

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Крюков Вадим Николаевич

Должность: Проректор по образовательной деятельности и инновационной политике

Дата подписания: 17.06.2026 18:11:51

Уникальный программный ключ:

1b0adb7fd710f6a0705d90c58682bd0c5f2f25b2

**Министерство науки и высшего образования РФ**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Заполярный государственный университет им. Н. М. Федоровского»**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**по дисциплине**  
**WEB-программирование**

Направление подготовки 09.03.02 «Информационные системы и технологии»  
Профиль «Проектирование и реализация информационных систем и технологий»  
Уровень образования: бакалавриат

Кафедра «Информационные системы и технологии»

Разработчик ФОС:  
кандидат экономических наук, доцент, Беляев И.С.  
\_\_\_\_\_ Беляев И.С.

Оценочные материалы по дисциплине рассмотрены и одобрены на заседании кафедры, протокол от 10.04.2026г. № 5.

Заведующий кафедрой \_\_\_\_\_ к.э.н., Беляев И.С.

Фонд оценочных средств по дисциплине WEB-программирование для текущей/промежуточной аттестации разработан в соответствии с Федеральным государственным образовательным стандартом высшего образования по специальности / направлению подготовки 09.03.02 Информационные системы и технологии на основе Рабочей программы дисциплины WEB-программирование, утвержденной решением ученого совета от г., Положения о формировании Фонда оценочных средств по дисциплине (ФОС), Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся ЗГУ, Положения о государственной итоговой аттестации (ГИА) выпускников по образовательным программам высшего образования в ЗГУ им. Н.М. Федоровского.

**1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами образовательной программы**

Таблица 1. Компетенции и индикаторы их достижения

Код и наименование компетенции	Индикаторы достижения
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Определяет круг задач в рамках поставленной цели, определяет связи между ними
	УК-2.2 Предлагает способы решения поставленных задач и ожидаемые результаты; оценивает предложенные способы с точки зрения соответствия цели проекта
	УК-2.3 Планирует реализацию задач в зоне своей ответственности с учетом имеющихся ресурсов и ограничений, действующих правовых норм
ПК-2 Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	ПК-2.1 Демонстрирует навыки управления процессами формирования и проверки требований к разрабатываемому программному обеспечению с учетом действующих правовых норм и законодательных актов в области разработки программного обеспечения
	ПК-2.2 Использует навыки планирования процесса разработки программного продукта

ПК-2 Способность разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	ПК-2.3 Составляет планы процесса разработки программного продукта
ПК-3 Способность обеспечения эффективной работы баз данных, включая развертывание, сопровождение, оптимизация функционирования баз данных, являющихся частью различных информационных систем	ПК-3.1 Выбирает и применяет практики и методологии управления разработкой ПО
	ПК-3.2 Комбинирует навыки выбора средств создания и ведения репозитория, учета задач, сборки и непрерывной интеграции базы знаний
	ПК-3.3 Использует современные информационные, компьютерные и сетевые технологий для поиска, хранения и анализа информации из различных источников и баз данных
ПК-6 Способность выполнять проектирование и графический дизайн интерактивных пользовательских интерфейсов	ПК-6.1 Демонстрирует знания к требованиям, предъявляемым к графическому дизайну интерфейса, тенденциям в графическом дизайне
	ПК-6.2 Выбирает оптимальные характеристики элементов интерфейса, создавать визуальный стиль интерфейса

Таблица 2. Паспорт фонда оценочных средств

№п/п	Контролируемые разделы(темы) дисциплины	Кодрезультатаобучения по дисциплине/ модулю	Оценочные средства текущей		Оценочные средства промежуточной	
			Наименование	Форма	Наименование	Форма
<b>5 семестр</b>						

**2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы.**

**2.1. Задания для текущего контроля успеваемости**

1. Основные компоненты технологии Web. Понятие гипертекста, URI, URL, HTTP.
2. Структура HTML-документа. Теги `<html>`, `<head>`, `<body>`. Роль метатегов.
3. Блочная и строчная верстка. Семантические теги HTML5.

