ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол №2

Рабочая программа дисциплины (модуля)
«Технологии фронтенд-разработки»

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Разработка*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2025*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2025

Содержание

1. **Краткая характеристика дисциплины (модуля)** 3
2. **Перечень планируемых результатов обучения** 6
3. **Тематический план** 9
4. **Содержание дисциплины (модуля)** 10
5. **Учебно-методическое обеспечение** 11
6. **Материально-техническое обеспечение** 11
7. **Методические и оценочные материалы** 13

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Технологии фронтенд-разработки» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 «Математика и компьютерные науки», профиль «Разработка», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Технологии фронтенд-разработки» формирует прочную фундаментальную базу для успешной работы в современной веб-разработке, позволяя глубоко понимать принципы работы браузера, языков и инструментов. Полученные знания и навыки обеспечивают возможность эффективного освоения любых фронтенд-фреймворков и создания масштабируемых, производительных и поддерживаемых веб-приложений.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 «Математика и компьютерные науки», профиль «Разработка» и входит в обязательную часть Блока 1, как дисциплина по выбору.

Дисциплина (модуль) изучается на 2 и 3 курсе в 3 или 5 семестре на выбор. Доступна к изучению после успешного освоения одной из дисциплины (модуля): «Основы промышленной разработки».

Цель изучения дисциплины (модуля): заключается в формировании системного понимания веб-платформы и навыков создания современных, производительных, доступных и поддерживаемых веб-интерфейсов с использованием HTML, CSS, JavaScript и TypeScript без зависимости от фреймворков.

Задачи изучения дисциплины (модуля):

— изучить принципы работы веб-платформы, включая HTTP/HTTPS, DNS, клиент-серверное взаимодействие, механизмы безопасности (CORS, cookies) и процессы кэширования;

— освоить современные подходы к верстке с использованием семантического HTML, адаптивного дизайна, Flexbox, Grid и практик доступности (a11y) и SEO;

— развить навыки программирования на JavaScript и TypeScript, включая работу с DOM, асинхронным кодом, модулями и современными языковыми возможностями;

— научиться проектировать и реализовывать одностраничные приложения (SPA) на ванильном TypeScript с клиентским роутингом и типобезопасной работой с API;

— освоить инструментальную экосистему фронтенд-разработки, включая Node.js, npm, Vite, линтеры, форматтеры и инструменты анализа производительности (Lighthouse, Web Vitals).

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— принципы работы веб-платформы: HTTP/HTTPS, DNS, клиент-серверное взаимодействие, cookies, кэширование, CORS;

— назначение и возможности инструментов разработчика браузера, включая анализ HTML, CSS, сетевых запросов и ошибок JavaScript;

— основы семантической HTML-разметки, форм, доступности, SEO и структуры веб-страницы;

— основные принципы CSS: каскад, специфичность, box model, адаптивная вёрстка, Flexbox, Grid, анимации и CSS-переменные;

— базовые и продвинутые особенности JavaScript: типы данных, функции, области видимости, замыкания, прототипы, объекты, классы и контекст this;

— принципы работы DOM API, событийной модели браузера и рендеринга страницы;

— механизмы асинхронного программирования в JavaScript: event loop, callbacks, Promise, async/await, Fetch API;

— современные возможности JavaScript: модули, итераторы, генераторы, Map, Set, Symbol, Proxy, Reflect;

— основы инструментальной среды фронтенд-разработки: Node.js, npm, Vite, сборка проекта, линтеры и форматтеры;

— основы TypeScript: статическая типизация, компиляция, tsconfig, интерфейсы, типы, объединения, сужение типов и структурная типизация;

— продвинутые возможности TypeScript: дженерики, utility types, mapped types, conditional types и типобезопасная работа с данными;

— базовые архитектурные подходы и паттерны организации фронтенд-кода без использования фреймворков;

— принципы построения SPA на ванильном TypeScript, клиентского роутинга, загрузки модулей и оптимизации производительности;

— основные метрики производительности веб-приложений и подходы к их анализу с использованием Lighthouse и Web Vitals.

уметь:

— анализировать работу веб-страниц и сетевых запросов с помощью DevTools;

— создавать семантическую, доступную и валидную HTML-разметку;

— верстать адаптивные интерфейсы с использованием современных возможностей CSS;

— работать с макетом в Figma и переносить базовые визуальные параметры в код;

— писать JavaScript-код с учётом особенностей языка, областей видимости, замыканий, прототипов и контекста выполнения;

— работать с DOM: создавать, изменять и удалять элементы, обрабатывать события, управлять формами и интерактивностью страницы;

— реализовывать асинхронную логику с использованием Promise, async/await, Fetch API и AbortController;

— использовать современные структуры и возможности JavaScript для решения прикладных задач;

— создавать и настраивать фронтенд-проект с использованием Node.js, npm, Vite, ESLint и Prettier;

— переводить JavaScript-код на TypeScript и типизировать данные, функции, DOM-операции и ответы внешних API;

— использовать дженерики и utility types для построения типобезопасных абстракций;

— организовывать код фронтенд-приложения по модулям и применять базовые архитектурные паттерны;

— проектировать и реализовывать SPA на ванильном TypeScript;

— работать с внешними API, обрабатывать JSON-данные и ошибки сетевого взаимодействия;

— анализировать производительность веб-приложения и применять базовые методы оптимизации загрузки и рендеринга.

