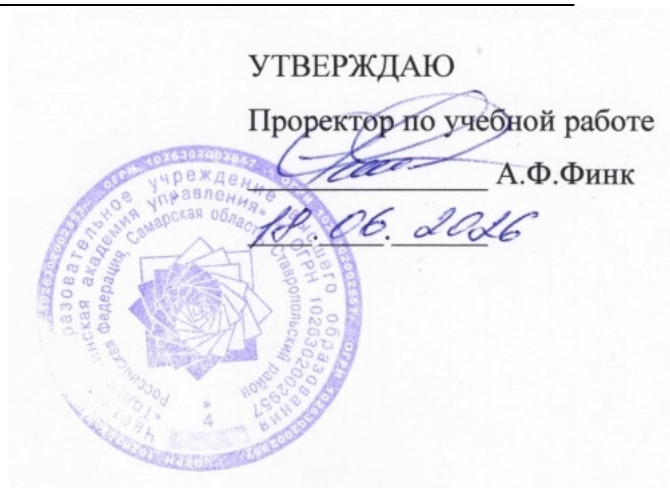


Документ подписан простой электронной подписью
 Информация о владельце:
 ФИО: Финк Анастасия Федоровна
 Должность: Проректор по учебной работе
 Дата подписания: 30.06.2026 14:32:09
 Уникальный программный ключ:
 2431bd5130e74d20a9fc74baab365dd497e3afa3

ЧОУ ВО «Тольяттинская академия управления»

Кафедра

прикладной информатики и высшей математики



Б1.В.06

РАБОЧАЯ ПРОГРАММА

Учебная дисциплина	Фреймворки front-end разработчика
По направлению подготовки	09.03.03 «Прикладная информатика»
Профиль (программа бакалавриата)	«Дизайн и разработка цифровых программных продуктов»
Форма обучения	Заочная

Программа дисциплины рассмотрена (актуализирована) и утверждена на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

1. ОБЩАЯ ИНФОРМАЦИЯ

Рабочая программа дисциплины «Фреймворки front-end разработчика» составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденного приказом Министерства образования и науки Российской Федерации 19.09.2017 №922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021, 19.07.2022, 27.02.2023), и учебного плана направления подготовки 09.03.03 «Прикладная информатика», профиль «Дизайн и разработка цифровых программных продуктов» (программа бакалавриата).

Трудоемкость дисциплины: 3 ЗЕТ / 108 академических часов, в том числе: 16 часов контактной работы и 83 часа самостоятельной работы обучающихся.

Распределение часов дисциплины по семестрам и видам занятий (по учебному плану)

Вид учебной работы	Всего по учебному плану	Количество часов													
		Курсы, сессии													
		1 курс			2 курс			3 курс			4 курс			5 курс	
		Уст.	Зим.	Лет.	Уст.	Зим.	Лет.	Уст.	Зим.	Лет.	Уст.	Зим.	Лет.	Уст.	Зим.
Контактная работа (всего)	16							2	14						
в том числе:															
лекции	6							2	4						
практические занятия	10								10						
контроль самостоятельной работы (КСР)															
Самостоятельная работа	83								83						
Виды промежуточной аттестации	Экзамен 9								Экзамен 9						
ОБЩАЯ трудоемкость дисциплины	Часы:	108						2	106						
	Зач. ед.:	3													

2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Цель дисциплины – изучение современных фреймворков и библиотек front-end разработчика для создания внешней (англ. front) стороны веб-сайтов в части разработки пользовательского интерфейса, для правильной работы и отражения web-ресурсов в существующих браузерах и на различных электронных гаджетах. Благодаря использованию фреймворков и библиотек для создания web-приложений становятся динамичными, в них используются различные интерактивные эффекты: плавные переходы, появляющиеся меню, всплывающие окна, анимационные элементы и др.

Задачи дисциплины:

- получение теоретических знаний о языках программирования, которые используют в своей работе front-end разработчика;
- получение теоретических знаний о современных фреймворках и библиотеках front-end разработчика;
- приобретение умения использования современных фреймворков и библиотек front-end разработчика;
- получение практических навыков разработки web-ресурсов и их компонентов с использованием современных фреймворков и библиотек front-end разработчика.

3. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПЛАНЕ

Учебная дисциплина «Фреймворки front-end разработчика» относится к дисциплинам части Блока 1. Дисциплины (модули), формируемой участниками образовательных отношений. Изучение данной дисциплины базируется на материале, изученном в дисциплине «Система управления контентом (CMS)», «Fullstack-разработка web-приложений», «Разработка облачных приложений и сервисов. Знания, умения и навыки, приобретенные в результате изучения дисциплины, будут востребованы при освоении дисциплин «UX-дизайн интерфейсов», «Технологии виртуальной и дополненной реальности», «SEO-продвижение и сопровождение программных продуктов», «Подходы к трафику: таргетирование и контекст» и написании выпускной квалификационной работы.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ ДЛЯ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ

Результаты освоения образовательной программы (компетенции обучающихся) устанавливаются в соответствии с федеральным государственным образовательным стандартом по направлению подготовки и профессиональными стандартами, соответствующими профессиональной деятельности выпускников, а также на основе анализа требований работодателей, предъявляемых к выпускникам. Планируемые результаты обучения по дисциплине (знания, умения, навыки) обеспечивают достижение результатов освоения образовательной программы.

Шифр и название компетенции	Индикаторы компетенции	Планируемые результаты обучения по дисциплине
ПК-2. Способен проектировать цифровой программный продукт	ПК-2.3. Проектирует пользовательский интерфейс с учетом тенденций развития цифровых технологий	Знать: - основы прототипирования и разработки дизайна интерфейса web-приложений; - этапы и инструменты проектирования web-приложений и их компонентов; Уметь: - продумать и визуализировать логику web-приложения, его пользовательский интерфейс и отдельные компоненты;

		- создавать графические решения, максимально соответствующие функциональности web-приложения Владеть: - инструментами разработки пользовательских интерфейсов web-приложений.
ПК-3. Способен разрабатывать цифровой программный продукт	ПК-3.1. Осуществляет разработку программного кода и компонентов цифрового программного продукта с помощью специализированных цифровых платформ для индивидуальной и совместной разработки	Знать: - основы языка JavaScript; - инструменты разработки Frontend приложений; - основы разработки web-приложений на Vue.js, Angular, React; Уметь: - разрабатывать web-приложения с помощью фреймворков и библиотек JavaScript (Vue.js, Angular, React); Владеть: - навыками построения приложений с Redux.js; - навыками применения компонентного подхода при разработке web-приложений; - навыками разработки одностраничных приложений (SPA).
	ПК-3.2. Обеспечивает интеграцию программных модулей и компонентов цифрового программного продукта на основе облачных и отраслевых ИТ-решений	Знать: - принципы и назначение фреймворков и библиотек (Vue.js, Angular, React); - приемы написания минималистического кода с гибкой архитектурой. Уметь: - обеспечивать интеграцию программных модулей и компонентов web-приложений. Владеть: - методами и инструментальными средствами интеграции программных модулей и компонентов web-приложений.
ПК-5. Способен обеспечивать качество работы цифрового программного продукта	ПК-5.3. Осуществляет работы по интеграционному тестированию цифрового программного продукта с внешним окружением	Знать: - современные инструменты и практики тестирования JavaScript-кода; - процесс интеграции программного обеспечения и интеграцию разработки API Уметь: - проводить интеграционное тестирование кода с помощью различных инструментов. Владеть: - навыками создания среды для запуска тестов.

Этап изучения: 3 курс, установочная и зимняя сессия

Этап изучения: 3 курс, установочная и зимняя сессия

6. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Раздел 1. Vue.js

Тема 1. Основы Vue.js

Введение в JavaScript: основы, диалог с пользователем и вычисления, массивы, обработка событий.

Инструменты разработки Frontend приложений. Синтаксис шаблонов. Реактивность вычисляемые свойства, отслеживание. Работа с DOM событиями, атрибутами, формами. Хуки жизненного цикла. Компонентный подход. Создание компонентов. Параметры и события компонентов. Слоты. Однофайловые компоненты (SFC) и инкапсуляция стилей. Создание и разработка приложений с @vue/cli. Одностраничные приложения (SPA). Маршрутизация на клиенте, vue-router. Unit-тестирование Vue.js приложения с Jest и vue-test-utils

Тема 2. Углубление во Vue.js и разработку компонентов

Компоненты-обёртки. Компоненты-формы и однонаправленный поток данных. Реактивность вне компонентов и реализация реактивности. Основы рендеринга, Virtual DOM, render-функции, JSX. Ограничения Vue.js, его зона ответственности и работа с другими библиотеками. Scoped-slots, Renderless компоненты и другие подходы к разработке компонентов. Composition API

Раздел 2. Angular

Тема 3. Основные принципы и архитектура

Назначение фреймворка Angular. Модульная архитектура Angular. Атрибутивные и структурные директивы Angular. Структура приложения и практические рекомендации по её формированию. Синтаксис привязки данных и шаблоны. Формы и валидация данных (реактивные формы и валидаторы). Проекция контента. Стратегия обнаружения изменений onPush.

