

Утверждаю
Генеральный директор ООО
«ИТ КОНСАЛТ»
Тетеревлев Игорь Николаевич
«07» сентября 2023 г.

**Дополнительная программа
повышения квалификации
«Веб программирование»**

Срок реализации: 3 месяца (240 часов)

**Автор-составитель:
Тетеревлев Игорь Николаевич**

**г. Таганрог
2023 г**

СОДЕРЖАНИЕ

1. Лист обновлений и актуализации.....	3
2. Общие положения.....	6
3. Цели и задачи программы.....	8
4. Планируемые результаты обучения.....	9
5. Учебный план.....	11
6. Календарный учебный график.....	17
7. Рабочая программа.....	19
8. Материально-техническое обеспечение.....	26
9. Оценочные материалы.....	27
10. Итоговая аттестация.....	41
11. Организационно-педагогические условия.....	45
12. Список использованной литературы.....	46

ЛИСТ ОБНОВЛЕНИЙ И АКТУАЛИЗАЦИИ

дополнительной профессиональной программы
повышения квалификации
«Веб программирование»

№ п/ п	Дата внесения изменений в программу	Характер изменений	Дата и номер протокола согласований документа
1	01.09.2023	Расширен модуль алгоритмика для глубокого погружения в современные методы решения задач. Были включены 20 новых задач, охватывающих различные темы — от основных структур данных до сложных алгоритмов поиска и сортировки. Каждая задача сопровождается детальным решением, позволяя слушателям разбираться в мельчайших деталях алгоритмов. Для контроля качества решения каждой задачи предоставляются тестовые кейсы, которые помогут убедиться в корректности реализации. Кроме того, для удобства студентов к каждой задаче прилагается развернутый ответ, что обеспечивает возможность самоконтроля и повышает эффективность обучения.	№1 от 01.09.2023
2	01.09.2023	Добавлены подробные ответы для 50 самых популярных вопросов на интервью в рамках модуля Реакт. Проработаны детальные ответы на все 50 вопросов, включая примеры практического использования	№2 от 01.09.2023

3	01.09.2023	Введен усовершенствованный модуль "CSS верстка", акцентирующий внимание на последних трендах и лучших практиках в области веб-дизайна. Этот модуль затрагивает все ключевые аспекты создания адаптивных и кроссбраузерных интерфейсов: от основных принципов структурирования и оформления контента до использования современных техник и инструментов, таких как Flexbox и Grid Layout.	№3 от 01.09.2023
4	07.09.2023	Внедрен модуль по работе с современными инструментами сборки, такими как Webpack и Babel. Охвачены основные аспекты конфигурации и оптимизации для улучшения производительности веб-приложений.	№4 от 07.09.2023
5	07.09.2023	Запущен модуль "TypeScript для веб-разработчиков". Рассмотрены основы строгой типизации, внедрение TypeScript в существующие проекты и основные практические примеры использования.	№5 от 07.09.2023
6	07.09.2023	Обновлен раздел "Адаптивная и мобильная верстка". Добавлены практические задания и материалы, позволяющие студентам разрабатывать сайты, которые корректно отображаются на всех устройствах.	№6 от 07.09.2023
7	07.09.2023	Интегрирован раздел "Понимание Event Loop в JavaScript". В этом разделе представлено детальное изучение принципов работы асинхронности в JS, начиная с	№7 от 07.09.2023

		основных понятий и заканчивая глубоким погружением в механизмы Event Loop. Слушатели познакомятся с порядком выполнения макро- и микро задач, ролью стека вызовов и очереди событий. Через практические примеры и визуализацию демонстрируется, как JavaScript обрабатывает асинхронные операции	
8	07.09.2023	Внедрен модуль "Promises в JavaScript". Их роли в асинхронном программировании и практическому применению в современных веб-приложениях. Слушатели изучат жизненный цикл промиса, особенности методов .then(), .catch(), и .finally(), а также углубленно рассмотрят вопросы обработки ошибок. Через серию практических заданий студенты научатся создавать свои промисы, использовать Promise.all(), Promise.race() и другие полезные методы, а также применять промисы в реальных сценариях взаимодействия с веб-API и другими асинхронными операциями.	№8 от 07.09.2023

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1 Характеристика программы:

Дополнительная профессиональная программа повышения квалификации «**Веб программирование**» (далее – программа) является учебно-методическим нормативным документом, регламентирующим содержание, организационно-методические формы и трудоемкость обучения.

Дополнительная профессиональная программа повышения квалификации «**Веб программирование**» разработана в соответствии с нормами Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» с учетом требований приказа Минобрнауки России от 28.07.2014 N 804 (ред. от 21.10.2019) «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 программирование в компьютерных системах», требований Приказа от 18 января 2017 года № 44н «Об утверждении профессионального стандарта "Разработчик Web и мультимедийных приложений».

1.2. Характеристика профессиональной деятельности слушателей:

Слушатели программы "Веб-программирование" призваны овладеть современными методами и технологиями разработки веб-приложений. Их профессиональная деятельность включает в себя проектирование, разработку, тестирование и поддержку веб-сайтов и веб-приложений разного уровня сложности. Они способны работать с различными веб-фреймворками, библиотеками и языками программирования, а также применять лучшие практики и стандарты в области веб-разработки.

Основная цель вида профессиональной деятельности слушателей:

Основная цель слушателей программы "Веб-программирование" - приобретение глубоких теоретических знаний и практических навыков в области создания и оптимизации современных веб-приложений, чтобы стать квалифицированными специалистами, готовыми к решению широкого круга задач в сфере веб-разработки.

1.3. Категория обучающихся: лица, имеющие среднее профессиональное и (или) высшее образование; лица, получающие среднее профессиональное и (или) высшее образование.

1.4. Объем программы (трудоемкость): общая трудоемкость 240

академических часов.

1.5. Срок освоения программы — 3 месяца.

1.6. Форма обучения: очная, сочетание теоретических и практических занятий.

1.7. Режим занятий:

- учебные занятия проводятся 5 дней в неделю
- количество учебных часов в неделю – 20 часов.

1.8 Промежуточная аттестация:

Проводится по результатам освоения каждого модуля в форме проведения контрольной или практической работы, по критерию «аттестован»/«не аттестован».

Обучающийся считается аттестованным, если он успешно освоил все три модуля и получил положительные результаты при контрольной работе (или тестировании) по каждому из них.