владеть:

— инструментами анализа и отладки веб-приложений в браузере;

— навыками разработки адаптивных, доступных и интерактивных веб-интерфейсов;

— навыками уверенного программирования на JavaScript в браузерной среде;

— методами работы с DOM, событиями, формами, браузерными API и асинхронными запросами;

— подходами к написанию поддерживаемого JavaScript/TypeScript-кода;

— инструментами современной фронтенд-разработки: Node.js, npm, Vite, ESLint, Prettier;

— навыками типизации фронтенд-приложений с использованием TypeScript;

— методами построения типобезопасных функций, моделей данных и обёрток для работы с API;

— подходами к модульной организации кода и применению архитектурных паттернов во фронтенд-разработке;

— навыками построения одностраничного приложения на ванильном TypeScript без использования фреймворков;

— методами анализа и оптимизации производительности веб-приложений;

— навыками чтения, анализа, рефакторинга и объяснения фронтенд-кода.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализация математических моделей в профессиональной деятельности
ОПК-2.	Способен проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности	ОПК-2.1.	Знает основные методы и подходы к проведению исследований в области разработки; научные и этические стандарты, применяемые в исследовательской практике
		ОПК-2.2.	Умеет формулировать исследовательские вопросы и гипотезы на основе существующих знаний; анализировать и интерпретировать данные, полученные в ходе исследования
		ОПК-2.3.	Имеет практический опыт исследований в конкретной области профессиональной деятельности в области разработки
ОПК-4.	Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	ОПК-4.1.	Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности

		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности
		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ОПК-5.	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-5.1.	Знаете технологии, необходимые для прикладного программирования, включая современные функциональные языки программирования, а также основные принципы и понятия, применяемыми при использовании компьютерных сетей
		ОПК-5.2.	Умеет пользоваться технологиями прикладного программирования, включая среды высокоуровневого программирования
		ОПК-5.3.	Имеет практический опыт использования технологий прикладного программирования
ПК-1.	Способен формулировать задачи с математической точностью, обосновывать утверждения строго и анализировать полученные результаты в области математики и компьютерных наук	ПК-1.1.	Обладает базовыми знаниями, полученными в области математических наук, программирования и информационных технологий
		ПК-1.2.	Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и информатике
		ПК-1.3.	Имеет опыт работы с задачами в области математики и компьютерных наук, включая применение математических методов для решения практических задач
ПК-3	Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты
		ПК-3.3.	Имеет практический опыт использования методов математического и алгоритмического моделирования в

			реальных проектах или исследованиях

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары			
1	Веб-платформа		12		40	Домашние задания
2	JavaScript в браузере		20	4	44	Домашние задания, Контрольная работа
3	TypeScript		20	4	44	Домашние задания, Контрольная работа
	Экзамен			2		
	Итого		***52***	***10***	***128***	
Объем дисциплины (модуля) (в ак. ч.)		***190***				
Объем дисциплины (модуля) (в зач. ед.)		***5***				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Веб-платформа	Как работает веб HTML: семантика, формы, доступность Таблицы стилей CSS
2	JavaScript в браузере	JavaScript: основы языка JavaScript: объекты, прототипы, ООП DOM API и работа с браузером JavaScript: асинхронность JavaScript: продвинутые возможности
3	TypeScript	Инструментальная среда фронтенд-разработки TypeScript: основы типизации и компиляция TypeScript: дженерики и утилитарные типы Архитектура и паттерны SPA на ванильном TypeScript, производительность и оптимизация

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Хорстманн, К. С. Современный JavaScript для нетерпеливых : практическое руководство / К. 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/2109482.

2. Черный, Б. Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений / Б. Черный. - Санкт-Петербург : Питер, 2021. - 352 с. - ISBN 978-5-4461-1651-5. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2142477.