Тема 4. Компоненты и шаблоны

Компоненты. Модули. Загрузка приложения. Стили и шаблоны компонента. Привязка данных. Работа с компонентами. Взаимодействие между компонентами. Привязка к событиям дочернего компонента. Жизненный цикл компонента. Шаблонные переменные, ViewChild, ContentChild. Взаимодействие между модулями. ngClass и ngStyle. Создание атрибутивных директив. Взаимодействие с пользователем, HostListener и HostBinding. Получение параметров в директивах. Структурные директивы ngIf, ngFor, ngSwitch. Создание структурных директив.

Тема 5. Реактивность. Маршрутизация. Формы

HttpClient и отправка запросов. Объект Observable и библиотека RxJS. Обработка ошибок. Отправка данных в запросе. POST-запросы. Определение маршрутов. Создание

ссылок. Параметры маршрута. Параметры строки запроса. Программная навигация. Дочерние маршруты. Guards. Модуль FormsModule и директива NgModel. Получение и изменение модели. Состояние модели и валидация. Директива NgForm. Reactive Forms

Раздел 3. React

Тема 6. Знакомство с экосистемой React. Построение приложений с Redux.js.

Создание простых компонентов. Декларативный подход. Отличие React.js от других популярных фреймворков. Знакомство с экосистемой React. Использование сторонних компонентов и работа с формами. Особенности Redux.js: функциональный подход, Redux dev tools. Store как иммутабельный объект: Redux + Immutable.js. Синхронный поток с Redux. React-redux для связи компонентов с логикой. Мемоизированные селекторы reselect. Side-effects в Redux: создание и использование Middlewares. Получение данных от сервера. Асинхронные экшны с redux-thunk.

Тема 7. React для SPA: react-router и продвинутое API

Разработка single-page applications. Использование react-router и продвинутое API React.JS. Настройка вложенных роутов. Выбор и настройка history для приложения. Объединение react-router и Redux. Обработка серверных ошибок и декларативное управление роутером. Использование context. Обзор Advanced API react-router. Использование props.children для композиции компонентов. Анимации в React, CSSTransitionGroup. GraphQL + Relay/Apollo. MobX vs Redux. Разные подходы к сайд-эффектам: redux-thunk, redux-loop, redux-saga, redux-observable.

7. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

В рамках данной учебной дисциплины обучающиеся выполняют самостоятельную внеаудиторную работу в виде повторения пройденного материала по всем темам, подготовки к устному опросу по темам 1, 3 и 6-7, изучения дополнительного теоретического материала, выполнения практических заданий по всем темам, подготовки к защите практических заданий и к промежуточной аттестации.

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При изучении данной дисциплины используются следующие образовательные технологии:

Тема занятия	Вид образовательной технологии	Форма проведения занятия
Тема 1. Основы Vue.js	Традиционная технология	Лекция Практическое занятие
Тема 2. Углубление во Vue.js и разработку компонентов	Традиционная технология	Лекция Практическое занятие

Тема 3. Основные принципы и архитектура	Традиционная технология	Лекция Практическое занятие
Тема 4. Компоненты и шаблоны	Интерактивная технология	Лекция Практическое занятие
Тема 5. Реактивность. Маршрутизация. Формы	Традиционная технология	Лекция Практическое занятие
Тема 6. Знакомство с экосистемой React. Построение приложений с Redux.js.	Традиционная технология	Лекция Практическое занятие
Тема 7. React для SPA: react-router и продвинутое API	Традиционная технология	Лекция Практическое занятие
	Интерактивная технология	Представление выполненных практических заданий по индивидуальным вариантам

9. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ДИСЦИПЛИНЕ

9.1. Формы контроля по дисциплине

Текущий контроль. В процессе изучения учебной дисциплины обучающиеся участвуют в устных теоретических опросах, выполняют практические работы и представляют результат выполненных практических заданий по индивидуальным вариантам. Результаты их выполнения являются основанием для выставления оценок текущего контроля по данной учебной дисциплине. Выполнение всех видов работ является обязательным для всех обучающихся. Обучающиеся, не выполнившие в полном объеме все практические задания и не представившие результат выполнения практических заданий по индивидуальным вариантам, не допускаются к сдаче экзамена по данной учебной дисциплине.

