

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой  
ПиИТ  
/Махортов С.Д./  
05.03.2024

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Б1.В.ДВ.05.04. Основы разработки frontend**

- 1. Код и наименование направления подготовки/специальности:**  
09.03.02 Информационные системы и технологии
- 2. Профиль подготовки/специализация:**  
Инженерия информационных систем и технологий
- 3. Квалификация выпускника:** бакалавр
- 4. Форма обучения:** очная
- 5. Кафедра, отвечающая за реализацию дисциплины:**  
Программирования и информационных технологий (ПиИТ)
- 6. Составители программы:** Махортов Сергей Дмитриевич, д.ф.-м.н., доцент
- 7. Рекомендована** НМС ФКН, протокол № 5 от 05.03.2024.

---

*отметки о продлении вносятся вручную)*

---

**8. Учебный год:** 2027 / 2028

**Семестр(ы):** 7

### 9. Цели и задачи учебной дисциплины:

- ознакомить студентов с современными версиями языка JavaScript и надстройки JSX;
- изложить основные концепции высокопроизводительного программирования части Frontend при помощи библиотеки React;
- рассмотреть реальные примеры кода и доступные для понимания блок-схемы;
- научить студентов создавать и использовать в HTML собственные компоненты React наряду с универсальными компонентами DOM;
- выработать способности и мотивацию к решению новых задач, а также практические навыки применения полученных знаний.

**10. Место учебной дисциплины в структуре ООП:** Часть, формируемая участниками образовательных отношений (вариативная) блока Б1. Требуется предварительное знание основ программирования, алгоритмических языков, алгоритмов и структур данных, архитектуры web-приложений. Предшествует дисциплинам: производственная практика, научно-исследовательская работа.

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):**

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-5	Способен выполнять работы по созданию (модификации) и сопровождению информационных систем	ПК-5.3  ПК-5.6	Обеспечивает разработку и тестирование ИС на базе типовой ИС в соответствии с требованиями. Разрабатывает и реализует алгоритмы обмена данными между ИС и существующими системами.	Знать: современные версии языка JavaScript и надстройки JSX, основы библиотеки React.  Уметь: создавать и использовать в HTML собственные компоненты React наряду с универсальными компонентами DOM.  Владеть: эффективной методикой построения части Frontend web-приложений.

**12. Объем дисциплины в зачетных единицах/час.**(в соответствии с учебным планом) – 2 / 72.

**Форма промежуточной аттестации** – зачет с оценкой.

### 13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоемкость			
	Всего	По семестрам		
		№ 7	№ семестра	...
Аудиторные занятия	48	48		
в том числе:	лекции	34	34	
	практические	-	-	
	лабораторные	16	16	
Самостоятельная работа	22	22		
в том числе: курсовая работа (проект)	-	-		
Форма промежуточной аттестации	-	-		

