

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 23.12.2025 13:42:26  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Разработка программно-  
информационных систем»



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования  
**«Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И.Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)**

---

**РАБОЧАЯ ПРОГРАММА**

дисциплины

**«ОСНОВЫ ФРОНТЕНД-РАЗРАБОТКИ Ч.2»**

для подготовки бакалавров

по направлению

09.03.04 «Программная инженерия»

по профилю

**«Разработка программно-информационных систем»**

Санкт-Петербург

2025

## **ЛИСТ СОГЛАСОВАНИЯ**

Разработчики:

доцент, к.т.н., доцент Беляев С.А.

Рабочая программа рассмотрена и одобрена на заседании кафедры МОЭВМ  
20.01.2025, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 28.01.2025, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## **1 СТРУКТУРА ДИСЦИПЛИНЫ**

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	МОЭВМ
Общая трудоемкость (ЗЕТ)	3
Курс	3
Семестр	6

### **Виды занятий**

Электронные лекции (акад. часов)	34
Электронные практические (академ. часов) (академ. часов)	34
Иная контактная работа (академ. часов)	1
Все контактные часы (академ. часов)	1
Самостоятельная работа, включая часы на контроль (академ. часов)	39
Всего (академ. часов)	108

### **Вид промежуточной аттестации**

Дифф. зачет (курс)	3
--------------------	---

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ОСНОВЫ ФРОНТЕНД-РАЗРАБОТКИ Ч.2»**

Дисциплина является второй частью курса, который посвящен основам, необходимым для разработчиков интернет-сайтов и web-приложений. Рассматриваются основы построения web-приложений с использованием HTML, CSS. Обучающиеся последовательно изучают основы протокола HTTP, настройку web-сервера, основы JavaScript, TypeScript, сборку кода на Vite. Обсуждаются вопросы разработки клиентских приложений с использованием библиотеки React. Обучающиеся получают навыки тестирования React-приложений.

#### **SUBJECT SUMMARY**

#### **«FUNDAMENTALS OF FRONTEND DEVELOPMENT PART 2»**

The discipline is the second part of the course, which focuses on the basics necessary for developers of websites and web applications. The basics of building web applications using HTML and CSS are considered. Students consistently learn the basics of the HTTP protocol, configuring a web server, the basics of JavaScript, TypeScript, and building code on Vite. The issues of developing client applications using the React library are discussed. Students gain skills in testing React applications.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью дисциплины является приобретение теоретических знаний о современных web-технологиях и фреймворках и формирование практических умений и навыков по всем этапам разработки web-приложений, включая разработку web-серверов, создания и оформления web-клиентов, тестирования приложений.
2. Задачами изучения дисциплины является приобретение теоретических знаний по проектированию web-приложений, формирование практических умений и навыков по разработке серверной и клиентской части web-приложений, оформлению и тестированию приложений.
3. Обучающиеся приобретают знания основных подходов по построению клиентской и серверной части web-приложений, по оформлению web-страниц, в том числе для просмотра с мобильных устройств, по решению типовых задач тестирования, сборки и обеспечения безопасности web-приложений с использованием языков JavaScript, TypeScript.
4. В результате изучения курса приобретаются умения разрабатывать клиентскую и серверную часть web-приложений, осуществлять настройку, тестирование и автоматизированную сборку web-приложений.
5. У обучающихся формируются навыки создания статических HTML-страниц, динамических web-приложений на языках JavaScript и TypeScript, навыки применения современных фреймворков разработки клиентских и серверных частей web-приложений.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Алгоритмы и структуры данных»

2. «Основы фронтенд-разработки ч.1»

и обеспечивает подготовку выпускной квалификационной работы.

