

Министерство науки и высшего образования Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Научно-образовательный центр «Высшая ИТ школа»

УТВЕРЖДЕНО:  
Исполнительный директор НОЦ ВИТШ

Т.С.Кетова

Рабочая программа дисциплины

**Web-разработка приложений  
(Web-разработка)**

по направлению подготовки  
**09.03.04 (33.04) Программная инженерия**

Направленность подготовки:  
**«Программная инженерия»**

Форма обучения  
**Очная**

Квалификация  
**Программный инженер**

Год приема  
**2023**

СОГЛАСОВАНО:  
Руководитель ОП  
О.А.Змеев

Председатель УМК  
Д.О. Змеев

## 1. Цель и планируемые результаты освоения дисциплины (модуля)

Целью освоения дисциплины является формирование следующих компетенций и результатов обучения:

БК-1	Способен применять общие и специализированные компьютерные программы при решении задач профессиональной деятельности	Знает: правила и принципы применения общих и специализированных компьютерных программ для решения задач профессиональной деятельности Умеет: применять современные ИТ-технологии для сбора, анализа и представления информации; использовать в профессиональной деятельности общие и специализированные компьютерные программы
БК-7	Способен создавать программное обеспечение на основе предоставленных требований к дизайну, функциональности, безопасности, использованием стандартных подходов, библиотек, инструментов контроля версий	Знает: Принципы оценки характеристик программного обеспечения. Правила, языки и методы фиксации требований к программному обеспечению, архитектуре или ожидаемому поведению, на уровне чтения технической и проектной документации. Основные принципы безопасной разработки приложений. Теорию использования и работы с хранилищами данных, протоколами передачи данных, алгоритмы обработки и кодирования данных Умеет: Модифицировать программный код приложения с целью исправления дефектов, расширения функциональности, или при изменении требований к приложению; разрабатывать программное обеспечение совместно с другими членами команды разработки; использовать инструменты для совместной разработки и развертывания приложений; проверять приложение на дефекты; проверять критерии готовности приложения для практического использования или передачи другой команде разработчиков; оценивать требования запущенного приложения к среде его функционирования и развертывания
ОПК-3	Способен качественно разрабатывать элементы программной системы, не имея полной спецификации и учитывая аспекты, связанные с безопасностью, поддерживаемостью и эффективностью работы элемента.	Знает: Правила трансляции артефактов проектирования в программный код. Основные правила и особенности используемого стека технологий. Парадигму используемого языка программирования. Паттерны, связанные с парадигмой используемого языка программирования. Основные паттерны используемого языка программирования Особенности процессов компиляции/выполнения программного кода Основу правил безопасной разработки. Основу принципов оптимизации вычислений.

Умеет:

Определять структуру отдельных частей элемента системы для имплементации требуемой логики к элементу;

решать технические вопросы, связанные с особенностями используемого стека технологий или языка программирования;

проверять работоспособность написанного кода;

совместно разрабатывать программный код при командной разработке общих компонентов системы

ОПК-4	Способен использовать технологии, методы, инструменты и процессы выявления и предотвращения дефектов в соответствии установленными требованиями к качеству.	Знает: Концепции тестирования методом «чёрного ящика», методом «белого ящика». Структуру и основные правила написания тест-кейсов. Основные виды и принципы тестов, таких как unit-тесты, автотесты, интеграционные тесты, имитационные тесты, нагрузочные тесты. Концепцию использования автотестов при разработке с продуктами. Техники тест-дизайна. Разницу между атрибутами качества: - заметными во время выполнения (производительность, безопасность, доступность, функциональность, удобство использования) - не заметными во время выполнения (модифицируемость, переносимость, возможность повторного использования, интегрируемость и тестируемость) - связанными с внутренними качествами архитектуры и детального проектирования (концептуальная целостность, правильность и полнота). Умеет: Выполнять проверку программного обеспечения следуя написанному тест-кейсу; выполнять интеграционное тестирование и анализ компонентов программного обеспечения с использованием методов «черного ящика» и «сценария использования» в сотрудничестве со стейкхолдерами; оценить тестируемость программного обеспечения с учетом атрибутов качества при проектировании/реализации подсистем и модулей; применять хотя бы одну технику тест-дизайна для создания тест-кейсов программного обеспечения
-------	---	--

## 2. Задачи освоения дисциплины

– Освоить аппарат разработки современных клиентских приложений с использованием библиотеки React и принципы разработки Single Page Application.

