

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**  
Заместитель директора

**Ю.О. Соболев**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Основы Frontend-разработки
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Разработка IT-продукта
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Экзамен

Аудиторных часов: 20 всего, в том числе:

лекции: 10 час.

семинары: 10 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 85 час.

Подготовка к экзамену: 30 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: К.А. Лапин, старший методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и  
онлайн-образования "Пуск" 01.03.2025

## Аннотация

Дисциплина "Основы фронтенд-разработки" направлена на формирование у студентов базовых основ фронтенд разработки. В рамках курса студенты изучат HTML и CSS, включая основные теги и стилизацию элементов, а также синтаксис JavaScript, манипуляции с DOM и переменными, рассмотрят вопросы, связанные с созданием интерактивных веб-приложений, работой с AJAX и обработке пользовательского ввода, познакомятся с фреймворками и библиотеками, методами хранения данных на клиенте и безопасностью. Завершающий модуль посвящен оптимизации производительности фронтенда и использованию инструментов разработчика. Курс поможет студентам приобрести необходимые навыки для создания современных веб-приложений и понять важность взаимодействия фронтенда и бэкенда.

## 1. Цели и задачи

### Цель дисциплины

- предоставление студентам базовых знаний и формирование навыков в области фронтенд-разработки, необходимых для успешного выполнения задач в качестве бэкенд-разработчиков.

### Задачи дисциплины

- освоение основ HTML, CSS и JavaScript для создания структурированных и стилизованных веб-страниц;
- изучение методов манипулирования DOM-элементами и обработки событий для создания интерактивных пользовательских интерфейсов;
- освоение принципов асинхронного программирования и работы с AJAX для динамической загрузки данных на веб-страницы;
- введение в основные фреймворки и библиотеки фронтенд-разработки для создания простых компонентов и понимания их взаимодействия;
- ознакомление с методами хранения данных на клиенте и обеспечения безопасности доступа к ним;
- изучение методов оптимизации производительности фронтенда и использование инструментов разработчика для эффективной работы с веб-приложениями;
- развитие практических навыков через выполнение заданий, направленных на применение изученного материала в реальных проектах.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
	УК-1.3 Разрабатывает стратегию достижения поставленной цели как последовательность шагов, предвидя результат каждого из них и оценивая их влияние на внешнее окружение планируемой деятельности и на взаимоотношения участников этой деятельности
УК-2 Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1 Формулирует в рамках обозначенной проблемы, цель, задачи, актуальность, значимость (научную, практическую, методическую и иную в зависимости от типа проекта), ожидаемые результаты и возможные сферы их применения
	УК-2.2 Способен прогнозировать результат деятельности и планировать последовательность шагов для достижения данного результата. Формирует план-график реализации проекта в целом и план контроля его выполнения

ОПК-1 Владеет системой фундаментальных научных знаний в области информатики и вычислительной техники	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания и новые научные принципы и методы исследований в области информатики и вычислительной техники
	ОПК-1.3 Понимает междисциплинарные связи в области информатики и вычислительной техники и способен их применять при решении задач профессиональной деятельности
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-3.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.2 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
	ОПК-3.4 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основы HTML, CSS и JavaScript для создания структурированных и стилизованных веб-страниц;
- принципы манипулирования DOM-элементами и обработки событий для создания интерактивных пользовательских интерфейсов;
- принципы асинхронного программирования;
- основные концепции фреймворков и библиотек фронтенд-разработки, таких как Vue, Angular, React;
- как создавать React-компоненты и подключать к ним Bootstrap;
- методы хранения данных на клиенте и обеспечения безопасности доступа к ним.

уметь:

- создавать простые компоненты и взаимодействовать с ними в выбранном фреймворке или библиотеке;
- создавать интерфейсы с помощью Bootstrap;
- выполнять запросы GET, PUT, PATCH, DELETE;
- использовать куки и локальное хранилище для хранения данных на клиенте и обеспечения их безопасности.

владеть:

- навыками применения знания HTML, CSS и JavaScript для создания интерактивных веб-приложений.

### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

#### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Основы JavaScript	1	1		12
2	React, Vue, Angular. Компоненты в React	1	1		11
3	Стилизация React-компонентов	1	1		11
4	Вёрстка на основе Bootstrap	1	1		8
5	Пропсы и состояние	1	1		8
6	Разработка форм	1	1		10
7	React Hook Form	1	1		10
8	Работа с API: GET-запрос	1	1		5
9	Обработка ожидания и ошибок. Запросы PUT, PATCH, DELETE	1	1		5
10	Клиентский Routing	1	1		5
Итого часов		10	10		85
Подготовка к экзамену		30 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

#### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

##### Семестр: 2 (Весенний)

##### 1. Основы JavaScript

Работа с массивами: Изучение массивов в JavaScript, их создание и методов для манипуляции элементами. Функции: Понятие функций, их синтаксис и применение.

Объекты: Введение в объекты, создание и доступ к свойствам. Решение задач на языке JavaScript.

##### 2. React, Vue, Angular. Компоненты в React

Основы HTML: Возможности JavaScript в браузере. Способ поиска HTML-элементов: Использование методов для поиска элементов на странице. Объект события: Обработка событий в JavaScript. DOM: Манипуляции с DOM-элементами. Работа с HTML-формами: Создание и обработка форм. Изменение структуры HTML-документа: Динамическое изменение структуры страницы. Написание простых компонентов на React.

##### 3. Стилизация React-компонентов

Введение в вёрстку: Основы CSS для стилизации компонентов. Синхронные занятия: Вебинар «Стилизация React-компонентов». Написание простых компонентов на React со стилизацией.

##### 4. Вёрстка на основе Bootstrap

Основные преимущества и возможности Bootstrap. Основы сетки Bootstrap: 12-колоночная система для адаптивных макетов. Готовые компоненты Bootstrap: Кнопки, формы, навигационные панели. Создание адаптивных макетов. Создание веб-страницы с использованием Bootstrap.

##### 5. Пропсы и состояние

Передача данных между компонентами с помощью пропсов. Условный рендеринг и списковый рендеринг. Разработка интерактивной веб-страницы на основе компонентов Bootstrap React с использованием useState, props, условного рендеринга, списков.

## 6. Разработка форм

Основы создания форм в React. Обработка событий. Управление состоянием: Как использовать useState для хранения и обновления данных формы. Валидация форм: Базовые методы проверки введенных данных. Интеграция с Bootstrap. Создание простой формы сбора данных с обработкой событий и валидацией. Домашнее задание: Разработка формы сбора данных.

## 7. React Hook Form

Введение в React Hook Form. Использование хука useForm для создания и управления состоянием форм. Обработка событий с помощью handleSubmit. Валидация форм и отображение ошибок. Создание сложных форм с несколькими шагами. Интеграция с UI-библиотеками, такими как Material-UI или Bootstrap. Разработка формы с несколькими шагами с большим количеством полей или с несколькими шагами заполнения формы.

## 8. Работа с API: GET-запрос

Выполнение GET-запросов с помощью Fetch API. Объявление состояния для хранения данных из API. Обработка ответа и преобразование его в JSON. Отрисовка полученных данных в компоненте React. Использование хука useEffect для выполнения запросов при монтировании компонента. Выполнение GET-запросов и отрисовка полученных данных.

## 9. Обработка ожидания и ошибок. Запросы PUT, PATCH, DELETE

Обработка ожидания запросов: использование состояний loading и error. Выполнение запросов PUT, PATCH и DELETE с помощью Fetch API. Настройка заголовков и тела запроса для разных методов. Обработка ошибок и отображение сообщений об ошибках. Выполнение CRUD-операций в React-приложении.

## 10. Клиентский Routing

Введение в клиентский роутинг: основные концепции и преимущества. Использование библиотеки React Router для создания навигационного меню. Создание маршрутов и параметров маршрутов. Переходы между страницами и обработка параметров. Создание многостраничного сайта в формате SPA с навигационным меню.

## **5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Занятия по учебной дисциплине проводятся с применением дистанционных образовательных технологий. Каждый обучающийся обеспечен доступом к образовательной платформе (LMS).