### **3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

<b>Код компетенции/ индикатора компетенции</b>	<b>Наименование компетенции/индикатора компетенции</b>
ПК-0	Способен разрабатывать информационные модели и применять их для решения задач профессиональной деятельности
ПК-0.1	<i>Знает современные виды информационных моделей, применяемых при решении задач профессиональной деятельности</i>
ПК-0.2	<i>Создает и модифицирует информационные модели для решения задач профессиональной деятельности</i>
ПК-0.3	<i>Применяет информационные модели для решения задач профессиональной деятельности</i>

## **4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

### **4.1 Содержание разделов дисциплины**

#### **4.1.1 Наименование тем и часы на все виды нагрузки**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>ЭЛек, ач</b>	<b>ЭПр, ач</b>	<b>ИКР, ач</b>	<b>СР, ач</b>
1	Введение	1			
2	Основы разработки с помощью React	10	10		12
3	Роутинг и авторизация	10	10		12
4	Тестирование React-приложений	12	14		14
5	Заключение	1		1	1
	Итого, ач	34	34	1	39
	Из них ач на контроль	0	0	0	0
	Общая трудоемкость освоения, ач/зе	108/3			

#### **4.1.2 Содержание**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
1	Введение	Содержание второй части курса
2	Основы разработки с помо- щью React	Подготовка окружения, сборка проектов Vite или Webpack, установка ESLint с набором правил Airbnb, Prettier, использование SCSS, Stylelint, Husky. React и JSX, синтаксис JSX: основы, списки и события, функциональные компоненты. Классовые компоненты, жизненный цикл классовых компонентов, virtual DOM. Хуки, рефы, useState, useEffect, useLayoutEffect, useRef, кастомные хуки. Установка и использование расширения React DevTools, сборка проекта. Импорт модулей и изображений, работа с CSS, шрифты, CSS-модули, использование clsx. Storybook, реализация компонента в изоляции, разработка через тестирование, сборка проекта как библиотеки для переисполь- зования.

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
3	Роутинг и авторизация	Маршрутизация на стороне сервера и клиента, клиентский роутинг и History API, установка и подключение React Router, создание маршрутов, навигация и ссылки, динамические маршруты, фильтрация и сортировка значений. Идентификация, аутентификация и авторизация, токены, авторизация доступа с помощью сессий, сохранение данных о пользователе, cookie в веб-приложениях, защита маршрутов на фронтенде, выход из системы. Открытие и закрытие соединения, события WebSocket, получение и отправка данных, отладка и тестирование WebSocket в консоли разработчика. Redux Toolkit, RTK Query, redux-thunk, redux-saga, кастомные middleware и enhancer, интеграция с API, Redux DevTools.
4	Тестирование React-приложений	Принципы выстраивания удобной для тестирования структуры компонентов. Оптимизация приложения для тестирования. Проведение юнит-и интеграционного тестирования. Подготовка для публикации, развертывание на сервере.
5	Заключение	Общие выводы по изложенному материалу. Связь с последующими дисциплинами траектории.

## 4.2 Перечень лабораторных работ

Лабораторные работы не предусмотрены.

## 4.3 Перечень практических занятий

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
1. Роутинг и авторизация	12
2. Развёртывание	10
3. Тестирование	12
Итого	34

## 4.4 Курсовое проектирование

Курсовая работа (проект) не предусмотрены.

## 4.5 Реферат

Реферат не предусмотрен.

#### **4.6 Индивидуальное домашнее задание**

Индивидуальное домашнее задание не предусмотрено.

#### **4.7 Доклад**

Доклад не предусмотрен.

#### **4.8 Кейс**

Кейс не предусмотрен.

#### **4.9 Организация и учебно-методическое обеспечение самостоятельной работы**

Изучение дисциплины сопровождается самостоятельной работой студентов с рекомендованными преподавателем литературными источниками и информационными ресурсами сети Интернет.

Планирование времени для изучения дисциплины осуществляется на весь период обучения, предусматривая при этом регулярное повторение пройденного материала. Обучающимся, в рамках внеаудиторной самостоятельной работы, необходимо регулярно дополнять сведениями из литературных источников материал, законспектированный на лекциях. При этом на основе изучения рекомендованной литературы целесообразно составить конспект основных положений, терминов и определений, необходимых для освоения разделов учебной дисциплины.

Особое место уделяется консультированию, как одной из форм обучения и контроля самостоятельной работы. Консультирование предполагает особым образом организованное взаимодействие между преподавателем и студентами, при этом предполагается, что консультант либо знает готовое решение, которое он может предписать консультируемому, либо он владеет способами деятель-

ности, которые указывают путь решения проблемы.

Самостоятельное изучение студентами теоретических основ дисциплины обеспечено необходимыми учебно-методическими материалами (учебники, учебные пособия, конспект лекций и т.п.), выполненными в печатном или электронном виде.

Изучение студентами дисциплины сопровождается проведением регулярных консультаций преподавателей, обеспечивающих практические занятия по дисциплине, за счет бюджета времени, отводимого на консультации (внеаудиторные занятия, относящиеся к разделу «Самостоятельные часы для изучения дисциплины»).

В случае применения ДОТ с заменой аудиторных занятий:

Самостоятельной записи на курс нет. Студент заходит на курс, используя логин/пароль от единой учетной записи университета (единий логин и пароль). Каждую неделю будет доступна новая тема курса: видеолекции, кратко раскрывающие содержание каждой темы, презентации и конспекты, с которыми обучающиеся смогут ознакомиться в любое удобное время. Все темы включают практические занятия, которые предусматривают самостоятельное выполнение заданий, а также задания с автоматической проверкой, результаты которых учитываются при общей аттестации полученных знаний. В конце каждой лекции необходимо пройти небольшой контрольный тест, который покажет насколько усвоен предложенный материал. Рекомендуем изучать материал последовательно, что существенно облегчит работу. У каждого контрольного задания имеется своя форма (тест или практическое задание) есть срок выполнения (окончательный срок), по истечении которого даже правильные ответы система принимать не будет! В расписании курса указан окончательный срок каждого задания, который варьируется от двух до четырех недель в зависимости от его сложности. Весь учебный курс рассчитан на 16 недель. Его итоги будут подведены в течение нескольких недель после его окончания.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	15
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	6
Самостоятельное изучение разделов дисциплины	6
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	6
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	0
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференцированному зачету, экзамену	6
<b>ИТОГО СРС</b>	<b>39</b>

## **5 Учебно-методическое обеспечение дисциплины**

### **5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины**

<b>№ п/п</b>	<b>Название, библиографическое описание</b>	<b>К-во экз. в библ.</b>
<b>Основная литература</b>		
1	Беляев С. А. Разработка игр на языке JavaScript : учебное пособие для вузов / С. А. Беляев, 2023. -152 с. -Текст : электронный.	неогр.
2	Беляев, Сергей Алексеевич. Web-технологии. Современные фреймворки : лаб. практикум / С. А. Беляев, 2022. -1 эл. опт. диск (CD-ROM). -Текст : электронный.	неогр.
<b>Дополнительная литература</b>		
1	Дронов В. А. Nodejs, Express, MongoDB и React : 23 урока для начинающих / В. А. Дронов, 2024. -608 с. -Текст : непосредственный.	неогр.

### **5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины**

<b>№ п/п</b>	<b>Электронный адрес</b>
1	React. JavaScript-библиотека для создания пользовательских интерфейсов <a href="https://ru.reactjs.org/">https://ru.reactjs.org/</a>

### **5.3 Адрес сайта курса**

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=23643>

## **6 Критерии оценивания и оценочные материалы**

### **6.1 Критерии оценивания**

Для дисциплины «Основы фронтенд-разработки ч.2» формой промежуточной аттестации является дифф. зачет. Оценивание качества освоения дисциплины производится с использованием рейтинговой системы.

#### **Дифференцированный зачет**

<b>Оценка</b>	<b>Количество баллов</b>	<b>Описание</b>
Неудовлетворительно	0 – 7	теоретическое содержание курса не освоено, необходимые практически навыки и умения не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над курсом не приведет к существенному повышению качества выполнения учебных заданий
Удовлетворительно	8 – 11	теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки и умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки
Хорошо	12 – 13	теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки и умения сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Отлично	14 – 15	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки и умения сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено количеством баллов, близким к максимальному

## **Особенности допуска**

К дифференцированному зачету допускаются студенты, показавшие прогресс в ЭОС Moodle (не менее 80%) и выполнившие 3 практические работы на оценку не менее "удовлетворительно".

Дифференцированный зачет проводится в виде тестирования в электронной образовательной среде Moodle. Тест содержит 15 вопросов с развернутыми ответами. Каждый вопрос теста оценивается в 1 балл в случае правильного ответа. Оценка за дифф. зачет выставляется в зависимости от количества баллов, полученных за итоговый тест следующим образом:

"Неудовлетворительно" -менее 8 баллов

"Удовлетворительно" -от 8 до 11 баллов

"Хорошо" -от 12 до 13 баллов

"Отлично" -от 14 до 15 баллов.

## **6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине**

### **Вопросы к дифф.зачету**

<b>№ п/п</b>	<b>Описание</b>
1	Сборка проектов Vite или Webpack
2	Установка ESLint с набором правил Airbnb, Prettier
3	Использование SCSS, Stylelint, Husky, React и JSX
4	Классовые компоненты, жизненный цикл классовых компонентов, virtual DOM
5	Установка и использование расширения React DevTools, сборка проекта
6	Импорт модулей и изображений, работа с CSS, шрифты, CSS-модули, использование clsx
7	Storybook, реализация компонента в изоляции
8	Маршрутизация на стороне сервера и клиента
9	Клиентский роутинг и History API, установка и подключение React Router
10	Создание маршрутов, навигация и ссылки, динамические маршруты
11	Идентификация, аутентификация и авторизация, токены
12	Cookie в веб-приложениях
13	Открытие и закрытие соединения, события WebSocket, получение и отправка данных
14	Отладка и тестирование WebSocket в консоли разработчика

15	Redux Toolkit, RTK Query
16	Redux-thunk, redux-saga
17	Кастомные middleware и enhancer, интеграция с API, Redux DevTools.
18	Принципы выстраивания удобной для тестирования структуры компонентов
19	Оптимизация приложения для тестирования
20	Подготовка для публикации, развертывание на сервере

## Вариант теста

### 1. Что такое Refs в React?

- a) Стили для компонентов.
- b) Способ доступа к DOM-элементам или экземплярам компонентов непосредственно из React.
- c) Переменные состояния.
- d) Маршруты в приложении.

**Правильный ответ:** b)

### 2. Какие существуют способы стилизации React-компонентов?

- a) Только inline стили.
- b) CSS-модули, Styled Components, JSS, обычные CSS файлы.
- c) Только внешние CSS файлы.
- d) Использование только JavaScript.

**Правильный ответ:** b)

### 3. Как создать маршрутизацию (routing) в React-приложении?

- a) Использовать только ссылки HTML.
- b) Использовать библиотеки, такие как React Router.
- c) Маршрутизация не требуется в React.
- d) Использовать только JavaScript.

**Правильный ответ: b)**

**4. Что такое Redux и для чего он используется в React?**

- a) Фреймворк для стилизации компонентов.
- b) Библиотека для управления состоянием приложения, особенно полезна для сложных приложений с большим количеством компонентов.
- c) Библиотека для работы с базами данных.
- d) Инструмент для оптимизации производительности.

**Правильный ответ: b)**

**5. Что такое keys в React и для чего они используются?**

- a) Стили для компонентов.
- b) Уникальные идентификаторы, используемые при рендеринге списков для отслеживания изменений элементов.
- c) Переменные состояния.
- d) Параметры для передачи данных между компонентами.

**Правильный ответ: b)**

**6. Как предотвратить обновление компонента в React, если props не изменились?**

- a) Использовать shouldComponentUpdate (в классовых компонентах) или React.memo (для функциональных).
- b) Использовать setState.
- c) Использовать useEffect.
- d) Невозможно предотвратить обновление компонента.

**Правильный ответ: a)**

**7. Что такое обработка событий в React?**

- a) Метод для создания анимаций.
- b) Способ реагирования на действия пользователя (например, клики, нажатия клавиш).
- c) Функция для рендеринга компонентов.
- d) Процесс стилизации элементов.

**Правильный ответ:** b)

## **8. Как передать данные от дочернего компонента к родительскому в React?**

- a) Изменить состояние родительского компонента напрямую.
- b) Передать функцию обратного вызова (callback) через props от родительского компонента к дочернему.
- c) Использовать глобальное состояние (например, Redux).
- d) Нельзя передавать данные от дочернего компонента к родительскому.

**Правильный ответ:** b)

## **9. Что такое React Hooks?**

- a) Новый способ стилизации компонентов.
- b) Функции, позволяющие использовать состояние и другие возможности React в функциональных компонентах.
- c) Метод для создания классов.
- d) Библиотека для работы с данными.

**Правильный ответ:** b)

## **10. Как обновить состояние компонента в React?**

- a) Напрямую изменить значение свойства state.
- b) Использовать метод setState.

с) Использовать функцию `updateState`.

д) Обновление состояния не требуется.

**Правильный ответ:** б)

## Образцы задач (заданий) для контрольных (проверочных) работ

### Пример практической работы

Реализовать проект о путешествиях по стране. Он должен корректно отображаться на экранах популярных размеров.

Дизайн-макет — это демонстрация, как должен выглядеть сайт на различных разрешениях. Используйте пять интервалов:

- от 1280 пикселей — от стандартных ноутбуков и до бесконечности,
- от 1024 и до 1280 — для небольших ноутбуков,
- от 768 и до 1024 пикселей — от вертикального до горизонтального положения планшета,
- от 425 и до 768 пикселей — от большого смартфона до вертикального положения планшета,
- до 425 пикселей — мобильные устройства.

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

### **6.3 График текущего контроля успеваемости**

<b>Неделя</b>	<b>Темы занятий</b>	<b>Вид контроля</b>
1	Основы разработки с помощью React	
2		
3		
4		
5		Практическая работа
6	Роутинг и авторизация	
7		
8		
9		
10		Практическая работа
11	Тестирование React-приложений	
12		
13		
14		
15		
16		Практическая работа

### **6.4 Методика текущего контроля**

#### **на лекционных занятиях**

Текущий контроль включает в себя:

- контроль прогресса в ЭОС Moodle (не менее 80%).

#### **на практических занятиях**

- контроль прогресса в ЭОС Moodle (не менее 80%)

В процессе обучения по дисциплине студент обязан выполнить 3 практические работы в ЭОС Moodle.

Студент должен показать: понимание постановки задачи, подхода к ее решению, ход решения, выбор тех или иных методик решения задачи. Преподаватель определяет глубину понимания теоретического материала, который лежит в основе решения задачи практической работы, а также самостоятельность ее выполнения.

**Критерии оценивания практических работ:**

«отлично» - вопрос раскрыт полностью, задача решена правильно

«хорошо» - вопрос раскрыт не полностью, задача решена частично

«удовлетворительно» - в ответе на вопрос имеются существенные ошибки; задача не решена или решена неправильно, ход решения правильный

«неудовлетворительно» - отсутствует ответ на вопрос или содержание ответа не совпадает с поставленным вопросом, задача не решена, ход решения неправильный

### **самостоятельной работы студентов**

Контроль самостоятельной работы студентов осуществляется на лекционных, практических занятиях по методикам, описанным выше.

## **7 Описание информационных технологий и материально-технической базы**

<b>Тип занятий</b>	<b>Тип помещения</b>	<b>Требования к помещению</b>	<b>Требования к программному обеспечению</b>
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше

## **8 Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

## **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

<b>№ п/п</b>	<b>Дата</b>	<b>Изменение</b>	<b>Дата и номер протокола заседания УМК</b>	<b>Автор</b>	<b>Начальник ОМОЛА</b>