Допуск к сдаче итогового экзамена осуществляется на основании положительного результата промежуточной аттестации.

1.9. Итоговая аттестация

Обучение завершается обязательной итоговой аттестацией в виде письменного экзамена.

Сдача экзамена входит в нормативный срок освоения программы. По результатам выполнения итогового экзамена и проекта по веб-программированию выставляется итоговая оценка.

По результатам прохождения экзамена выдается удостоверение о повышении квалификации в области веб-программирования установленного образца.

2. ЦЕЛИ И ЗАДАЧИ ПРОГРАММЫ

2.1. Цель программы заключается в совершенствовании имеющихся и приобретении новых профессиональных компетенций, необходимых для профессиональной деятельности в области веб-программирования, разработки современных веб-приложений, а также углубленного изучения функциональных веб возможностей

2.2. Задачи программы:

- 1) Изучить основные принципы и методы веб-программирования, включая структуру и дизайн веб-сайтов.
- 2) Освоить популярные языки программирования для веба, такие как HTML, CSS, JavaScript и другие, и их применение в реальных проектах.
- 3) Разработать навыки по работе с современными веб-фреймворками и библиотеками для создания динамических и адаптивных веб-приложений.
- 4) Изучить принципы и методы оптимизации веб-сайтов для различных устройств и браузеров, обеспечивая лучшую производительность и удобство пользовательского интерфейса.
- 5) Познакомиться с лучшими практиками и стандартами веб-разработки, чтобы гарантировать высокое качество создаваемых проектов.
- 6) Развивать навыки работы в команде, осваивать принципы Agile и других методологий разработки.
- 7) Провести практические задания и проекты, позволяющие студентам применить полученные знания на практике и создать портфолио для демонстрации потенциальным работодателям.
- 8) Изучить принципы и методы оптимизации веб-сайтов для различных устройств и браузеров, обеспечивая лучшую производительность и удобство пользовательского интерфейса.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

3.1. Программа направлена на подготовку квалифицированных специалистов в области веб-программирования, способных разрабатывать, тестировать и оптимизировать современные веб-приложения, соответствующие актуальным требованиям и стандартам индустрии.

3.2. Программа направлена на получение и совершенствование слушателями универсальных, общепрофессиональных и профессиональных компетенций.

В результате освоения программы слушатели будут обладать *универсальными компетенциями (УК)*, включающими в себя способность:

- Организовывать собственную деятельность, определять методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
- Использовать информационно-коммуникационные технологии в профессиональной деятельности.
- Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
- Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
- Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

В результате освоения программы слушатели будут обладать *общепрофессиональными компетенциями (ОПК)*:

- Реализовывать методы и технологии защиты информации в базах данных.
- Осуществлять разработку тестовых наборов и тестовых сценариев.
- Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
- Выполнять тестирование программных модулей.
- Осуществлять оптимизацию программного кода модуля.
- Реализовывать базу данных в конкретной СУБД.
- Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.
- Осуществлять ревьюирование кода и технической документации.
- Выполнять работы по модификации отдельных компонент программного обеспечения.

3.3. Планируемым результатом обучения является освоение как теоретических знаний, так и практических умений, и навыков.

В результате освоения программы обучающиеся

будут знать:

- Особенности отображения ИР в размерах рабочего пространства устройств;
- Основы современных систем управления базами данных;
- Основы информационной безопасности web-ресурсов;
- Современные принципы построения интерфейсов пользователя;
- Современные методики тестирования эргономики пользовательских интерфейсов;
- Современный отечественный и зарубежный опыт в профессиональной деятельности;
- Программные средства и платформы для разработки web-ресурсов;
- Современные стандарты взаимодействия компонентов распределенных приложений;
- Основы современных систем управления базами данных;
- Методики описания и моделирования процессов, средства моделирования процессов;
- Основы теории системного анализа и построения диаграмм взаимодействия;
- Системы хранения и анализа баз данных;
- Технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии;
- Методы и приемы отладки программного кода;

будут уметь:

- Определять возможности отображения web-страниц в размерах рабочего пространства устройств для разных видов дизайн-макетов;
- Производить настройку параметров web-сервера;
- Идентифицировать права пользователей в зависимости от функционала ИР;
- Координировать решение запросов заказчиков со специалистами соответствующих подразделений;
- Осуществлять процесс проектирования интерфейса с учетом

- существующих правил для предметной области проекта;
- Применять инструменты для оценки эффективности и удобства созданного интерфейса, применять полученные данные для оптимизации интерфейса;
- Пользоваться нормативно-технической документацией в области программного обеспечения;
- Тестировать ИР с использованием тест-планов;
- Устанавливать СУБД;
- Производить настройку параметров web-сервера;
- Применять установленные правила делового общения при общении с заказчиком;
- Использовать программные продукты для графического отображения алгоритмов;
- Выявлять ошибки в программном коде;

4. УЧЕБНЫЙ ПЛАН

дополнительной профессиональной программы повышения квалификации «Веб программирование»

Цель обучения заключается в формировании у слушателей глубоких теоретических знаний и практических навыков в области фронтенд-разработки. Обучающиеся приобретут компетенции в создании современных, адаптивных и высокопроизводительных веб-интерфейсов, ориентированных на потребности конечного пользователя. Это позволит им стать высококвалифицированными специалистами, готовыми к решению сложных профессиональных задач и адаптации к быстро меняющимся технологическим трендам в сфере веб-разработки.

Категория слушателей – лица, имеющие среднее профессиональное и (или) высшее образование; лица, получающие среднее профессиональное и (или) высшее образование
Трудоемкость обучения: 240 академических часов.

