

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ИНГУШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
Кафедра «Информационные системы и технологии»**

**СОГЛАСОВАНО**

Руководитель образовательной программы

\_\_\_\_\_/М.Х. Мальсагов  
«20» мая 2024г.

**УТВЕРЖДАЮ**

И.о. декана Физико-математического  
факультета

\_\_\_\_\_/Б.С.Кульбужев  
«23» мая 2024г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Б1.В.16 Web-программирование**

**Направление подготовки**

**09.03.02 Информационные системы и технологии**

**Направленность (профиль подготовки)**

**Перспективные информационные технологии**

**Квалификация выпускника**

**Бакалавр**

**Форма обучения**

**Очная, очно-заочная**

## ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### Перечень формируемых компетенций

#### Паспорт фонда оценочных средств

ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности

ОПК-4. Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью с использованием стандартов, норм и правил.

ПК-4. Способен проектировать и эксплуатировать ИС и их подсистемы.

ПК-5. Способен разрабатывать программные средства, модули и компоненты ИС.

№ п/п	Контролируемые разделы (темы) дисциплины	Код Контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	<b>Тема 1:</b> Введение в веб-программирование • Содержание: Основные концепции веб-программирования, клиент-серверная архитектура, основы HTTP протокола.	ОПК-2, ОПК-4, ПК-4, ПК-5	Тест
2.	<b>Тема 2:</b> Структура HTML документа, основные теги, стилизация с помощью CSS.	ОПК-2, ОПК-4, ПК-4, ПК-5	Тест
3.	<b>Тема 3:</b> Основы JavaScript • Содержание: Основы синтаксиса JavaScript, переменные, операторы, функции, объекты.	ОПК-2, ОПК-4, ПК-4, ПК-5	Коллоквиум
4.	<b>Тема 4:</b> Современные JavaScript-фреймворки (React, Angular, Vue) • Содержание: Обзор фреймворков, основные концепции, создание компонентов, маршрутизация.	ОПК-2, ОПК-4, ПК-4, ПК-5	Коллоквиум
5.	<b>Тема 5:</b> Серверная часть веб-приложений • Содержание: Введение в серверную разработку, обзор популярных фреймворков (Node.js,	ОПК-2, ОПК-4, ПК-4, ПК-5	Тест, коллоквиум

	Django, Ruby on Rails), работа с запросами и ответами.		
6.	<b>Тема 6</b> Работа с базами данных • Содержание: Основы баз данных, SQL запросы, подключение к базам данных из веб-приложений.	ОПК-2, ОПК-4, ПК-4, ПК-5	Коллоквиум
7	<b>Тема 7</b> Интеграция веб-приложений с внешними API • Содержание: Работа с внешними сервисами и API, отправка и получение данных, аутентификация и авторизация	ОПК-2, ОПК-4, ПК-4, ПК-5	
8	<b>Тема 8</b> Тестирование и отладка веб-приложений • Содержание: Основы тестирования веб-приложений, различные виды тестов, отладка и исправление ошибок.	ОПК-2, ОПК-4, ПК-4, ПК-5	
9	<b>Тема 9</b> Подготовка и публикация веб-приложений • Содержание: Подготовка веб-приложений к развертыванию на сервере, выбор хостинга, настройка домена.	ОПК-2, ОПК-4, ПК-4, ПК-5	
10	<b>Тема 10</b> Оптимизация и безопасность веб-приложений • Содержание: Методы оптимизации производительности веб-приложений, защита от уязвимостей и атак.	ОПК-2, ОПК-4, ПК-4, ПК-5	

### Типовой тест промежуточной аттестации

- Что такое HTML?
  - а) Гипертекстовый маркап язык
  - б) Гипертекстовый программный язык
  - в) Гипертекстовый стилевой язык
  - Ответ: а) Гипертекстовый маркап язык (+)**
- Какие основные теги используются для создания структуры HTML-документа?
  - а) <html>, <head>, <body>
  - б) <header>, <main>, <footer>
  - в) <section>, <article>, <nav>
  - Ответ: а) <html>, <head>, <body> (+)**
- Какие файлы необходимы для создания базового веб-сайта с использованием HTML и CSS?
  - а) Только файл HTML
  - б) Только файл CSS

- в) Файлы HTML и CSS
  - **Ответ: в) Файлы HTML и CSS (+)**
4. Что такое CSS?
- а) Каскадные таблицы стилей
  - б) Каскадные таблицы скриптов
  - в) Каскадные таблицы селекторов
  - **Ответ: а) Каскадные таблицы стилей (+)**
5. Какой селектор CSS применяет стили ко всем элементам на веб-странице?
- а) \*
  - б) #
  - в) .
  - **Ответ: а) \* (+)**
6. Какие типы единиц измерения используются в CSS для задания размеров элементов?
- а) Пиксели (px), проценты (%), em, rem
  - б) Целые числа (int), дробные числа (float), строки (str)
  - в) Паскалы (px), Килограммы (kg), Метры (m)
  - **Ответ: а) Пиксели (px), проценты (%), em, rem (+)**
7. Какие основные принципы оптимизации производительности веб-приложений?
- а) Минимизация загрузки ресурсов, оптимизация работы JavaScript, использование кеширования
  - б) Использование максимального числа картинок, большое количество анимаций, регулярное обновление содержимого
  - в) Использование максимального числа шрифтов, большое количество фоновых изображений, регулярное использование видео
  - **Ответ: а) Минимизация загрузки ресурсов, оптимизация работы JavaScript, использование кеширования (+)**
8. Какой метод HTTP используется для получения данных с сервера?
- а) GET
  - б) POST
  - в) UPDATE
  - **Ответ: а) GET (+)**
9. Какой фреймворк JavaScript чаще всего используется для разработки одностраничных приложений (SPA)?
- а) React
  - б) Vue.js
  - в) Angular
  - **Ответ: а) React (+)**
10. Каким образом можно интегрировать CSS стили в HTML-документ?
- а) Встроить стили прямо в HTML с помощью атрибута style
  - б) Использовать тег <link> для подключения внешнего файла стилей
  - в) Использовать тег <style> для вставки CSS правил внутри HTML документа
  - **Ответ: б) Использовать тег <link> для подключения внешнего файла стилей (+)**
11. Что такое адаптивный дизайн веб-приложений?
- а) Тип дизайна, который автоматически адаптируется под разные разрешения экранов устройств
  - б) Дизайн, который использует только адаптивные элементы
  - в) Стил кодирования, который динамически меняется в зависимости от устройства

- **Ответ: а) Тип дизайна, который автоматически адаптируется под разные разрешения экранов устройств (+)**
- 12. Какие технологии используются для обеспечения безопасности веб-приложений?
  - а) HTTPS, CORS, CSRF
  - б) HTML, CSS, JavaScript
  - в) SQL, XML, JSON
  - **Ответ: а) HTTPS, CORS, CSRF (+)**
- 13. Каким образом можно организовать передачу данных между сервером и клиентом в веб-приложениях?
  - а) Используя AJAX запросы
  - б) Через параметры URL
  - в) Используя только метод POST
  - **Ответ: а) Используя AJAX запросы (+)**
- 14. Какой HTTP метод используется для отправки данных на сервер?
  - а) POST
  - б) GET
  - в) PUT
  - **Ответ: а) POST (+)**
- 15. Какие основные принципы RESTful архитектуры?
  - а) Отсутствие состояния, клиент-серверное взаимодействие, кэширование
  - б) Модель запрос-ответ, универсальность интерфейса, слой ограничений
  - в) Одноранговая архитектура, масштабируемость, управление сетевым узлом
  - **\*\*Ответ: б) Модель запрос-ответ,**

☐ Какие основные компоненты включает в себя архитектура MVC (Model-View-Controller)?

- а) Модель, Вид, Страница
- б) Модель, Вид, Контроллер
- в) Модуль, Веб-сервер, Клиент
- **Ответ: б) Модель, Вид, Контроллер (+)**

☐ Какие принципы следует соблюдать при проектировании пользовательского интерфейса в веб-приложениях?

- а) Принцип единственной ответственности, принцип открытости/закрытости, принцип подстановки Лисков
- б) Принцип разделения интерфейса и реализации, принцип единства, принцип простоты
- в) Принцип гетерогенности, принцип полиморфизма, принцип наследования
- **Ответ: б) Принцип разделения интерфейса и реализации, принцип единства, принцип простоты (+)**

☐ Что такое аутентификация и авторизация в веб-приложениях?

- а) Аутентификация - проверка подлинности пользователя, авторизация - предоставление доступа к определенным ресурсам
- б) Аутентификация - предоставление доступа к определенным ресурсам, авторизация - проверка подлинности пользователя
- в) Аутентификация - передача учетных данных, авторизация - создание учетной записи

- **Ответ: а) Аутентификация - проверка подлинности пользователя, авторизация - предоставление доступа к определенным ресурсам (+)**
- ☐ Каким образом можно организовать маршрутизацию в одностраничных веб-приложениях?
- а) Используя серверную маршрутизацию
  - б) Используя клиентскую маршрутизацию
  - в) Используя API запросы
  - **Ответ: б) Используя клиентскую маршрутизацию (+)**
- ☐ Какие инструменты используются для отладки веб-приложений?
- а) Браузерные инструменты разработчика, логгирование данных на сервере, специализированные IDE
  - б) Препроцессоры, сборщики, линтеры
  - в) Тестовые фреймворки, системы контроля версий, CI/CD инструменты
  - **Ответ: а) Браузерные инструменты разработчика, логгирование данных на сервере, специализированные IDE (+)**
- ☐ Какие принципы обеспечения безопасности информации соблюдаются при разработке веб-приложений?
- а) Принцип наименьших привилегий, принцип разделения ответственности, принцип защиты по умолчанию
  - б) Принцип инкапсуляции, принцип аутентификации, принцип прозрачности
  - в) Принцип маршрутизации, принцип шифрования, принцип балансировки нагрузки
  - **Ответ: а) Принцип наименьших привилегий, принцип разделения ответственности, принцип защиты по умолчанию (+)**