Дополнительная литература:

1. Вандеркам, Д. Эффективный TypeScript: 62 способа улучшить код : практическое руководство / Д. Вандеркам. - Санкт-Петербург : Питер, 2021. - 288 с. - (Серия «Бестселлеры O'Reilly»). - ISBN 978-5-4461-1623-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/1733517.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том

числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое

CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Технологии фронтенд-разработки» в рамках текущего контроля успеваемости используются такие виды учебной работы, как семинары, домашние задания и контрольные работы, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Технологии фронтенд-разработки»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***экзамена***, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	**Пятибалльная оценка**	**Общая характеристика результата обучения по дисциплине (модулю)**
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если
2	Не сдан	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
1	Не сдан	преподаватель задает дополнительные наводящие вопросы.

Дисциплина (модуль) «Технологии фронтенд-разработки» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашнее задание	40%	13	Набор задач по темам недели
Контрольная работа	30%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
Экзамен	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по курсу

Формула расчёта итоговой оценки по дисциплине (модулю) «Технологии фронтенд-разработки»: «0,4 × среднее за домашние задания + 0,3 × среднее за контрольные работы + 0,3 × экзамен».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1.

1. Объясните, как работает веб: из каких этапов состоит запрос к веб-сайту, начиная с ввода URL и заканчивая отображением страницы в браузере. Укажите роль DNS, HTTP, HTML, CSS.
2. Создайте семантическую HTML-разметку для страницы новостного сайта. Используйте теги <header>, <nav>, <main>, <article>, <aside>, <footer>. Объясните, зачем нужна семантика и как она влияет на доступность и SEO.
3. Добавьте на страницу форму обратной связи с полями: имя, email, сообщение. Укажите правильные атрибуты (type, required, placeholder) и объясните, как форма обеспечивает доступность (например, через <label> и aria-* атрибуты).
4. Напишите CSS-стили для созданной страницы: задайте цвета, шрифты, отступы. Используйте каскад и специфичность, чтобы продемонстрировать, как браузер выбирает стиль для элемента.
5. Сделайте страницу адаптивной: используйте медиазапросы, чтобы изменить компоновку на мобильных устройствах. Примените Flexbox или Grid для построения сетки. Объясните, как работает адаптивная вёрстка.

Домашнее задание 2.

1. Напишите JavaScript-код, который при загрузке страницы добавляет заголовок <h1> в <main> с текстом "Добро пожаловать!". Объясните, как работает DOM API и как найти/создать/вставить элементы.
2. Создайте объект user с полями name, age, isActive. Напишите функцию, которая принимает такой объект и возвращает строку вида "Пользователь: Имя, возраст: 25, активен: да/нет". Используйте прототипы, чтобы добавить метод getInfo() к объекту.
3. Найдите на странице кнопку и привяжите к ней обработчик события click, который выводит сообщение в консоль. Добавьте возможность отключить обработчик после трёх нажатий. Объясните, как работает событийная модель браузера.

4. Напишите асинхронную функцию, которая с помощью fetch получает данные с публичного API (например, https://jsonplaceholder.typicode.com/users). Выведите имена пользователей на страницу. Используйте async/await и обработайте возможные ошибки.

5. Создайте итератор для массива пользователей, который возвращает одного пользователя за раз. Реализуйте простой генератор, который фильтрует активных пользователей. Объясните разницу между итераторами и генераторами.

Домашнее задание 3.

1. Установите Node.js и инициализируйте проект с помощью npm init. Настройте Vite для работы с TypeScript. Добавьте ESLint и Prettier. Опишите, зачем нужны эти инструменты и как они улучшают разработку.

2. Перепишите код из второго домашнего задания на TypeScript: определите интерфейс User с типами полей, типизируйте функции и ответ API. Объясните, как статическая типизация помогает избежать ошибок.

3. Создайте дженерик-функцию filterByProperty<T>(array: T[], key: keyof T, value: any), которая фильтрует массив по свойству. Используйте utility types, например Partial<User>, чтобы типизировать форму редактирования. Объясните преимущества дженериков и утилитарных типов.

4. Реализуйте простой одностраничный роутер на ванильном TypeScript: при изменении хэша в URL (#home, #about) отображается соответствующий контент. Используйте модули для разделения логики. Опишите принцип работы клиентского роутинга.

5. Проанализируйте производительность своей страницы с помощью Lighthouse. Оцените метрики Web Vitals (FCP, LCP, CLS). Предложите 3 способа оптимизации загрузки и рендеринга (например, ленивая загрузка, минификация, кэширование).

Примерные задания для контрольной работы
Контрольная работа 1.

1. Напишите краткое описание основных особенностей языка JavaScript: типы данных, области видимости, замыкания, прототипное наследование. Объясните, чем отличается var, let, const.

2. Создайте объект Product с полями name, price, inStock. Добавьте метод getInfo() через прототип. Реализуйте функцию-конструктор или класс для создания нескольких экземпляров. Объясните, как работает прототипная цепочка.