Форма обучения: очная (сочетание теоретических и практических занятий)

№ п/ п	Наименование компонентов программы	Трудоемкость, ак. ч. ¹				Форма контро ля
		Вс его	Лекци и	Практич еские занятия / Самосто	Конт роль	

				атель- ная работа		
Модуль 1. Типы данных		32	16	6	10	
1.	Тема 1. Введение в типы данных	10	6	2	2	Контроль- ные вопросы
2	Тема 2. Прimitивные типы данных	10	6	2	2	Контроль- ные вопросы
3	Тема 3. Составные типы данных:	8	4	2	2	Контроль- ные вопросы
4	Промежуточная аттестация. Модуль 1	4	0	0	4	Контроль- ные вопросы
Модуль 2. Алгоритмы, методы Построения		20	8	4	8	
5	Тема 1. Введение в алгоритмы	8	4	2	2	Контроль- ные вопросы
6	Тема 2. Основные методы построения алгоритмов	8	4	2	2	Контроль- ные вопросы
7	Промежуточная аттестация. Модуль 2	4	0	0	4	Контроль- ные вопросы
Модуль 3. Основные этапы		28	12	6	10	

разработки веб программ						
8	Тема 1. Введение в веб-разработку	8	4	2	2	Контрольные вопросы
9	Тема 2. Технологии и инструменты веб-разработки	8	4	2	2	Контрольные вопросы
10	Тема 3. Проектирование и дизайн веб-приложений	8	4	2	2	Контрольные вопросы
11	Промежуточная аттестация. Модуль 3	4	0	0	4	Контрольные вопросы
Модуль 4. Базы данных и их роль в веб-приложении		26	12	6	8	
12	Тема 1. Основы модульного программирования	8	4	2	2	Контрольные вопросы
13	Тема 2. Создание и использование модулей	8	4	2	2	Контрольные вопросы
14	Тема 3. Интеграция модулей	8	4	2	2	Контрольные вопросы
15	Промежуточная аттестация. Модуль 4	2	0	0	2	Контрольные вопросы

Модуль 5. Базы данных, хранимые процедуры, триггеры		27	12	6	9	
16	Тема 1. Введение в базы данных	8	4	2	2	Контрольные вопросы
17	Тема 2. Хранимые процедуры	8	4	2	2	Контрольные вопросы
18	Тема 3. Триггеры в базах данных	8	4	2	2	Контрольные вопросы
19	Промежуточная аттестация. Модуль 5	3	0	0	3	Контрольные вопросы
Модуль 6. Разработка технической документации		27	12	6	9	
20	Тема 1. Значение технической документации	8	4	2	2	Контрольные вопросы
1	Тема 2. Структура и элементы технической документации	8	4	2	2	Контрольные вопросы
22	Тема 3. Создание и обновление документации	8	4	2	2	Контрольные вопросы
23	Промежуточная	3	0	0	3	Контрольные вопросы

	аттестация. Модуль 6					
Модуль 7. Практическая работа на примере создания веб приложения		72	36	18	26	
24	Тема 1. Планирование веб-приложения	8	4	2	2	Защита проекта
25	Тема 2. Разработка клиентской части	8	4	2	2	Защита проекта
26	Тема 3. Разработка серверной части	8	4	2	2	Защита проекта
27	Тема 4. Интеграция баз данных	8	4	2	2	Защита проекта
28	Тема 5. Безопасность веб-приложения	8	4	2	2	Защита проекта
29	Тема 6. Взаимодействие с внешними API	8	4	2	2	Защита проекта
30	Тема 7. Оптимизация производительности	8	4	2	2	Защита проекта
31	Тема 8. Мобильная адаптация	8	4	2	2	Защита проекта

32	Тема 9. Тестирование веб-приложения	8	4	2	2	Защита проекта
33	Промежуточная аттестация. Модуль 7	0	0	0	8	Защита проекта
34	Итоговая аттестация	8	0	0	8	Контрольные вопросы
35	Итого	240	108	52	88	

5. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Месяцы	Дисциплины	Всего академических часов	Продолжительность одного занятия
1 месяц	Модуль 1. Типы данных Тема 1. Введение в типы данных Тема 2. Примитивные типы данных Тема 3. Составные типы данных Модуль 2. Алгоритмы, методы Построения Тема 1. Введение в алгоритмы Тема 2. Основные методы построения алгоритмов Модуль 3. Основные этапы разработки веб программ Тема 1. Введение в веб-разработку Тема 2. Технологии и инструменты веб-разработки Тема 3. Проектирование и дизайн веб-приложений	80	4
2 месяц	Модуль 4. Модульная разработка Тема 1. Основы модульного программирования Тема 2. Создание и использование модулей Тема 3. Интеграция модулей Модуль 5. Базы данных, хранимые процедуры, триггеры Тема 1. Введение в базы данных Тема 2. Хранимые процедуры Тема 3. Триггеры в базах данных	80	4

	Модуль 6. Разработка технической документации Тема 1. Значение технической документации Тема 2. Структура и элементы технической документации Тема 3. Создание и обновление документации		
3 месяц	Модуль 7. Практическая работа - создание веб приложения Тема 1. Планирование веб-приложения Тема 2. Разработка клиентской части Тема 3. Разработка серверной части Тема 4. Базы данных и их роль в веб-приложении Тема 5. Безопасность веб-приложения Тема 6. Взаимодействие с внешними API Тема 7. Оптимизация производительности Тема 8. Мобильная адаптация Тема 9. Тестирование веб-приложения	80	4

6. РАБОЧАЯ ПРОГРАММА

Модуль 1. Типы данных:

Тема 1. Введение в типы данных:

1.1. Понятие типов данных и их роль в программировании.

1.2. Классификация типов данных.

1.3. Применение различных типов данных.

Тема 2. Примитивные типы данных:

2.1. Числовые, строковые, логические типы.

2.2. Специальные типы: null и undefined.

2.3. Операции с примитивными типами данных.

Тема 3. Составные типы данных:

3.1. Массивы и объекты.

3.2. Структуры и списки.

3.3. Преобразования и операции с составными типами данных.

Модуль 2. Алгоритмы, методы Построения:

Тема 1. Введение в алгоритмы:

1.1. Основные понятия и классификация алгоритмов.

1.2. Методы построения и анализа алгоритмов.

1.3. Реализация алгоритмов.

1.4. Тестирование алгоритмов

1.5. Способы отладки алгоритмов

Тема 2. Основные методы построения алгоритмов:

2.1. Рекурсивные алгоритмы.

2.2. Жадные алгоритмы и динамическое программирование.

2.3. Разделяй и властвуй.

2.4. Другие важные алгоритмы

Модуль 3. Основные этапы разработки веб программ:

Тема 1. Введение в веб-разработку:

1.1. Основы клиент-серверной архитектуры.

1.2. Фронтенд и бэкенд разработка.

1.3. Жизненный цикл веб-приложения.

Тема 2. Технологии и инструменты веб-разработки:

2.1. Основы HTML, CSS и JavaScript.

2.2. Фреймворки и библиотеки для фронтенда (React, Vue, Angular).