Промежуточная аттестация. Для контроля усвоения обучающимися данной дисциплины и достижения запланированных результатов обучения учебным планом предусмотрен экзамен. При выставлении итоговой оценки по дисциплине учитываются результаты текущего контроля, отражающие сформированность практических умений и навыков обучающегося.

9.2. Оценочные материалы (оценочные средства) для текущего контроля успеваемости и промежуточной аттестации по дисциплине

Текущий контроль

Примерные вопросы для устного опроса

1. Инструменты разработки Frontend приложений.

2. Синтаксис шаблонов Vue.js.
3. Работа с DOM событиями, атрибутами, формами.
4. Компонентный подход.
5. Создание и разработка приложений с @vue/cli.
6. Одностраничные приложения (SPA).
7. Назначение фреймворка Angular.
8. Модульная архитектура Angular.
9. Формы и валидация данных (реактивные формы и валидаторы).
10. Отличие React.js от других популярных фреймворков.
11. Особенности Redux.js: функциональный подход, Redux dev tools.
12. Анимации в React.

Промежуточная аттестация.

Список вопросов для подготовки к экзамену

1. JavaScript: назначение и основные инструменты для создания современных web-приложений
2. Инструменты разработки Frontend приложений.
3. Компонентный подход.
4. Однофайловые компоненты (SFC) и инкапсуляция стилей.
5. Создание и разработка приложений с @vue/cli.
6. Маршрутизация на клиенте, vue-router.
7. Unit-тестирование Vue.js приложения с Jest и vue-test-utils
8. Компоненты-обёртки. Компоненты-формы и однонаправленный поток данных.
9. Реактивность вне компонентов и реализация реактивности.
10. Основы рендеринга, Virtual DOM, render-функции, JSX.
11. Ограничения Vue.js, его зона ответственности и работа с другими библиотеками.
12. Scoped-slots, Renderless компоненты и другие подходы к разработке компонентов.
13. Composition API.
14. Назначение фреймворка Angular.
15. Модульная архитектура Angular.
16. Атрибутивные и структурные директивы Angular.
17. Структура приложения и практические рекомендации по её формированию в Angular.
18. Синтаксис привязки данных и шаблоны в Angular.
19. Формы и валидация данных (реактивные формы и валидаторы) в Angular.
20. Проекция контента в Angular.
21. Стратегия обнаружения изменений onPush в Angular.

22. Стили и шаблоны компонента в Angular.
23. Жизненный цикл компонента в Angular.
24. Шаблонные переменные, ViewChild, ContentChild.
25. Взаимодействие между модулями. ngClass и ngStyle. Создание атрибутивных директив.
26. Взаимодействие с пользователем, HostListener и HostBinding. Получение параметров в директивах. Структурные директивы ngIf, ngFor, ngSwitch.
27. HttpClient и отправка запросов.
28. Объект Observable и библиотека RxJS. Обработка ошибок.
29. Отправка данных в запросе. POST-запросы. Определение маршрутов. Создание ссылок. Параметры маршрута. Параметры строки запроса.
30. Модуль FormsModule и директива NgModel. Получение и изменение модели. Состояние модели и валидация. Директива NgForm. Reactive Forms
31. Создание простых компонентов React.
32. Отличие React.js от других популярных фреймворков.
33. Знакомство с экосистемой Использование сторонних сторонних компонентов и работа с формами.
34. Особенности Redux.js: функциональный подход, Redux dev tools.
35. Store как иммутабельный объект: Redux + Immutable.js.
36. Синхронный поток с Redux. React-redux для связи компонентов с логикой. Использование react-router и продвинутые элементы API React.JS.
37. Объединение react-router и Redux.
38. Обработка серверных ошибок и декларативное управление роутером.
39. Анимации в React, CSSTransitionGroup.
40. Разные подходы к сайд-эффектам: redux-thunk, redux-loop, redux-saga, redux-observable.

Порядок проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине для инвалидов и лиц с ОВЗ предусмотрен Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся.