4. Теги форматирования текста (<h1>-<h6>, <p>, <strong>, <em>, <hr>).
5. Создание списков: упорядоченных, неупорядоченных, определений.
6. Встраивание изображений и видео. Атрибуты src, alt, width, height.
7. Гиперссылки. Атрибуты href, target, rel. Якоря.
8. Формы в HTML: элементы <input>, <textarea>, <select>, <button>. Атрибуты name, value, type.
9. Основы CSS: селекторы, свойства, значения. Внешние, внутренние, инлайновые стили.
10. Бокс-модель CSS: margin, border, padding, content. Блочные и строчные элементы.
11. Позиционирование: static, relative, absolute, fixed, sticky.
12. Flexbox и Grid: основные свойства, выравнивание, расположение элементов.
13. Адаптивная верстка: медиа-запросы, viewport, резиновые единицы.
14. Основы JavaScript: типы данных, переменные, операторы, выражения.
15. Условные конструкции (if, switch) и циклы (for, while, do...while).
16. Функции в JavaScript: объявление, вызов, параметры, возврат значений.
17. Объекты и массивы. Методы объектов и массивов (push, pop, map, filter, reduce).
18. DOM: получение элементов (getElementById, querySelector, getElementsByClassName), изменение содержимого, стилей, атрибутов.
19. Обработчики событий: onclick, onmouseover, addEventListener. Всплытие и погружение.
20. Создание калькулятора: HTML-верстка, CSS-стилизация, JS-логика (сложение, вычитание, деление, умножение).
21. Работа с Node.js: установка, npm, создание проекта, подключение модулей.
22. Знакомство с npm: package.json, npm init, npm install, npm start.
23. Использование mock-объектов для генерации данных (например, товаров, пользователей).
24. Прототипное наследование в JavaScript. Конструкторы и классы (ES6).
25. Замыкания, контекст выполнения (this), стрелочные функции.
26. Модули в JavaScript: CommonJS, ES6 modules. Импорт и экспорт.
27. Обработка ошибок: try...catch, throw, finally.
28. XMLHttpRequest: отправка GET/POST запросов, обработка ответа, обработка ошибок.
29. Fetch API: базовое использование, обработка ответа, обработка ошибок, async/await.
30. Замена колбэков на промисы и async/await. Цепочки промисов.
31. Подключение к стороннему API (VK, GitHub, JSONPlaceholder). Обход CORS.
32. Понятие клиент-серверной модели. DNS, браузерный рендеринг, HTTP-запросы.
33. Cookie, localStorage, sessionStorage. Хранение данных на стороне клиента.
34. CORS: причины возникновения, способы обхода (проху, CORS-заголовки).
35. Ajax: принцип работы, отличие от fetch, использование jQuery.ajax (опционально).
36. Node.js: сервер на Express, маршруты (GET, POST, PUT, DELETE), middleware.
37. Создание REST API: CRUD-операции, валидация данных, обработка ошибок.



38. Тестирование API: использование supertest, mocha, chai, покрытие кода.
39. Подключение frontend к backend: отправка запросов, обновление интерфейса.
40. Структура проекта: MVC, разделение на контроллеры, модели, маршруты, middleware.
41. Обновление данных через polling: setInterval, обновление UI без перезагрузки.
42. WebSocket: принцип работы, подключение, отправка сообщений.
43. Развертывание приложения: .env, pm2, nodemon, логирование, Docker (опционально).
44. Безопасность: защита от XSS, CSRF, валидация входных данных.
45. Документация API: Swagger/OpenAPI, Postman, README.md.

## 2.2 Темы письменных работ (эссе, рефераты, курсовые работы и др.)

### Эссе

1. Роль веб-технологий в современном мире.
2. Перспективы развития веб-программирования в ближайшие 10 лет.
3. Влияние адаптивного дизайна на пользовательский опыт.
4. Безопасность данных в веб-приложениях: вызовы и решения.
5. Как искусственный интеллект меняет подходы к разработке веб-сайтов.

### Рефераты

1. История развития веб-программирования: от HTML до современных фреймворков.
2. Обзор популярных библиотек и фреймворков для фронтенда (React, Vue, Angular).
3. Принципы построения REST API для современных веб-приложений.
4. Сравнительный анализ систем управления контентом (CMS): WordPress, Joomla, Drupal.
5. Методы оптимизации производительности веб-сайтов.

1. Что такое Интернет. Хронология создания Интернет (события, даты). Интернет протоколы. Протоколы прикладного уровня: HTTP, SMTP, POP3, IMAP, FTP, DNS, Telnet.

2. Что такое Интернет. Хронология создания Интернет (события, даты). Интернет протоколы. Протоколы передачи данных: назначение протоколов TCP и IP, IP пакеты, схема передачи данных по стеку протоколов TCP/IP, IP адрес, доменное имя, DNS сервер.

3. Что такое World Wide Web. Основные идеи WWW. Что такое Веб-браузер. Как работает Веб-браузер. HTTP, HTML. Наиболее популярные Веб-браузеры (Microsoft Internet Explorer, Mozilla Firefox, Opera, Netscape), их возможности.

4. Что такое электронная почта. Как работает электронная почта. Структура заголовка электронного письма. Почтовые клиенты. Веб-интерфейсы. Протоколы POP3, IMAP, SMTP.

5. Наиболее популярные почтовые клиенты (Microsoft Outlook, Mozilla Thunderbird, The Bat), их возможности (установка, настройки, фильтры, адресная книга,

поиск и пр.). Основные параметры настройки почтовых фильтров. Сравнение способов работы с электронной почтой (почтовые клиенты и Веб-интерфейсы).

6. Спам. Виды спама. Способы борьбы со спамом. Возможности почтовых клиентов по борьбе со спамом.

7. Определение понятия Веб-сайт. Статические Веб-сайты и динамические Веб-сайты (определение, назначение, свойства, примеры). Инструментальные средства для создания Веб-сайтов.

8. Веб-порталы (определение, назначение, свойства, типы, примеры порталов). Основные отличия сайта от портала. Примеры порталов.

9. История развития поисковых систем (даты, примеры, возможности). Общие принципы работы поискового движка. Правила формирования поисковых запросов. Релевантность поиска.