2.3. Технологии бэкенда (Node.js, Django, Flask).

Тема 3. Проектирование и дизайн веб-приложений:

3.1. Макетирование и прототипирование интерфейсов.

3.2. UX и UI дизайн.

3.3. Адаптивная и кроссбраузерная верстка.

Модуль 4. Модульная разработка:

Тема 1. Основы модульного программирования:

1.1. Понятие модуля и его роль.

1.2. Преимущества модульной разработки.

1.3. Инкапсуляция и повторное использование.

Тема 2. Создание и использование модулей:

2.1. Модульная структура и зависимости.

2.2. Экспорт и импорт модулей.

2.3. Управление модулями и библиотеками.

Тема 3. Интеграция модулей:

3.1. Модульные фреймворки и инструменты.

3.2. Работа с пакетными менеджерами.

3.3. Решение конфликтов и устранение ошибок.

Модуль 5. Базы данных, хранимые процедуры, триггеры:

Тема 1. Введение в базы данных:

- 1.1. Основные понятия баз данных.
- 1.2. Типы и структуры баз данных.
- 1.3. Язык SQL и его особенности.

Тема 2. Хранимые процедуры:

- 2.1. Основы создания и использования.
- 2.2. Параметризация и возвращаемые значения.
- 2.3. Оптимизация хранимых процедур.

Тема 3. Триггеры в базах данных:

- 3.1. Основы работы с триггерами.
- 3.2. Создание и удаление триггеров.
- 3.3. Применение триггеров для автоматизации действий.

Модуль 6. Разработка технической документации:

Тема 1. Значение технической документации:

- 1.1. Цели и задачи.
- 1.2. Аудитория и стейкхолдеры.
- 1.3. Форматы и стандарты документации.

Тема 2. Структура и элементы технической документации:

2.1. Технические требования и спецификации.

2.2. Диаграммы и схемы.

2.3. Инструкции и руководства.

Тема 3. Создание и обновление документации:

3.1. Инструменты для написания документации.

3.2. Версионирование и учет изменений.

3.3. Коллаборация и совместная работа.

Модуль 7. Практическая работа на примере создания веб приложения:

Тема 1. Планирование веб-приложения:

1.1. Определение требований и задач.

1.2. Прототипирование и дизайн.

1.3. Выбор инструментов и технологий.

Тема 2. Разработка клиентской части:

2.1. Создание интерфейса и взаимодействие с пользователем.

2.2. Интеграция с бэкендом и внешними сервисами.

2.3. Тестирование и оптимизация клиентской части.

Тема 3. Разработка серверной части:

3.1. Настройка сервера и базы данных.

3.2. Реализация бизнес-логики и API.

3.3. Безопасность и оптимизация серверной части.

Тема 4. Базы данных и их роль в веб-приложении:

4.1. Структурирование данных.

4.2. Запросы и манипуляции с данными.

4.3. Оптимизация и индексирование.

Тема 5. Безопасность веб-приложения:

5.1. Основы кибербезопасности.

5.2. Защита от основных видов атак.

5.3. Работа с сертификатами и шифрованием.

Тема 6. Взаимодействие с внешними API:

6.1. Подключение к сторонним сервисам.

6.2. Работа с ответами API.

6.3. Обработка ошибок и исключений.

Тема 7. Оптимизация производительности:

7.1. Анализ скорости загрузки страниц.

7.2. Оптимизация ресурсов и кэширование.

7.3. Улучшение пользовательского опыта.

Тема 8. Мобильная адаптация:

8.1. Особенности мобильных веб-приложений.

8.2. Адаптивный дизайн.

8.3. Работа с мобильными функциями.

Тема 9. Тестирование веб-приложения:

9.1. Виды тестирования.

9.2. Автоматизированные тесты.

9.3. Дебаггинг и поиск ошибок.

.
³ Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут

Материально-техническое обеспечение

- учебный стол - 5 шт
- стол преподавателя - 1 шт
- стулья для учеников - 5 шт
- стул для преподавателя - 1 шт
- ноутбуки - 6 шт, в таблице приводим серийный номер и модели

№	Модель техники	Серийный номер
1	MacBook Air (M1, 2020)	C02G42F3Q6LR
2	MacBook Air (M1, 2020)	C02GR0SMQ6M0
3	MacBook Air (13-inch, 2017)	FVFXP86WJ1WK
4	MacBook Air (M1, 2020)	C02GD1UJQ6LR
5	MacBook Pro(13-inch, 2017, Two Thunderbolt 3 ports)	FVFXD147HV29
6	MacBook Air (13-inch, 2017)	FVFD74ZHJ1WK

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

Вопросы для проведения контрольных работ:

Модуль 1. Типы данных:

Тема 1. Введение в типы данных:

1. Что такое тип данных, и какова его роль в программировании?
2. Какие основные категории типов данных существуют, и можно ли их классифицировать?
3. Почему важно правильно выбирать тип данных для переменных в программе?
4. Какие языки программирования поддерживают динамическую типизацию, и как это влияет на разработку?
5. Какие проблемы могут возникнуть из-за неправильного выбора типа данных?
6. Какие типы данных используются для чисел в разных системах счисления?
7. В чем разница между знаковыми и беззнаковыми целочисленными типами данных?
8. Какие типы данных используются для хранения логических значений?
9. Какие типы данных поддерживают работу с символами и строками?
10. Какие преимущества предоставляет статическая типизация в языках программирования?
11. Какие средства предоставляют языки программирования для преобразования типов данных?
12. Какие особенности имеют типы данных для работы с датами и временем?
13. Какие типы данных используются для хранения множественных значений, таких как перечисления (enums)?
14. Какие языки программирования поддерживают пользовательские типы данных, и как их создавать?
15. Какие методы и инструменты можно использовать для проверки и обработки типов данных в программе?