(зачет с оценкой – 0 час.)				
Итого:	72	72		

### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<b>1. Лекции</b>		
1.1	Введение. Стартовое приложение	Основы JavaScript. Установка React. Специальные атрибуты DOM. Расширение браузера React DevTools.
1.2	Настраиваемые компоненты	Компоненты и их свойства. Поддержка состояния. Компонент textarea. DOM-события и их обработка. Управление жизненным циклом компонента. Повышение производительности.
1.3	Табличный компонент	Источник табличных данных. Создание заголовков таблицы. Сортировка и ее индикация. Редактирование данных. Поиск в таблице. Скачивание данных.
1.4	Надстройка JSX	Простой пример JSX. Транспилатор Babel. Преобразования JSX. Основные элементы JSX. Распространяемые атрибуты. JSX и формы.
1.5	Настройки для разработки приложения	Типовое приложение React. Инструментальные средства. Сборка приложения. Развертывание.
1.6	Создание приложения React	Инициализация. Формы приложения. Основные компоненты. Диалоги. Задание стилей. Настройка приложения. Компонент <Excel>.
1.7	Проверка качества кода, тестирование	Настройка Babel в package.json. Инструмент ESLint: установка и запуск. Инструмент Flow: установка, запуск, проверка соответствия типов. Тестирование приложения.
1.8	Архитектура Flux	Основная идея хранилища. События и их использование в компонентах. Действия: CRUD, поиск и сортировка. Использование действий в компонентах. Развитие Flux: библиотека immutable. Хранилище и работа с данными в immutable.
1.9	Работа с внешними данными в React	Основы веб-запросов. Получение IP-адреса. Визуальные эффекты.
1.10	Создание планировщика в React	Построение интерфейса. Добавление и отображение элементов. Форматирование контента. Анимация.
1.11	Плавающее меню в React	Принципы работы и настройка. Отображение и сокрытие меню. Использование кнопок.
1.12	Уменьшение рендеринга	Метод render. Уменьшение количества вызовов. Анализ вызовов render. Переопределение обновления контента. Использование компонента PureComponent.
1.13	Одностраничное приложение React на основе роутера	Стартовый пример. Отображение начального фрейма. Создание страниц с контентом. Использование библиотеки React Router. Пути маршрутизации. Добавление правил CSS.
1.14	Введение в Redux	Первое знакомство с Redux. Простое приложение на основе Redux. Управление редуктором. Работа с хранилищем.
1.15	Совместное использование Redux с React	Управление состояниями React с помощью Redux. Области взаимодействия React и Redux. Создание приложения.
<b>2. Практические занятия</b>		
<b>3. Лабораторные работы</b>		
3.1	Введение. Стартовое приложение	Основы JavaScript. Установка React. Специальные атрибуты DOM. Расширение браузера React DevTools.
3.2	Настраиваемые компоненты	Компоненты и их свойства. Поддержка состояния. Компонент textarea. DOM-события и их обработка. Управление жизненным циклом компонента. Повышение производительности.
3.3	Табличный компонент	Источник табличных данных. Создание заголовков таблицы. Сортировка и ее индикация. Редактирование данных. Поиск в таблице. Скачивание данных.
3.4	Надстройка JSX	Простой пример JSX. Транспилатор Babel. Преобразования JSX. Основные элементы JSX. Распространяемые атрибуты. JSX и формы.
3.5	Настройки для разработки приложения	Типовое приложение React. Инструментальные средства. Сборка приложения. Развертывание.
3.6	Создание приложения	Инициализация. Формы приложения. Основные компоненты. Диалоги. Задание стилей. Настройка приложения. Компонент <Excel>.
3.7	Проверка качества кода, тестирование	Настройка Babel в package.json. Инструмент ESLint: установка и запуск. Инструмент Flow: установка, запуск, проверка соответствия типов. Тестирование приложения.

3.8	Архитектура Flux	Основная идея хранилища. События и их использование в компонентах. Действия: CRUD, поиск и сортировка. Использование действий в компонентах. Развитие Flux: библиотека immutable. Хранилище и работа с данными в immutable.
3.9	Работа с внешними данными в React	Основы веб-запросов. Получение IP-адреса. Визуальные эффекты.
3.10	Создание планировщика в React	Построение интерфейса. Добавление и отображение элементов. Форматирование контента. Анимация.
3.11	Плавающее меню в React	Принципы работы и настройка. Отображение и сокрытие меню. Использование кнопок.
3.12	Уменьшение рендеринга	Метод <code>render</code> . Уменьшение количества вызовов. Анализ вызовов <code>render</code> . Переопределение обновления контента. Использование компонента <code>PureComponent</code> .
3.13	Одностраничное приложение React на основе роутера	Стартовый пример. Отображение начального фрейма. Создание страниц с контентом. Использование библиотеки <code>React Router</code> . Пути маршрутизации. Добавление правил <code>CSS</code> .
3.14	Введение в Redux	Первое знакомство с Redux. Простое приложение на основе Redux. Управление редуктором. Работа с хранилищем.
3.15	Совместное использование Redux с React	Управление состояниями React с помощью Redux. Области взаимодействия React и Redux. Создание приложения.

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				Всего
		Лекции и	Практические	Лабораторные	Самостоятельная работа	
1	Введение. Стартовое приложение	2		1	2	5
2	Настраиваемые компоненты	2		1	2	5
3	Табличный компонент	2		1	2	5
4	Надстройка JSX	2		1	2	5
5	Настройки для разработки приложения	2		1	2	5
6	Создание приложения React	4		2	2	8
7	Проверка качества кода, тестирование	2		1	2	5
8	Архитектура Flux	2		1	1	4
9	Работа с внешними данными в React	2		1	1	4
10	Создание планировщика в React	2		1	1	4
11	Плавающее меню в React	2		1	1	4
12	Уменьшение рендеринга	2		1	1	4
13	Одностраничное приложение React на основе роутера	4		1	1	6
14	Введение в Redux	2		1	1	4
15	Совместное использование Redux с React	2		1	1	4
	Итого	34		16	22	72

### 14. Методические указания для обучающихся по освоению дисциплины

(рекомендации обучающимся по освоению дисциплины: работа с конспектами лекций, презентационным материалом, выполнение практических заданий, тестов, заданий текущей аттестации и т.д.)

Работа с конспектами лекций и презентационным материалом; выполнение практических заданий и тестов; выполнение лабораторных заданий; подготовка к заданиям текущей аттестации.

### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1	Стефанов С. React.js. Быстрый старт / С. Стефанов ; пер. с англ.. – СПб.: Питер, 2017. –

	304 с.
2	Чиннатамби К. Изучаем React./ К. Чиннатамби ; пер. с англ. – М.: Эксмо, 2019. – 368 с.