### Типовой вариант задания на контрольную работу

Разработайте простую веб-страницу, которая включает в себя следующие элементы:

- Заголовок страницы
- Текстовый блок с описанием вашего любимого хобби или увлечения
- Изображение, связанное с вашим хобби или увлечением
- Форму для отправки обратной связи с полями "Имя", "Email" и "Сообщение"

#### Инструкции:

- Используйте HTML и CSS для создания макета и стилей страницы.
- Добавьте JavaScript для валидации формы обратной связи перед отправкой данных на сервер.
- Обязательно протестируйте работу страницы в различных браузерах.

### Экзаменационные вопросы по дисциплине

1. Что такое веб-программирование и какова его роль в разработке современных веб-приложений?
2. Какие основные компоненты включает в себя веб-технология?
3. Чем отличаются клиентская и серверная части веб-приложений?
4. Какие протоколы используются в веб-программировании и для чего они предназначены?
5. Что такое HTML и CSS? Какие задачи они решают в веб-разработке?
6. Какие основные принципы JavaScript? В каких сферах его применения в веб-разработке?
7. Какие существуют методы внедрения JavaScript-кода на веб-страницу?
8. Что такое JavaScript-фреймворк? Приведите примеры популярных фреймворков.
9. В чем различие между React, Angular и Vue.js?
10. Какие шаблонные движки используются в серверной части веб-приложений?
11. Что такое RESTful API? Какие принципы оно соблюдает?
12. Каким образом веб-приложения могут взаимодействовать с базами данных?
13. Какие инструменты и технологии используются для обеспечения безопасности веб-приложений?
14. Что такое сессии и куки в контексте веб-разработки?
15. Каким образом можно оптимизировать производительность веб-приложений?
16. Какие методы тестирования применяются в веб-разработке?
17. Какие основные этапы развертывания веб-приложений на сервере?
18. Какие принципы следует соблюдать при проектировании пользовательского интерфейса в веб-приложениях?
19. Какие принципы составления адаптивного дизайна в веб-приложениях?
20. Какие меры можно принять для обеспечения безопасности веб-приложений от атак на стороне клиента?
21. Что такое аутентификация и авторизация в веб-приложениях? Какие методы их реализации существуют?
22. Какие инструменты можно использовать для управления версиями кода в веб-разработке?
23. Какие основные шаблоны проектирования применяются в веб-разработке?
24. Какие технологии используются для оптимизации загрузки ресурсов в веб-приложениях?
25. Какие принципы и методы использования асинхронного программирования в JavaScript?
26. Каким образом веб-приложения могут поддерживать многоязычность?
27. Какие методы существуют для обеспечения кросс-браузерной совместимости в веб-разработке?
28. Каким образом можно улучшить SEO-оптимизацию веб-приложений?
29. Какие инструменты используются для профилирования и анализа производительности веб-приложений?
30. Какие технологии используются для реализации реактивного программирования в веб-разработке?

## Типовая лабораторная работа по дисциплине

### Лабораторная работа №1: Создание простой HTML-страницы

**Цель работы:** Научиться создавать базовую HTML-страницу с использованием основных тегов и структур.

#### Задачи:

1. Создание HTML-документа.
2. Добавление заголовка и основного контента.
3. Вставка изображений.
4. Создание ссылок и списка.

#### Шаги выполнения:

1. Откройте любой текстовый редактор (например, Notepad++) или интегрированную среду разработки (IDE), такую как Visual Studio Code.
2. Создайте новый файл с расширением ".html" и назовите его "index.html".
3. Введите следующий код:

```
html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Моя первая HTML-страница</title>
</head>
<body>
  <h1>Добро пожаловать на мою первую HTML-страницу!</h1>
  <p>Это простая страница, созданная с использованием HTML.</p>
  
  <h2>Ссылки</h2>
  <ul>
    <li><a href="https://www.example.com">Пример ссылки</a></li>
    <li><a href="#bottom">Прокрутить вниз</a></li>
  </ul>
  <h2 id="bottom">Конец страницы</h2>
</body>
</html>
```

4. Создайте изображение с названием "example.jpg" и сохраните его в той же папке, что и ваш HTML-файл.
5. Откройте HTML-файл в вашем веб-браузере для просмотра результатов.

#### Контрольные вопросы:

1. Какие основные элементы включает в себя HTML-документ?
2. Какие атрибуты <meta> тега используются для установки кодировки символов и масштабирования страницы?
3. Как создать ссылку на другую веб-страницу?
4. Как создать якорь (anchor) для прокрутки к определенному разделу страницы?