Тема 2. Примитивные типы данных:

1. Какие примитивные типы данных существуют в языках программирования?
2. Какие операции можно выполнять с целочисленными типами данных, и какие результаты они дают?
3. В чем разница между целочисленными и вещественными типами данных, и когда их следует использовать?
4. Какие операции можно выполнять с вещественными числами, и какие особенности точности и округления у них есть?
5. Какие типы данных используются для хранения символов и строк, и какие операции можно выполнять с ними?
6. Какие особенности есть у логических типов данных, и как они используются в условных операторах?
7. Какие типы данных используются для работы с байтами и битами?
8. Какие языки программирования поддерживают длинную арифметику, и в каких случаях она полезна?
9. Какие преимущества и ограничения есть у типов данных с плавающей запятой?
10. Какие типы данных используются для работы с комплексными числами?
11. Какие системы счисления могут использоваться в программах, и какие типы данных соответствуют каждой из них?
12. Какие особенности есть у типов данных для хранения символов и юникодных символов?
13. Какие операции и функции предоставляются для работы с текстовыми данными?
14. Какие языки программирования поддерживают пользовательские примитивные типы данных?
15. Какие рекомендации существуют по выбору примитивных типов данных для оптимизации работы программы?

Тема 3. Составные типы данных:

1. Что представляют собой составные типы данных и как они отличаются от примитивных?
2. Какие типы данных используются для хранения коллекций элементов, таких как списки и массивы?

3. Какие операции и функции предоставляются для работы с массивами и списками?
4. Какие разновидности массивов существуют в разных языках программирования?
5. Какие типы данных подходят для представления таблиц и матриц?
6. Как работают структуры и классы, и как они могут использоваться для создания пользовательских типов данных?
7. Какие особенности у структур и классов в объектно-ориентированном программировании?
8. Какие операции и методы могут быть определены для пользовательских типов данных?
9. Какие типы данных используются для хранения ассоциативных массивов или словарей?
10. Какие языки программирования поддерживают кортежи, и как они используются?
11. Какие структуры данных предназначены для хранения деревьев и графов?
12. Какие типы данных подходят для работы с файлами и потоками данных?
13. Какие библиотеки и фреймворки предоставляют расширенные составные типы данных?
14. Какие архитектурные шаблоны могут быть использованы для создания сложных составных типов данных?
15. Как обеспечить безопасность и целостность данных при работе с составными типами данных в программе?

Модуль 2. Алгоритмы, методы Построения:

Тема 1. Введение в алгоритмы и их структуры:

1. Что такое алгоритм?
2. Какие основные характеристики у алгоритмов?
3. Что такое псевдокод?
4. Какие основные структуры управления в алгоритмах вы знаете?
5. Что такое сложность алгоритма?
6. Чем отличается линейный алгоритм от квадратичного?
7. Какие алгоритмы относятся к "жадным"?
8. Что такое рекурсия?

9. Какие проблемы могут возникнуть при использовании рекурсии?
10. Что такое базовый и рекурсивный случаи в рекурсии?
11. Что такое итерация?
12. Чем отличается итерация от рекурсии?
13. Как оптимизировать алгоритм?
14. Что такое стек вызовов?
15. Каким образом можно представить алгоритм на бумаге?

Тема 2. Основные методы построения алгоритмов:

1. Что такое метод "Разделяй и властвуй"?
2. Какие задачи решаются методом динамического программирования?
3. Что такое "жадный" метод?
4. Как работает метод бэктрекинга?
5. В чем разница между методом мемоизации и табуляцией?
6. Что такое "градиентный спуск"?
7. Какие алгоритмы относятся к методу ветвей и границ?
8. Что такое "локальный" и "глобальный" минимумы в алгоритмах оптимизации?
9. Что такое оптимизация алгоритма?
10. Как определить, что один алгоритм лучше другого?
11. Что такое NP-полные задачи?
12. В чем заключается принцип наименьшего действия в контексте алгоритмов?
13. Что такое "симуляция отжига" в оптимизации алгоритмов?
14. В чем преимущества и недостатки использования генетических алгоритмов?
15. Что такое "метод Монте-Карло"?

Модуль 3. Основные этапы разработки веб программ:

Тема 1. Введение в веб-разработку:

1. Что такое веб-разработка, и какова ее роль в современном мире?
2. Какие основные компоненты составляют веб-сайт или веб-приложение?
3. Какие языки программирования и технологии используются в веб-разработке?

4. Какие этапы включает в себя процесс разработки веб-проекта?
5. Что такое клиентская и серверная сторона веб-разработки, и в чем их различия?
6. Каким образом работает протокол HTTP, и почему он важен для веб-разработчика?
7. Какие браузеры существуют, и какие особенности нужно учитывать при разработке под них?
8. Какие основные концепции веб-доступности следует учитывать при разработке?
9. Какие тенденции и новации в веб-разработке в настоящее время наиболее актуальны?
10. Какова роль мобильной адаптивности и отзывчивого дизайна в веб-разработке?
11. Какие методы тестирования и отладки используются в веб-разработке?
12. Какие вопросы безопасности нужно учитывать при создании веб-приложений?
13. Каким образом можно оптимизировать производительность веб-сайта или веб-приложения?
14. Какие инструменты и ресурсы доступны для обучения веб-разработке?
15. Какие ключевые навыки и компетенции важны для успешного веб-разработчика?

Тема 2. Технологии и инструменты веб-разработки:

1. Какие серверные языки программирования используются в веб-разработке, и для чего они предназначены?
2. Какие клиентские языки программирования применяются в веб-разработке, и какие задачи они решают?
3. Что такое HTML и какие основные элементы и атрибуты он содержит?
4. Как работает CSS, и какие методы оформления веб-страниц он предоставляет?
5. Каким образом JavaScript используется для создания интерактивных элементов на веб-сайте?
6. Что такое AJAX, и как он позволяет обеспечить динамичность веб-приложений?

7. Какие фреймворки и библиотеки наиболее популярны в веб-разработке?
8. Какие инструменты и среды разработки используются для написания, отладки и тестирования кода?
9. Что такое системы управления версиями, и почему они важны для коллаборативной разработки?
10. Какие базы данных применяются в веб-разработке, и какие задачи они решают?
11. Какие веб-серверы используются для развертывания веб-приложений, и как их настраивать?
12. Какие средства анализа производительности и отладки HTTP-запросов доступны для веб-разработчиков?
13. Какие технологии веб-сервисов и API интеграции важны в современной веб-разработке?
14. Какие инструменты и методы тестирования безопасности используются в веб-разработке?
15. Какие требования к хостингу и развертыванию веб-приложений стоит учитывать?

Тема 3. Проектирование и дизайн веб-приложений:

1. Что такое пользовательский интерфейс (UI) и опыт пользователя (UX) в веб-разработке?
2. Какие этапы включает в себя процесс проектирования веб-приложений?
3. Какие методы сбора и анализа требований клиента используются в дизайне веб-приложений?
4. Какие инструменты и методы прототипирования помогают создавать макеты веб-сайтов и приложений?
5. Каким образом выбираются цветовые схемы, шрифты и графические элементы для веб-приложения?
6. Как работают и какие возможности предоставляют графические редакторы в дизайне веб-сайтов?
7. Какие принципы и техники улучшения доступности и удобства использования следует применять?
8. Какие методы и инструменты адаптивного и отзывчивого дизайна применяются в веб-разработке?
9. Каким образом проводится тестирование дизайна и

- пользовательского опыта?
10. Какие требования к мобильным устройствам и разрешениям экранов нужно учитывать?
 11. Какие методы и инструменты анимации и интерактивности используются в веб-дизайне?

Модуль 4. Модульная разработка:

Тема 1. Основы модульного программирования:

1. Что такое модульное программирование, и какие основные принципы оно включает?
2. Какие преимущества предоставляет модульная разработка по сравнению с монолитным подходом?
3. Какая роль у модулей в структуре программного кода, и как они организуются?
4. Как модульность способствует повторному использованию кода?
5. Какие понятия, такие как инкапсуляция и абстракция, связаны с модульным программированием?
6. Каким образом модульная разработка способствует упрощению сопровождения и тестирования программ?
7. Какие языки программирования поддерживают модульное программирование, и как это реализуется в них?
8. Каким образом модули взаимодействуют между собой в модульной системе?
9. Какие методы и инструменты доступны для документирования модульного кода?
10. Какие рекомендации существуют для разработки хорошо структурированных модулей?
11. Каким образом модульная архитектура может улучшить распараллеливание и масштабируемость системы?
12. Какие типы ошибок и конфликтов могут возникнуть в модульной системе, и как их предотвращать?
13. Какие практические примеры существуют для модульного программирования в разных областях?
14. Какие подходы существуют к версионированию модульного кода и управлению зависимостями?
15. Как обеспечить безопасность и защиту данных в модульных

системах?

Тема 2. Создание и использование модулей:

1. Какие основные элементы модуля в языке программирования?
2. Как создать модуль в своем программном проекте, и какие требования к структуре кода существуют?
3. Каким образом происходит экспорт и импорт функций и переменных между модулями?
4. Как можно обеспечить сокрытие данных и функций внутри модуля, используя инкапсуляцию?
5. Какие принципы и лучшие практики следует соблюдать при именовании модулей и их компонентов?
6. Как организовать иерархию модулей в больших проектах для удобства управления?
7. Какие возможности предоставляют модульные системы для тестирования модулей в изоляции?
8. Каким образом можно оценить качество и производительность модульного кода?
9. Какие инструменты и среды разработки поддерживают создание и использование модулей?
10. Какие способы документирования и комментирования кода помогают другим разработчикам понимать модульный код?
11. Какие языки и технологии используются для создания модульных интерфейсов пользователя?
12. Как обеспечить совместимость и поддержку сторонних модулей в вашем программном проекте?
13. Как управлять версиями модулей и учитывать изменения в интерфейсах?
14. Какие примеры существуют для создания и использования модулей в различных программных областях?
15. Какие инструменты и техники могут быть использованы для оптимизации модульного кода?

Тема 3. Интеграция модулей:

1. Что представляют собой модульные фреймворки, и как они упрощают разработку приложений?
2. Какие инструменты и среды поддерживают управление модулями

- и библиотеками в больших проектах?
3. Какие методы и практики могут помочь в решении конфликтов между модулями?
 4. Как обнаружить и устранить ошибки, связанные с интеграцией модулей в программе?
 5. Каким образом разрабатываются и поддерживаются плагины и расширения для модульных систем?
 6. Какие принципы и соглашения следует придерживаться при интеграции сторонних модулей?
 7. Какие задачи решает система управления зависимостями, и какие примеры таких систем существуют?
 8. Каким образом можно обеспечить согласованность интерфейсов и версий модулей в распределенных системах?
 9. Как происходит обновление и миграция модульных систем без нарушения работы приложений?
 10. Какие методы аутентификации и авторизации используются при интеграции модулей с внешними системами?

Модуль 5. Базы данных, хранимые процедуры, триггеры:

Тема 1. Введение в базы данных:

1. Что такое база данных, и какова ее роль в современных информационных системах?
2. Какие основные типы баз данных существуют, и в чем их отличия?
3. Что такое SQL, и как он используется для работы с базами данных?
4. Какие компоненты составляют базу данных, и как они взаимодействуют между собой?
5. Какие преимущества и недостатки имеют реляционные и нереляционные базы данных?
6. Каким образом происходит проектирование схемы базы данных, и какие этапы этому предшествуют?
7. Какие операции выполняются над данными в базе данных, и какие языки запросов существуют?
8. Какие требования к безопасности и уровню доступа существуют в базах данных?

9. Как обеспечивается целостность данных и их сохранность в базах данных?
10. Какие методы резервного копирования и восстановления данных используются?
11. Какие роли выполняют администраторы баз данных и разработчики при работе с БД?
12. Каким образом можно оптимизировать производительность запросов к базе данных?
13. Какие инструменты и системы управления базами данных (СУБД) существуют?
14. Какие тенденции и новации в области баз данных в настоящее время актуальны?
15. Какие рекомендации существуют для обеспечения согласованности и надежности баз данных?

Тема 2. Хранимые процедуры:

1. Что представляют собой хранимые процедуры, и какие задачи они решают в базах данных?
2. Как создать и вызвать хранимую процедуру в различных СУБД?
3. Какие преимущества предоставляют хранимые процедуры по сравнению с обычными SQL-запросами?
4. Каким образом можно передавать параметры в хранимую процедуру, и какие типы параметров существуют?
5. Какие операции и логика могут быть реализованы внутри хранимых процедур?
6. Как обеспечить безопасность и авторизацию при вызове хранимых процедур?
7. Какие инструменты и языки программирования поддерживают работу с хранимыми процедурами?
8. Каким образом хранимые процедуры способствуют снижению нагрузки на сеть и базу данных?
9. Какие механизмы обработки ошибок и исключений существуют в хранимых процедурах?
10. Каким образом можно версионировать и управлять хранимыми процедурами в больших проектах?
11. Какие лучшие практики существуют для проектирования и оптимизации хранимых процедур?