10. РЕСУРСНАЯ СОСТАВЛЯЮЩАЯ

Для проведения занятий лекционного типа по данной дисциплине используются аудитории с медиа-оборудованием (проектор, экран, ноутбук) и учебной мебелью.

Для проведения занятий семинарского типа (практических занятий) по данной дисциплине используются компьютерные классы, оснащенные компьютерами с доступом в Интернет и необходимым программным обеспечением.

Для самостоятельной работы обучающихся используется помещение для самостоятельной работы обучающихся, оснащенное компьютерами с необходимым программным обеспечением и доступом в Интернет и электронную информационно-образовательную среду вуза.

Для проведения контроля самостоятельной работы по данной дисциплине используются компьютерные классы, оснащенные компьютерами с доступом в Интернет и необходимым программным обеспечением, а также доступом в электронную информационно-образовательную среду вуза.

Для проведения текущего контроля успеваемости и промежуточной аттестации по данной дисциплине используются компьютерные классы, оснащенные компьютерами с доступом в Интернет и необходимым программным обеспечением, а также доступом в электронную информационно-образовательную среду вуза.

Перечень лицензионного программного обеспечения

№ п/п	Наименование	Тип ресурса
1	Microsoft Windows	Сублицензионный договор АО «СофтЛайн Трейд» № /131 от 10.07.2020. Срок действия договора и лицензий - бессрочный (лицензионное соглашение Microsoft - Open Value Subscription для решений Education Solutions №V8265046)
2	Microsoft Office	
3	Visual Studio Enterprise (для учащихся, преподавателей и лабораторий)	
4	СПС КонсультантПлюс - справочно-правовая система отечественного производства	Лицензионный договор ООО «Консультант Плюс Тольятти» договор №251 от 01.01.2024 (лицензия бессрочная, договор ежегодно продлеваемый)
5	Антивирус Касперского отечественного производства	Сублицензионный договор СЛД АО «СофтЛайн Трейд» №Tr000840657 от 04.12.2023, срок действия договора до 11.02.2026 (250-499 Node 2 year Educational Renewal License)

Перечень свободно распространяемого программного обеспечения

1. Браузеры Yandex;
2. Облачные сервисы Yandex, Miro и др.;
3. СПС КонсультантПлюс - справочно-правовая система отечественного производства в свободном доступе в интернет.
4. CODEPEN (<https://codepen.io>) - интернет-сообщество, созданное для тестирования и показа сниппетов на основе HTML, CSS и JavaScript производства в свободном доступе в интернет.

В соответствии с Положением о создании специальных условий для инвалидов и лиц с ОВЗ информационно-технологическая база образовательного процесса

предусматривает использование материально-технических средств с учетом различных нозологий инвалидов и лиц с ОВЗ.

11. ЛИТЕРАТУРА

11.1. Основная литература

№	Библиографическое описание	Тип издания	Количество в библиотеке
1	Лисьев, Г.А. Программное обеспечение компьютерных сетей и web-серверов : учебное пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. — Москва : ИНФРА-М, 2023. — 145 с. — ISBN 978-5-16-013565-6. - URL: https://znanium.com/catalog/product/1914008	учебное пособие	ЭБС Знаниум
2	Беспалов, Д. А. Операционные системы реального времени и технологии разработки кроссплатформенного программного обеспечения: в 3 ч. Часть 3 : учебное пособие / Д. А. Беспалов, С. М. Гушанский, Н. М. Коробейникова ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2021. - 214 с. - ISBN 78-5-9275-3628-3. - URL: https://znanium.com/catalog/product/1894418	учебное пособие	ЭБС Знаниум

11.2 Дополнительная литература

1. Рындин, Н. А. Технологии разработки клиентских WEB-приложений на языке JavaScript : учебное пособие / Н. А. Рындин. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 54 с. — ISBN 978-5-7731-0888-7. // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/108188>.
2. Хортон, А. Разработка веб-приложений в ReactJS : практическое руководство / А. Хортон, Р. Вайс ; пер. с англ. Р. Н. Рагимова. - 2-е изд. - Москва : ДМК Пресс, 2023. - 255 с. - URL: <https://znanium.ru/catalog/product/2107174>
3. Вагин, Д. В. Современные технологии разработки веб-приложений : учебное пособие / Д. В. Вагин, Р. В. Петров. - Новосибирск : Изд-во НГТУ, 2019. - 52 с. - ISBN 978-5-7782-3939-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1866926>

Периодические издания :

1. Открытые системы. СУБД: журнал. - УБД ИВИС. - URL: <https://eivis.ru/browse/publication/64072>.
2. Программные продукты и системы: международный научно-практический журнал. - URL: <https://znanium.com/catalog/magazines/issues?ref=f9bfb0e-239e-11e4-99c7-90b11c31de4c>.