## **6. Перечень рекомендуемой литературы**

### Основная литература

1. HTML, CSS, скрипты: практика создания сайтов [Текст], самоучитель : [для веб-разработчиков]/А. Ю. Ломов, -СПб., БХВ-Петербург, 2007
2. Веб-мастеринг: HTML, CSS, JavaScript, PHP, CMS, графика, раскрутка [Текст]/П. А. Ташков, -СПб., Питер, 2009

### Дополнительная литература

Полуэктова, Н. Р. Разработка веб-приложений: учебное пособие для вузов / Н. Р. Полуэктова. — 2-е изд. — Москва: Издательство Юрайт, 2024. — 204 с. — (Высшее образование). — ISBN 978-5-534-18645-1. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/545238>

Тузовский, А. Ф. Проектирование и разработка web-приложений: учебное пособие для вузов / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2024. — 219 с. — (Высшее образование). — ISBN 978-5-534-16300-1. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/537106>

Сысолетин, Е. Г. Разработка интернет-приложений: учебное пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев; под научной редакцией Л. Г. Доросинского. — Москва: Издательство Юрайт, 2023. — 90 с. — (Высшее образование). — ISBN 978-5-9916-9975-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/514303>

Янцев, В. В. JavaScript. Креативное программирование: учебное пособие для вузов / В. В. Янцев. — 2-е изд., стер. — Санкт-Петербург: Лань, 2024. — 232 с. — ISBN 978-5-507-49267-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/383837> — Режим доступа: для авториз. пользователей.

Кириченко, А. В. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков / А. В. Кириченко, А. П. Никольский, Е. В. Дубовик. — Санкт-Петербург: Наука и Техника, 2021. — 432 с. — ISBN 978-5-94387-271-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/191460> — Режим доступа: для авториз. пользователей.

Янцев, В. В. JavaScript. Готовые программы: учебное пособие для вузов / В. В. Янцев. — Санкт-Петербург: Лань, 2021. — 200 с. — ISBN 978-5-8114-6873-7. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/165842> — Режим доступа: для авториз. пользователей.

Хорстманн, К. С. Современный JavaScript для нетерпеливых: практическое пособие / Кэй С. Хорстманн; пер. с англ. А. А. Слинкина. — Москва: ДМК Пресс, 2021. — 288 с. — ISBN 978-5-97060-177-8. — Текст: электронный. — URL: <https://znanium.com/catalog/product/1225356> — Режим доступа: по подписке.

## **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

Не используются

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

Студенту для занятий потребуются:

1. Google Drive / Yandex disk для доступа к материалам курса
2. Zoom
3. LMS МФТИ
4. Приложение для коммуникации с преподавателями
5. Ноутбук для участия в интерактивных занятиях

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. В результате изучения дисциплины студент должен знать основные определения дисциплины, уметь применять полученные знания для решения прикладных задач.

Успешное освоение курса требует:

- посещения всех онлайн-занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение материалов на платформе дистанционного обучения и рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без вспомогательных материалов и конспектов отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

Возможен текущий контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

## ПРИЛОЖЕНИЕ

### ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Разработка IT-продукта центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	<u>1</u>
<b>квалификация:</b>	магистр
Семестр, формы промежуточной аттестации: 2 (весенний) - Экзамен	
<b>Разработчик:</b>	К.А. Лапин, старший методист



## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
	УК-1.3 Разрабатывает стратегию достижения поставленной цели как последовательность шагов, предвидя результат каждого из них и оценивая их влияние на внешнее окружение планируемой деятельности и на взаимоотношения участников этой деятельности
УК-2 Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1 Формулирует в рамках обозначенной проблемы, цель, задачи, актуальность, значимость (научную, практическую, методическую и иную в зависимости от типа проекта), ожидаемые результаты и возможные сферы их применения
	УК-2.2 Способен прогнозировать результат деятельности и планировать последовательность шагов для достижения данного результата. Формирует план-график реализации проекта в целом и план контроля его выполнения
ОПК-1 Владеет системой фундаментальных научных знаний в области информатики и вычислительной техники	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания и новые научные принципы и методы исследований в области информатики и вычислительной техники
	ОПК-1.3 Понимает междисциплинарные связи в области информатики и вычислительной техники и способен их применять при решении задач профессиональной деятельности
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-3.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.2 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
	ОПК-3.4 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Основы Frontend-разработки» обучающийся должен:

**знать:**

- основы HTML, CSS и JavaScript для создания структурированных и стилизованных веб-страниц;
- принципы манипулирования DOM-элементами и обработки событий для создания интерактивных пользовательских интерфейсов;
- принципы асинхронного программирования;
- основные концепции фреймворков и библиотек фронтенд-разработки, таких как Vue, Angular, React;
- как создавать React-компоненты и подключать к ним Bootstrap;
- методы хранения данных на клиенте и обеспечения безопасности доступа к ним.