12. Какие примеры существуют для использования хранимых процедур в реальных проектах?
13. Каким образом хранимые процедуры могут быть интегрированы с другими частями приложения?
14. Какие технологии масштабирования и репликации поддерживают хранимые процедуры?
15. Какие ограничения и осложнения могут возникнуть при использовании хранимых процедур?

Тема 3. Триггеры в базах данных:

1. Что такое триггеры, и какие задачи они решают в контексте баз данных?
2. Как создать и активировать триггеры в различных СУБД?
3. Какие типы триггеров существуют, и в чем их отличия?
4. Каков порядок выполнения триггеров при различных операциях с данными?
5. Какие события и операции могут быть связаны с триггерами в базе данных?
6. Как передавать данные и параметры внутрь триггеров и из них?
7. Каким образом триггеры могут воздействовать на данные и структуру базы данных?
8. Как обеспечить безопасность и контроль доступа при работе с триггерами?
9. Какие ограничения и потенциальные проблемы следует учитывать при использовании триггеров?
10. Какие лучшие практики существуют для проектирования и оптимизации триггеров?
11. Каким образом триггеры могут быть использованы для аудита и журналирования действий?
12. Как можно версионировать и управлять триггерами в больших проектах?
13. Какие примеры существуют для использования триггеров в реальных приложениях?

Модуль 6. Разработка технической документации:

Тема 1. Значение технической документации:

1. Какое значение имеет техническая документация в разработке продуктов и проектов?
2. Какую роль играет техническая документация в обеспечении эффективной коммуникации внутри команды разработчиков?
3. Какая связь существует между технической документацией и качеством разрабатываемого продукта?
4. Как техническая документация способствует обеспечению безопасности и соблюдению стандартов?
5. Какие требования и стандарты могут быть применены к разработке технической документации?
6. Какова роль технической документации в поддержке и обслуживании продукта после его выпуска?
7. Какие типы аудиторий и пользователей могут иметь доступ к технической документации, и как это влияет на ее содержание?
8. Какие основные виды технической документации используются в различных областях разработки?
9. Какие инструменты и технологии могут быть применены для создания и управления технической документацией?
10. Как можно оценить эффективность технической документации в конкретном проекте?
11. Каким образом техническая документация помогает управлять изменениями в проекте?
12. Какие практические рекомендации существуют для создания высококачественной технической документации?
13. Каким образом техническая документация может помогать в процессе обучения и передаче знаний?
14. Какие вызовы и трудности могут возникнуть при разработке и обновлении технической документации?
15. Какие будущие тенденции и инновации можно ожидать в области технической документации?

Тема 2. Структура и элементы технической документации:

1. Какова общая структура технической документации, и какие разделы она обычно включает?
2. Какие элементы заголовка и оглавления используются для организации документации?
3. Как описываются основные характеристики и спецификации

продукта в документации?

4. Какие разделы и главы могут включать в себя инструкции по установке и настройке продукта?
5. Каким образом описываются функции и возможности продукта, а также способы их использования?
6. Какие элементы документации могут содержать информацию о безопасности и соблюдении нормативов?
7. Как оформляются иллюстрации, схемы, графики и другие визуальные элементы в технической документации?
8. Как можно предоставить информацию о процессе обслуживания, технической поддержке и устранении неполадок?
9. Каким образом создаются и обновляются таблицы, списки и перечни в документации?
10. Какие виды ссылок и перекрестных ссылок используются для навигации в документации?
11. Какие термины, сокращения и определения включаются в словарь или глоссарий технической документации?
12. Как можно предоставить информацию о лицензировании, авторских правах и использовании материалов?
13. Каким образом организуется и поддерживается индекс и поиск в технической документации?
14. Какие шаблоны и стандарты форматирования могут быть применены к созданию документации?
15. Какие методы и инструменты поддерживают автоматизацию создания и публикации технической документации?

Тема 3. Создание и обновление документации:

1. Какие методы сбора и анализа информации используются при создании документации?
2. Как проводится верификация и проверка достоверности информации в документации?
3. Каким образом управляется процесс редактирования и коррекции ошибок в документации?
4. Как создать консистентность и стабильность в обновлении документации при изменениях в продукте?
5. Какие практики документирования кода и комментирования исходного кода существуют?

6. Каким образом можно учесть различные версии продукта и обеспечить актуальность документации?
7. Как осуществляется сотрудничество и коммуникация между авторами и редакторами документации?
8. Как обеспечивается качество и стандартизация структуры и стиля документации?
9. Каким образом поддерживается и управляется архивация и история изменений в документации?
10. Как можно улучшить доступность и удобство использования технической документации для пользователей?
11. Какие инструменты и системы управления версиями подходят для управления процессом документирования?
12. Каким образом можно интегрировать создание документации в процесс разработки продукта?
13. Какие практики существуют для обучения и развития навыков создания технической документации?

ИТОГОВАЯ АТТЕСТАЦИЯ

Формы проведения итоговой аттестации

Итоговая аттестация проводится в форме письменных ответов на контрольные вопросы и устной защиты практической работы

Критерии оценивания итоговой аттестации

Оценочное средство сформированности компетенций	Компетенция не сформирована, соответствует академической оценке «неудовлетворительно»	Уровень 1 (пороговый), соответствует академической оценке «удовлетворительно»	Уровень 2 (средний), соответствует академической оценке «хорошо»	Уровень 3 (высокий), соответствует академической оценке «отлично»
Требования к экзамену	Студент не демонстрирует базовых знаний и практических навыков по курсу. Неверное применение методик и инструментов в разработке. Неспособность анализировать и решать типовые задачи.	Студент демонстрирует базовые знания и навыки. Способность применять основные методики и инструменты. Отсутствие глубокого понимания некоторых ключевых аспектов.	Уверенное владение большинством тем курса. Продуктивное применение методик и инструментов разработки. Способность к анализу и решению более сложных задач.	Глубокие знания и навыки по всем темам курса. Творческий подход к решению сложных задач. Умение оптимизировать код и критически мыслить