б) дополнительная литература:

№ п/п	Источник
3	Бэнкс А. React: современные шаблоны для разработки приложений / А. Бэнкс, Е. Порселло ; пер. с англ. – СПб: Питер, 2022. – 320 с.
4	Скотт А.Д. Разработка на JavaScript. Построение кроссплатформенных приложений с помощью GraphQL, React, React Native и Electron / А.Д. Скотт ; пер. с англ. – СПб: Питер, 2021. – 320 с..
5	Мардан А. React быстро. Веб-приложения на React, JSX, Redux и GraphQL / А. Мардан ; пер. с англ. – СПб: Питер, 2019. – 560 с..

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)\*:

№ п/п	Ресурс
6	<a href="http://www.lib.vsu.ru">www.lib.vsu.ru</a> – ЗНБ ВГУ
7	<a href="http://www.cs.vsu.ru/msd">http://www.cs.vsu.ru/msd</a>

\* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы

**16. Перечень учебно-методического обеспечения для самостоятельной работы** (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)

№ п/п	Источник

**17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение)**

Для реализации учебного процесса используются: бесплатные библиотеки и полнофункциональные инструментальные системы React, Redux, ESLint, Flow, Visual Studio Code; ресурс «Электронный университет» (<https://edu.vsu.ru/>).

**18. Материально-техническое обеспечение дисциплины**

1. Компьютерный класс №5 (ауд. 295). ПК-Intel-Core2 14 шт., рабочее место преподавателя: проектор, видеокоммутатор, специализированная мебель: доска маркерная 1 шт., столы 16 шт., стулья 33 шт. В классе находится точка доступа беспроводной сети для доступа в Интернет и к учебно-методическим материалам, расположенным на внутренних серверах факультета.

2. Компьютерный класс №7 (ауд. 316п). ПК на базе IntelCore2Duo 2,8ГГц, ОЗУ 2ГБ, диск 160Gb – 30 шт. Специализированная мебель: доска маркерная 1 шт., доска интерактивная 1 шт., столы 32 шт., стулья 64 шт.; рабочее место преподавателя: проектор, видеокоммутатор. В классе находится точка доступа беспроводной сети для

доступа в Интернет и к учебно-методическим материалам, расположенным на внутренних серверах факультета.

## **19. Оценочные средства для проведения текущего контроля и промежуточной аттестаций**

### **19.1. Текущий контроль**

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в форме письменных работ (контрольные, выполнение практико-ориентированных заданий). Критерии оценивания приведены ниже.

<i>Критерии оценивания</i>	<i>Шкала оценок (в баллах)</i>
Обучающийся отвечает на все вопросы правильно.	3
Обучающийся отвечает на все вопросы. Допускаются незначительные неточности.	2
Обучающийся отвечает не на все вопросы. Ответ не содержит грубых ошибок.	1
Обучающийся отвечает не на все вопросы. Присутствуют грубые ошибки.	0

### **19.2. Промежуточная аттестация**

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний и практические задания, позволяющие оценить степень сформированности умений и навыков.

Для оценивания результатов обучения на зачете с оценкой используются следующие показатели: владение теоретическими основами дисциплины, способность иллюстрировать ответ примерами, применять теоретические знания для решения практических задач.

Для оценивания результатов обучения на зачете применяется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Используются следующие показатели и их соотношения:

- уверенное владение теоретическими основами дисциплины, способность применять теоретические знания для решения практических задач, результаты выполнения всех заданий лабораторного практикума – «отлично»;
- хорошее владение теоретическими основами дисциплины, способность применять теоретические знания для решения практических задач, результаты выполнения большинства заданий лабораторного практикума – «хорошо»;
- неполное владение теоретическими основами дисциплины, затруднения в применении теоретических знаний для решения практических задач, результаты выполнения не менее 30% заданий лабораторного практикума – «удовлетворительно»;
- слабое владение теоретическими основами дисциплины, неспособность применять теоретические знания для решения практических задач, результаты

выполнения менее 30% заданий лабораторного практикума – «неудовлетворительно».

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: собеседование по зачетным билетам (КИМ). Перечень вопросов к зачету:

1. Введение. Стартовое приложение
2. Настраиваемые компоненты
3. Табличный компонент
4. Надстройка JSX
5. Настройки для разработки приложения
6. Создание приложения React
7. Проверка качества кода, тестирование
8. Архитектура Flux
9. Работа с внешними данными в React
10. Создание планировщика в React
11. Плавающее меню в React
12. Уменьшение рендеринга
13. Одностраничное приложение React на основе роутера
14. Введение в Redux
15. Совместное использование Redux с React