3. Напишите JavaScript-код, который при загрузке страницы находит элемент по ID и изменяет его содержимое. Добавьте обработчик события click, который скрывает или показывает другой элемент. Объясните, как работает DOM API и события.

4. Реализуйте функцию, которая с помощью fetch получает данные с API (например, список пользователей). Используйте async/await, обработайте ошибки. Выведите результат на страницу. Объясните, как работает event loop и асинхронность в JavaScript.

5. Создайте итератор для массива товаров, который возвращает по одному элементу за раз. Реализуйте генератор, который фильтрует товары по цене. Объясните разницу между итераторами и генераторами.

6. Напишите функцию, которая использует замыкание для создания счётчика. Объясните, как замыкания помогают инкапсулировать данные и почему они важны в JavaScript.

Контрольная работа 2.

1. Опишите основные преимущества использования TypeScript по сравнению с JavaScript. Объясните, как работает компиляция, tsconfig.json и статическая типизация.

2. Создайте интерфейс User с полями id, name, email, isActive. Напишите функцию createUser, принимающую объект типа User. Передайте в неё объект с лишним полем — объясните, как TypeScript обработает это, и как включить строгую проверку.

3. Реализуйте дженерик-функцию filterBy<T>(array: T[], key: keyof T, value: any), которая фильтрует массив по свойству. Используйте Partial<T>, Pick<T> или Omit<T> для типизации формы редактирования. Объясните, зачем нужны утилитарные типы.

4. Настройте простой проект с помощью Vite и TypeScript. Добавьте ESLint и Prettier. Опишите, зачем нужны эти инструменты и как они улучшают качество кода и командную разработку.

5. Реализуйте клиентский роутинг на ванильном TypeScript: при изменении # в URL (#home, #profile) отображается соответствующий контент. Используйте модули для разделения логики. Объясните принцип работы SPA без фреймворков.

6. Опишите один из архитектурных паттернов (например, Module Pattern, Observer, Repository). Покажите пример его использования во фронтенд-приложении. Объясните, как паттерны улучшают структуру кода.

7. Проанализируйте производительность веб-приложения с помощью Lighthouse. Оцените метрики Web Vitals (LCP, FID, CLS). Предложите три способа оптимизации: ленивая загрузка, минификация, кэширование. Объясните, как каждое улучшение влияет на пользовательский опыт.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Укажите протокол прикладного уровня, используемый для передачи гипертекста в сети Интернет.	HTTP / HTTPS	ОПК-5
2.	Как называется распределенная система, преобразующая доменные имена в IP-адреса?	DNS	ОПК-5
3.	Укажите стандартный сетевой порт для защищённого HTTPS-соединения.	443	ОПК-5
4.	Как называется механизм сохранения копий веб-ресурсов для ускорения повторных запросов?	кэширование / caching	ОПК-5
5.	Укажите семантический HTML-тег, обозначающий основное содержимое документа.	**<main>**	ОПК-5
6.	Какой атрибут HTML связывает элемент **<label>** с соответствующим элементом формы?	for	ОПК-5
7.	Укажите CSS-свойство, задающее внешний отступ элемента относительно соседей.	margin	ОПК-5
8.	Как называется модель компоновки CSS, основанная на гибких контейнерах и осях?	Flexbox / flex	ОПК-5
9.	Укажите тип данных в JavaScript, к которому относятся функции.	function / object	ОПК-4
10.	Как называется конструкция в JavaScript, позволяющая функции запоминать лексическое окружение?	замыкание / closure	ОПК-4

11.	Укажите метод DOM API для поиска элемента по его уникальному идентификатору.	getElementById	ОПК-4
12.	Как называется событие, возникающее после полной загрузки и разбора HTML-документа?	DOMContentLoaded	ОПК-4
13.	Укажите современный интерфейс JavaScript для выполнения сетевых запросов.	Fetch API / fetch	ОПК-4
14.	Как называется механизм, обеспечивающий обработку событий и асинхронных задач в JavaScript?	event loop / цикл событий	ОПК-4
15.	Укажите синтаксическую конструкцию TypeScript для явного указания типа переменной.	**: Type** / аннотация типа	ПК-1
16.	Как называется утилитарный тип TypeScript, делающий все свойства интерфейса необязательными?	Partial<T>	ПК-1
17.	Укажите современный инструмент сборки, обеспечивающий быструю разработку на TypeScript.	Vite / Webpack	ОПК-5
18.	Как называется архитектурный подход, разделяющий код на независимые логические модули?	модульность / модульная архитектура	ПК-1
19.	Укажите метрику Web Vitals, измеряющую время отрисовки largest contentful элемента.	LCP / Largest Contentful Paint	ПК-3
20.	Как называется техника оптимизации, при которой ресурсы загружаются только по мере необходимости?	lazy loading / ленивая загрузка	ПК-3