11.3 Современные профессиональные базы данных, информационные справочные

системы, электронные библиотечные системы:

1. Znanium: электронно-библиотечная система. – URL: <http://znanium.ru>.
2. eLIBRARY.RU: научная электронная библиотека. – URL: <https://elibrary.ru/>.
3. Цифровая библиотека IPRsmart. – URL: <https://www.iprbookshop.ru/>.
4. Консультант плюс: справочно-правовая система. – URL: <http://www.consultant.ru/> ; T:\consultantplus\cons.exe.
5. УБД ИВИС. – URL : <https://eivis.ru/basic/details>.
6. Электронная библиотека ТАУ. – URL: <http://83.234.207.58/MarcWeb2/Default.asp>.

В соответствии с Положением о создании специальных условий для инвалидов и лиц с ОВЗ информационно-технологическая база образовательного процесса предусматривает использование материально-технических средств с учетом различных нозологий инвалидов и лиц с ОВЗ.

12. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Успешное освоение дисциплины «Фреймворки front-end разработчика» предполагает посещение лекционных и практических занятий, самостоятельную работу с теоретическим материалом. При подготовке к лекциям и практическим занятиям, к выполнению самостоятельной работы обучающемуся рекомендуется прочитать материалы предыдущих лекций и источники основной литературы по изучаемой теме. Дополнительную литературу рекомендуется использовать для расширения кругозора и в том случае, если основной литературы недостаточно для понимания сути изучаемого вопроса.

В процессе изучения учебной дисциплины обучающиеся представляют результат выполненных практических заданий по индивидуальным вариантам. Практическое задание по индивидуальным вариантам представляет собой разработку приложения на выбранном фреймворке. Выполнение практического задания по индивидуальным вариантам и положительная отметка за него являются обязательным условием допуска обучающегося к сдаче экзамена, который проходит в форме устного ответа на теоретический вопрос. Для представления результатов практического задания по индивидуальным вариантам обучающийся готовит презентацию и формирует репозиторий. Обучающиеся делают презентацию для защиты практического задания по предложенной структуре, после выступления демонстрируют результат разработки и отвечает на вопросы аудитории.

Для закрепления приобретенных знаний, умений и навыков, для развития способностей к самообучению в дисциплине предусмотрена самостоятельная работа. Самостоятельная работа может выполняться обучающимся дома или в аудиториях Академии, специально отведенных для самостоятельной работы и оснащенных необходимым техническим и программным обеспечением, доступом к ЭИОС и ЭБС.

Подготовка к лекциям и практическим занятиям с последующим участием в устном опросе предполагает: систематическое чтение конспектов лекций, учебников и источников дополнительной литературы; работу со справочниками и нормативными документами; аналитическую обработку, составление таблиц и схем для систематизации изученного материала; ответы на контрольные вопросы и составление плана и/или тезисов ответов; тренировка в кодировании на различных фреймворках.

Консультации по выполнению практических и самостоятельных работ, обсуждение отметок и допущенных ошибок осуществляется во время КСР на кафедре прикладной информатики или в аудитории по расписанию. Консультации могут осуществляться также посредством асинхронного (почта, ЭИОС) и синхронного (Zoom) коммуникационного взаимодействия по предварительной договоренности с преподавателем.

Формой промежуточного контроля по дисциплине выступает экзамен, который проводится в форме устного ответа на теоретический вопрос. Критерии допуска к экзамену (выполнение всех практических заданий, положительная отметка за выполнение практического задания по индивидуальным вариантам) и условия его сдачи озвучиваются преподавателем на первых занятиях по дисциплине.

13. ЛИСТ СОГЛАСОВАНИЯ

Составил:

О.И. Подулыбина, ст.преп.



(подпись)

Заведующий кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий выпускающей кафедрой

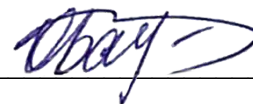
Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Директор БИК

О.В. Балакина



(подпись)

Начальник ООУП

С.В. Фирсова



(подпись)