**уметь:**

- создавать простые компоненты и взаимодействовать с ними в выбранном фреймворке или библиотеке;
- создавать интерфейсы с помощью Bootstrap;
- выполнять запросы GET, PUT, PATCH, DELETE;
- использовать куки и локальное хранилище для хранения данных на клиенте и обеспечения их безопасности.

**владеть:**

- навыками применения знания HTML, CSS и JavaScript для создания интерактивных веб-приложений.

### 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

В начале занятия проводится опрос по материалам предыдущего занятия и даются задачи для совместного решения.

Пример задания:

Цель выполнения задания: приобрести практические навыки во фронтенд разработке.

Описание задания:

Создать базовую структуру HTML страницы с формой для добавления новой задачи и списком задач. Использовать CSS для стилизации элементов страницы. Написать JavaScript код для обработки пользовательского ввода и взаимодействия с интерфейсом.

Критерии для самооценки по чек-листу:

HTML:

Создана базовая структура HTML страницы.

Добавлена форма для ввода новой задачи.

Добавлен список задач.

CSS:

Применены стили для элементов формы и списка задач.

Использованы CSS-селекторы для выбора элементов страницы.

Реализовано выравнивание и оформление элементов страницы для улучшения внешнего вида.

JavaScript:

Написан код для обработки событий ввода новой задачи.

Реализована функциональность добавления новой задачи в список.

Реализована возможность удаления задачи из списка при необходимости.

Общее:

Все элементы страницы корректно отображаются и взаимодействуют друг с другом.

Код организован и читаем, с использованием комментариев при необходимости.

Приложение работает без ошибок и предоставляет пользователю удобный интерфейс для работы с задачами.

Дополнительно задание (по желанию):

Добавлена анимация или эффекты при взаимодействии с элементами страницы.

Реализована проверка на пустой ввод в форме добавления задачи.

Добавлены дополнительные функции, такие как редактирование задачи или отметка о выполнении.

#### **4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся**

Экзамен проводится в форме защиты проекта.

Предполагаемый результат:

Разработка простого веб-приложения "Список задач", которое позволяет пользователям добавлять, просматривать, редактировать и удалять задачи.

Описание задания:

Создать базовую структуру HTML страницы с формой для добавления новой задачи и списком задач.

Использовать CSS для стилизации элементов страницы, чтобы приложение выглядело привлекательно и удобно для использования.

Написать JavaScript код для обработки пользовательского ввода, отправки запросов на сервер и обновления интерфейса без перезагрузки страницы.

Разработать серверную часть приложения, используя выбранный вами язык программирования (например, Node.js), чтобы обрабатывать запросы от клиента и выполнять операции CRUD в базе данных.

Проверить работу приложения, убедившись, что пользователь может успешно добавлять, просматривать, редактировать и удалять задачи через веб-интерфейс.

Критерии оценивания:

Корректность работы веб-приложения: возможность добавления, просмотра, редактирования и удаления задач.

Правильное взаимодействие клиентской и серверной частей приложения: отправка и обработка HTTP-запросов.

Качество кода: чистота, читаемость и модульность JavaScript кода, правильное использование HTML и CSS.

Работоспособность и стабильность приложения: отсутствие ошибок и непредвиденного поведения.

Возможность расширения функциональности приложения, например, добавление функционала фильтрации или сортировки задач.

#### **Критерии оценивания**

Оценка отлично (10 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5 баллов) - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2 балла) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1 балл) - выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Экзамен проходит в на lms платформе.

Время проведения экзамена оставляет 2 академических часа.

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины.