundamental Algorithms. / Дональд Кнут // 3-е изд. — М.: «Вильямс», 2006. — 720 с. — ISBN 0-201-89683-4
2. Кнут, Дональд Искусство программирования, том 2. Получисленные методы = The Art of Computer Programming, vol. 2. Seminumerical Algorithms. / Дональд Кнут // 3-е изд. — М.: «Вильямс», 2007. — 832 с. — ISBN 0-201-89684-2
3. Кнут, Дональд Искусство программирования, том 3. Сортировка и поиск = The Art of Computer Programming, vol. 3. Sorting and Searching. / Дональд Кнут // 2-е изд. — М.: «Вильямс», 2007. — 824 с. — ISBN 0-201-89685-0
4. Иванова, Г.С. Технология программирования. Учебник для вузов./ Г.С. Иванова – М.: Издательство МГТУ им. Н.Э.Баумана, 2002 (40 экз.)

6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА

6.1 Оценочные средства, показатели и критерии оценивания компетенций

Индекс компетенции	Оценочное средство	Показатели оценивания	Критерии оценивания сформированности компетенций
ОПК-3 ПК-5	Собеседование	Низкий (неудовлетворительно)	Студент отвечает неправильно, нечетко и неубедительно, дает неверные формулировки, в ответе отсутствует какое-либо представление о вопросе
		Пороговый (удовлетворительно)	Студент отвечает неконкретно, слабо аргументировано и не убедительно, хотя и имеется какое-то представление о вопросе

ОПК-3 ПК-5	Разноуровневые задачи и задания	Базовый (хорошо)	Студент отвечает в целом правильно, но недостаточно полно, четко и убедительно
		Высокий (отлично)	Ставится, если продемонстрированы знание вопроса и самостоятельность мышления, ответ соответствует требованиям правильности, полноты и аргументированности.
		Низкий (неудовлетворительно)	Ответ студенту не зачитывается если: • Задание выполнено менее, чем на половину; • Студент обнаруживает незнание большей части соответствующего материала, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно излагает материал.
		Пороговый (удовлетворительно)	Задание выполнено более, чем на половину. Студент обнаруживает знание и понимание основных положений задания, но: • Излагает материал неполно и допускает неточности в определении понятий; • Не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; • Излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
		Базовый (хорошо)	Задание в основном выполнено. Ответы правильные, но: • В ответе допущены малозначительные ошибки и недостаточно полно раскрыто содержание вопроса; • Не приведены иллюстрирующие примеры, недостаточно четко выражено обобщающее мнение студента; • Допущено 1-2 недочета в последовательности и языковом оформлении излагаемого.
		Высокий (отлично)	Задание выполнено в максимальном объеме. Ответы полные и правильные. • Студент полно излагает материал, дает правильное определение основных понятий; • Обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры; • Излагает материал последовательно

			и правильно с точки зрения норм литературного языка.
--	--	--	--

6.2 Промежуточная аттестация студентов по дисциплине

Промежуточная аттестация является проверкой всех знаний, навыков и умений студентов, приобретённых в процессе изучения дисциплины. Формой промежуточной аттестации по дисциплине является зачёт.

Для оценивания результатов освоения дисциплины применяется следующие критерии оценивания.

Критерии оценивания устного ответа на зачете

Оценка «зачтено» выставляется студенту, если:

- разработанные информационные системы на основе Web-технологий реализуют требуемую в задании функциональность;
- студент самостоятельно выполнил все этапы решения индивидуальной задачи на ЭВМ;
- студент осмысленно и уверенно объясняет программный алгоритм решения задачи;
- студент при обсуждении программного алгоритма решения задания при необходимости может самостоятельно в присутствии преподавателя внести изменения в исходный код.

Оценка «не зачтено» выставляется студенту, если:

- значительная часть программы решения индивидуальной задачи на ЭВМ выполнена студентом не самостоятельно;
- разработанная программа не использует изучаемые механизмы и в исходном тексте программы не соблюдаются выбранные правила оформления исходного кода;
- при обсуждении алгоритма решения задачи студент не может самостоятельно в присутствии преподавателя внести изменения в исходный код.

6.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов освоения дисциплины

6.3.1 Пример дискуссии

Современные технологии разработки web-приложений

Вопросы к дискуссии:

- 1) Актуальность разработки веб-приложений.
- 2) Современные ЯП, используемые для веб-разработки.
- 3) Веб-сайт, веб-портал, веб-ресурс. Различие понятий.
- 4) Популярные веб-ресурсы в сети интернет.

6.3.2 Пример задания к лабораторной работе

Лабораторная работа №1 «Язык разметки гипертекстов HTML»

Вопросы допуска:

- 1) Определение HTML, Общее представление.
- 2) Структура HTML-документа.
- 3) Версии

Задание 1.

Создать используя язык разметки HTML 2-3 статические страницы сайта, объединенные общей тематикой, содержащие текст, изображения, списки и таблицы.