Контрольные вопросы для проведения итоговой аттестации

1. Что такое типы данных и какова их роль в программировании?
2. Какие основные классы типов данных вы знаете?
3. В каких ситуациях вы бы использовали строковые типы данных?
4. Что отличает числовой тип данных от строкового?
5. Что такое логические типы данных и где они обычно применяются?
6. Как вы бы определили `null` и `undefined`?
7. Какие операции можно выполнять с примитивными типами данных?
8. Что такое массивы и как они отличаются от объектов?
9. Какова структура данных "список" и в чем ее особенности?
10. В чем разница между структурой данных и типом данных?
11. Какие операции можно выполнять с составными типами данных?
12. Что такое алгоритм?
13. Как вы бы классифицировали различные типы алгоритмов?
14. В чем заключается метод "разделяй и властвуй"?
15. Какие основные этапы входят в реализацию алгоритма?
16. Почему тестирование алгоритмов важно?
17. Какие основные методы отладки алгоритмов вы знаете?
18. В чем преимущество рекурсивных алгоритмов перед итеративными?
19. Как динамическое программирование помогает оптимизировать решения?
20. Какова основная идея жадных алгоритмов?
21. Что такое клиент-серверная архитектура?
22. В чем основное различие между фронтенд и бэкенд разработкой?
23. Какие этапы входят в жизненный цикл веб-приложения?
24. Каковы основные особенности языков HTML, CSS и JavaScript?
25. Какой фреймворк лучше использовать: React, Vue или Angular?
26. В чем отличие Node.js от Django и Flask?
27. Что такое прототипирование интерфейсов?
28. В чем заключается разница между UX и UI дизайном?
29. Что такое адаптивная верстка?
30. Что такое модуль в программировании и какова его роль?

31. Какие преимущества дает модульная разработка?
32. Что такое инкапсуляция в контексте модульного программирования?
33. Как экспортировать и импортировать модули?
34. Что такое пакетные менеджеры и как они используются в модульной разработке?
35. В чем основная цель базы данных?
36. Какие типы баз данных вы знаете?
37. Для чего используется язык SQL?
38. Что такое хранимая процедура и как ее создать?
39. В чем преимущества использования триггеров в базе данных?
40. Как создать и удалить триггер в базе данных?
41. Почему техническая документация важна для разработчиков?
42. Каковы основные элементы технической документации?
43. Для какой аудитории предназначена техническая документация?
44. Что такое технические требования и как их составлять?
45. Какие инструменты вы бы рекомендовали для написания документации?
46. Что такое версионирование документации и зачем оно нужно?
47. Как обеспечить коллаборацию при создании технической документации?
48. В чем заключается принцип повторного использования в модульной разработке?
49. Как оптимизировать хранимые процедуры в базе данных?
50. Что такое кроссбраузерная верстка и почему она важна?

Практические задания

Верстка статичной страницы:

- Задание: Создайте статичную веб-страницу на основе предоставленного макета.
- Требования: Использовать HTML5 и CSS3. Страница должна быть адаптивной и корректно отображаться на различных устройствах.

Интерактивное меню:

- Задание: Добавьте выпадающее меню на уже созданную веб-страницу.

- Требования: Использовать JavaScript или jQuery. Меню должно быть анимированным и интуитивно понятным.

Создание формы обратной связи:

- Задание: Разработайте форму для отправки сообщений с валидацией полей.
- Требования: Проверка на обязательные поля, корректность email и ограничение на длину сообщения.

Анимация элементов страницы:

- Задание: Добавьте анимации для выбранных элементов страницы (например, кнопок или изображений) при наведении курсора.
- Требования: Использовать CSS3 transitions и animations.

Интеграция с API:

- Задание: Интегрируйте ваш сайт с погодным API, чтобы показывать текущую погоду в заданном городе.
- Требования: Отправка запроса, обработка и отображение полученных данных на вашем сайте.

Создание карусели изображений:

- Задание: Реализуйте карусель изображений на вашем сайте.
- Требования: Возможность перелистывания, автоматическое проигрывание и пауза при наведении.

Реализация модального окна:

- Задание: Создайте модальное окно, которое отображается при клике на кнопку.
- Требования: Анимация появления и исчезновения, возможность закрыть окно по клику вне его или на специальной кнопке закрытия.

Работа с локальным хранилищем:

- Задание: Реализуйте функционал добавления товаров в корзину с сохранением выбранных товаров в localStorage.
- Требования: При обновлении страницы выбранные товары должны оставаться в корзине.

7. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

Реализация программы проходит в полном соответствии с требованиями законодательства Российской Федерации в области образования, нормативными правовыми актами, регламентирующими данные направления деятельности.

При обучении применяются различные виды занятий – лекции, практические занятия. Для закрепления изучаемого материала проводятся практические занятия.

Педагогические условия

Теоретические и практические занятия проводят преподаватели с высшим профильным образованием, имеющие достаточный опыт работы в области профессиональной деятельности, соответствующей направленности образовательной программы, состоящими в штате ООО «ИТ-Консалт» и привлекаемыми к педагогической деятельности на других законных основаниях.

СПИСОК ИСПОЛЬЗОВАННЫХ РЕСУРСОВ

<https://reactpatterns.com>
<https://dev.to/johongirr/33-react-libraries-every-react-developer-should-have-in-their-arsenal-29dd> <https://legacy.reactjs.org/docs/getting-started.html>
<https://stackoverflow.com/questions/tagged/javascript>
<https://developer.mozilla.org/en-US/docs/Web/HTML>
<https://sass-lang.com/documentation/>
<https://stackoverflow.com/questions/22876978/loop-inside-react-jsx>
<https://overreacted.io/before-you-memo/>
<https://habr.com/ru/articles/646887/>
https://dev.to/said_mounaim/awesome-things-related-to-react-hooks-30c4
<https://habr.com/ru/companies/southbridge/news/752952/>
<https://stackoverflow.com/questions/31048953/what-are-these-three-dots-in-react-doing>
<https://habr.com/ru/companies/otus/articles/759222/>
<https://habr.com/ru/articles/653431/>
<https://base.garant.ru/71600548/>
<https://dev.to/sakhnyuk/300-react-interview-questions-2ko4>
<https://stackoverflow.com/questions/34641582/what-is-the-difference-between-react-native-and-react>
<https://dev.to/bricourse/the-most-powerful-react-js-cheat-sheet-4ipd>
<https://dev.to/benghamine/full-list-of-react-conferences-in-2020-1a3o>
<https://stackoverflow.com/questions/31079081/programmatically-navigate-using-react-router>
<https://stackoverflow.com/questions/31079081/programmatically-navigate-using-react-router/42121109#42121109>