– Научиться применять понятийный аппарат разработки клиентских приложений для решения практических задач профессиональной деятельности.

### **3. Место дисциплины (модуля) в структуре образовательной программы**

Web-разработка приложений (Web-разработка)\* - дисциплина относится к Блоку 1 «Дисциплины (модули)» части образовательной программы, формируемой участниками образовательных отношений, предлагается обучающимся на выбор.

\*Для внесения оценки за дисциплину в зачетную книжку обучающегося принимается сокращенное название «Web-разработка».

### **4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине**

Семестр 4, экзамен.

### **5. Входные требования для освоения дисциплины**

Для успешного освоения дисциплины требуются компетенции, сформированные в ходе освоения образовательных программ предшествующего уровня образования.

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: Программирование (основы) 1, Программирование (основы) 2.

### **6. Язык реализации**

Русский

### **7. Объем дисциплины (модуля)**

Общая трудоемкость дисциплины составляет 5 з.е., 180 часов, из которых:

– лекции: 16 ч.;

– лабораторные работы: 32 ч.

в том числе практическая подготовка: 93 ч.

Объем самостоятельной работы студента определен учебным планом.

### **8. Содержание дисциплины (модуля), структурированное по темам**

Тема 1. Введение в одностраничные приложения

Разбор основ одностраничных приложений, базовые отличия от многостраничных приложений.

Тема 2. Введение в React

История фреймворка, установка и настройка Node.js сервера, создание и конфигурация проекта, базовые понятия React.

Тема 3. Продвинутая разработка с использованием React

Изучение глобальных хранилищ данных, библиотека Redux, redux-toolkit, работа с API средствами библиотеки Axios, изучение базовых хуков, построение многослойной архитектуры, работа с формами.

### **9. Текущий контроль по дисциплине**

Текущий контроль по дисциплине осуществляется путем контроля посещаемости, выполнения домашних заданий, и фиксируется в форме контрольной точки не менее одного раза в семестр. Оценочные материалы текущего контроля размещены на сайте ТГУ в разделе «Информация об образовательной программе» – <https://www.tsu.ru/sveden/education/eduop/>.

### **10. Порядок проведения и критерии оценивания промежуточной аттестации**

Экзамен во четвертом семестре проводится в устной форме в формате защиты разработанного проекта. Продолжительность экзамена 1,5 часа.

Оценочные материалы для проведения промежуточной аттестации размещены на сайте ТГУ в разделе «Информация об образовательной программе» – <https://www.tsu.ru/sveden/education/eduop/>.

## 11. Учебно-методическое обеспечение

а) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине (<https://www.tsu.ru/sveden/education/eduop/>).

б) Видеоматериалы по базовому и продвинутому курсу Frontend - <https://youtube.com/playlist?list=PLnV4Cn5BneX8lQ0804ivGFnj8gjt4Aiu&si=EwLSRPb-ENGYyKT>

## 12. Перечень учебной литературы и ресурсов сети Интернет

а) основная литература:

1. Learning React: Functional Web Development with React and Redux, by Alex Banks & Eve Porcello;
2. React in Action by Mark Tielens, Thomas;

б) ресурсы сети Интернет:

1. База знаний HTML - <https://htmlreference.io/>
2. База знаний CSS - <https://cssreference.io/>
3. JavaScript - <https://developer.mozilla.org/ru/docs/Web/JavaScript>
4. React docs - <https://legacy.reactjs.org/docs/getting-started.html>

## 13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:

– Microsoft Office Standart 2013 Russian: пакет программ. Включает приложения: MS Office Word, MS Office Excel, MS Office PowerPoint, MS Office On-eNote, MS Office Publisher, MS Outlook, MS Office Web Apps (Word Excel MS PowerPoint Outlook);

– публично доступные облачные технологии (Google Docs, Яндекс диск и т.п.).

б) информационные справочные системы:

– Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system>

– Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>

## 14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

Аудитории для проведения занятий лекционного и семинарского типа индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации в смешенном формате («Актру»).

## 15. Информация о разработчиках

Скиба Виктор Евгеньевич,  
старший преподаватель учебного офиса  
НОЦ «Высшая ИТ школа»,  
программист отдела разработки цифровых  
решений управления цифровых решений