Примерные темы для задания:

- 1) Выращивание кактусов в домашних условиях.
- 2) Столицы Европы.
- 3) Столицы крупнейших городов Азии.
- 4) Подводная фотосъемка.
- 5) Аквариумные рыбы.
- 6) Блюда из ананасов.
- 7) Технологии вязания подарочных бантов.
- 8) Освоение космоса.
- 9) Обитатель подводных глубин.
- 10) Ремонт велосипедов.
- 11) Компьютерные игры.
- 12) Техники росписи деревянной посуды.
- 13) Интересные факты из жизни комедийных актеров.
- 14) Как устроен дизельный двигатель.
- 15) Любая выбранная тема...

6.3.3 Примерный перечень тем курсовой работы

1. Сайт кафе,
2. Сайт интернет-магазина,
3. Сайт школы,
4. Сайт справочной службы и т.п.

6.3.4 Примерные вопросы к зачету

1. Взаимодействие протоколов сети Интернет. Характеристика протоколов: ICMP, TCP, UDP, FTP, HTTP, DNS, POP, SMTP.
2. Основные элементы, определяющие тело HTML-документа.
3. Каскадные Таблицы Стилей. Спецификация CSS2.
4. Скриптовый язык программирования JavaScript. Способы функционального применения скриптов. Типы и структуры данных.
5. Технология программирования PHP.

7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ В ПРОЦЕССЕ ОБУЧЕНИЯ

Информационные технологии – обучение в электронной образовательной среде с целью расширения доступа к образовательным ресурсам, увеличения контактного взаимодействия с преподавателем, построения индивидуальных траекторий подготовки, объективного контроля и мониторинга знаний студентов.

В образовательном процессе по дисциплине используются следующие информационные технологии, являющиеся компонентами Электронной информационно-образовательной среды БГПУ:

- официальный сайт БГПУ;
- корпоративная сеть БГПУ;
- система электронного обучения ФГБОУ ВО «БГПУ»;

- электронные библиотечные системы;
- мультимедийное сопровождение лекций и практических занятий.

8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

При обучении лиц с ограниченными возможностями здоровья применяются адаптивные образовательные технологии в соответствии с условиями, изложенными в раздел «Особенности организации образовательного процесса по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья» основной образовательной программы (использование специальных учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь и т.п.) с учётом индивидуальных особенностей обучающихся.

9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ

9.1 Литература

1. Астапчук, В. А. Корпоративные информационные системы: требования при проектировании : учебное пособие для вузов / В. А. Астапчук, П. В. Терещенко. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2022. – 113 с. – (Высшее образование). – ISBN 978-5-534-08546-4. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/492141> (дата обращения: 10.10.2022).
2. Зудилова, Т.В. Web-программирование JavaScript [Электронный ресурс] : учебно-методическое пособие / Т.В. Зудилова, М.Л. Буркова. – Электрон. дан. – Спб. : НИУ ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики), 2012. – 68 с. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=43561.
5. Иванова, Г.С. Технология программирования. Учебник для вузов / Г.С. Иванова – М.: Издательство МГТУ им. Н.Э.Баумана, 2002 (40 экз.)
6. Тузовский, А. Ф. Проектирование и разработка web-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> (дата обращения: 10.10.2022).
7. 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> (дата обращения: 10.10.2022).

9.2 Базы данных и информационно-справочные системы

1. Федеральный портал «Российское образование» – Режим доступа: <http://www.edu.ru>
2. Информационная система «Единое окно доступа к образовательным ресурсам» – Режим доступа: <http://www.window.edu.ru>
3. Федеральный центр информационно-образовательных ресурсов – Режим доступа: <http://fcior.edu.ru>
4. Сайт Федеральной службы по интеллектуальной собственности, патентам и товарным знакам (Роспатента). – Режим доступа: <http://www.fips.ru/rospatent/index.htm>

9.3 Электронно-библиотечные ресурсы

1. ЭБС «Юрайт». – Режим доступа: <https://urait.ru>
2. Полпред (обзор СМИ). – Режим доступа: <https://polpred.com/news>

10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются аудитории, оснащённые учебной мебелью, аудиторной доской, компьютером с установленным лицензионным специализированным программным обеспечением, с выходом в электронно-библиотечную систему и электронную информационно-образовательную среду БГПУ, мультимедийными проекторами, экспозиционными экранами, учебно-наглядными пособиями (мультимедийные презентации).

Самостоятельная работа студентов организуется в аудиториях оснащенных компьютерной техникой с выходом в электронную информационно-образовательную среду вуза, в специализированных лабораториях по дисциплине, а также в залах доступа в локальную сеть БГПУ.

Лицензионное программное обеспечение: операционные системы семейства Windows, Linux; офисные программы Microsoft office, Libreoffice, OpenOffice; Adobe Photoshop, Matlab, DrWeb antivirus, Сервер системы контроля версий CVS; сервер БД Firebird, Средства разработки – PHP, phpMyAdmin, MySQL

Разработчик: Матевосян А.С. – начальник центра программно-технического обеспечения ФГБОУ ВО «БГПУ», ст. преподаватель кафедры информатики и методики.

11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ

Утверждение изменений и дополнений в РПД для реализации в 2025/2026 уч. г.

РПД обсуждена и одобрена для реализации в 2025/2026 уч. г. без изменений на заседании кафедры информатики и методики преподавания информатики (протокол №6 от 26.05.2025 г.).