10. Типы поисковых систем (полнотекстовый поиск, поиск по каталогу, мета-поиск). Примеры и возможности поисковых систем разного типа. Возможности современных поисковых систем на примере Google и Яндекс. Дополнительные сервисы поисковых систем.

11. Общая идея работы WYSIWYG редакторов. Акронимом какого словосочетания является термин WYSIWYG. Основные преимущества и недостатки WYSIWYG редакторов. Назначение и основные возможности Microsoft FrontPage.

12. Основные характеристики изображений. Форматы изображений для Веб. Характерные особенности форматов JPEG, GIF, PNG. Критерии выбора наиболее подходящего формата изображения для публикации на Веб. Графические редакторы.

13. Платежные системы интернет. Назначение, способы применения, достоинства, недостатки. Оплата кредитными и дебетовыми картами. Особенности, достоинства и недостатки.

14. Платежные системы интернет. Назначение, способы применения, достоинства, недостатки. Зарубежные платежные системы на примере PayPal и E-Gold. Особенности этих платежных систем. Способы применения платежных систем для приема он-лайн платежей на сайтах.

15. Платежные системы интернет. Назначение, способы применения, достоинства, недостатки. Российские платежные системы на примере WebMoney и Яндекс-Деньги. Особенности этих платежных систем. Способы применения платежных систем для приема он-лайн платежей на сайтах.

16. Структура языка HTML. Общий синтаксис. Теги, атрибуты. Основные теги, используемые в заголовке. Основные теги используемые для оформления текста. Атрибуты.

17. Структура языка HTML. Общий синтаксис. Теги, атрибуты. Основные теги, используемые в заголовке. Списки. Основные теги для работы со списками. Атрибуты.

18. Структура языка HTML. Общий синтаксис. Теги, атрибуты. Основные теги, используемые в заголовке. Таблицы. Основные теги для работы с таблицами. Атрибуты.

19. Структура языка HTML. Общий синтаксис. Теги, атрибуты. Основные теги, используемые в заголовке. Ссылки. Теги для работы со ссылками. Особенности публикации HTML документов в Интернет. Абсолютные и относительные адреса.

20. Назначение таблиц стилей. Способы подключения таблиц к HTML документам. Возможности таблиц стилей. Единицы измерения. Изменение оформления блоков и абзацев с помощью таблиц стилей.

21. Назначение таблиц стилей. Способы подключения таблиц к HTML документам. Возможности таблиц стилей. Единицы измерения. Изменение оформления списков с помощью таблиц стилей.

22. Назначение таблиц стилей. Способы подключения таблиц к HTML документам. Возможности таблиц стилей. Единицы измерения. Изменение оформления



текста с помощью таблиц стилей.

23. Общие сведения о JavaScript. История языка. Назначение. Примеры задач, решаемых с помощью JavaScript. Объектная модель HTML документа. Операторы JavaScript. Примеры использования операторов. Переменные в JavaScript. Объявление переменных, арифметические операции.

24. Общие сведения о JavaScript. История языка. Назначение. Примеры задач, решаемых с помощью JavaScript. Объектная модель HTML документа. Объявление функций в JavaScript. Параметры функций, возвращаемые значения. Операторы условий в JavaScript. Примеры использования.

25. Общие сведения о JavaScript. История языка. Назначение. Примеры задач, решаемых с помощью JavaScript. Объектная модель HTML документа. Операторы циклов в JavaScript. Примеры использования. Включение JavaScript в HTML код. Способы, специфика методов включения.

26. Протокол HTTP. Схема взаимодействия между клиентом и сервером. Общие сведения о PHP. История языка. Назначение. Примеры задач, решаемых с помощью PHP.

27. Основной синтаксис языка PHP Основные операторы PHP. Переменные в PHP. Объявление переменных, типы переменных.

28. Основной синтаксис языка PHP Основные операторы PHP. Объявление функций в PHP. Параметры функций, возвращаемые значения.

29. Основной синтаксис языка PHP Основные операторы PHP. Операторы условий в PHP. Примеры использования.

30. Основной синтаксис языка PHP Основные операторы PHP. Операторы циклов в PHP. Примеры использования.

#### 1. Текущий контроль знаний:

- Тесты на знание основ HTML, CSS и JavaScript.
- Практические задания по верстке веб-страниц.
- Написание скриптов для обработки данных на клиентской стороне.
- Лабораторные работы по созданию форм и их обработке.

#### 2. Промежуточная аттестация:

- Выполнение контрольной работы (например, создание простой веб-страницы с использованием HTML, CSS и JavaScript).
- Рефераты или эссе на заданные темы.
- Мини-проект (например, разработка адаптивного сайта или веб-приложения).

#### 3. Итоговый контроль:

- Защита курсового проекта (разработка полноценного веб-приложения или сайта).
- Экзамен в виде теста или выполнения практического задания.
- Оценка портфолио выполненных лабораторных и практических работ.

#### 4. Дополнительные виды оценочных средств:

- Устное собеседование по ключевым темам курса.
- Анализ кода и исправление ошибок в предоставленном проекте.
- Выполнение заданий на оптимизацию производительности сайта или приложения.