

РАБОЧАЯ ПРОГРАММА
(курса)
Программирование на JavaScript (полный курс)

1. АННОТАЦИЯ

Цель. Курс представляет собой введение в программирование на языке JavaScript. Включает в себя основы синтаксиса, работу с операторами, массивами, функциями и т.д. Курс предполагает изучение языка JavaScript с нуля.

JavaScript — идеальный язык для начинающих разработчиков. Самым главным преимуществом языка является его распространенность. JavaScript поддерживается на всех операционных системах, во всех видах браузеров, и на компьютерах, и на мобильных устройствах. JavaScript-приложения работают без установки на компьютер. JavaScript используется при разработке фронтальных и мобильных приложений, бэкенде. Пользователь может выбрать разные направления разработки, изучив этот язык.

2. СОДЕРЖАНИЕ

Учебная программа по модулю

№ п/п	Наименование модуля, разделов и тем	Содержание обучения (по темам в дидактических единицах), наименование и тематика лабораторных работ, практических занятий (семинаров), самостоятельной работы, используемых образовательных технологий и рекомендуемой литературы
<i>1.</i>	<i>2.</i>	<i>3.</i>
1.	Тема 1. Введение в JavaScript. Переменные и типы данных	4 часа 1. Типы данных
	Практическое занятие 1	1. Виды переменных 2. Типы данных
2.	Тема 2. GitHub. Типы данных. Преобразование типов. Операторы сравнения	4 часа 1. GitHub 2. Преобразование типов 3. Операторы сравнения
	Практическое занятие	1. Понятие преобразования типов 2. Виды операторов для сравнения
3.	Тема 3. Преобразование типов. Операторы сравнения. Массивы	4 часа 1 Преобразование типов 2. Операторы сравнения 3. Массивы
	Практическое занятие	1. Массивы и их виды 2. Способы управления массивами данных
4.	Тема 4. Условные и логические операторы, циклы	4 часа 1. Условные операторы (if, ?). 2. Конструкция switch. 3. Логические операторы (&&, , !). 4. Циклы while и for.
	Практическое занятие	1. Сущность понятия условные операторы 2. Сущность понятия логические операторы 3. Циклы и их виды
5.	Тема 5. Введение в функции	6 часов 1. Основы функций. 2. Синтаксис функций.

		3. Функциональные выражения. 4. Стрелочные функции.
	Практическое занятие	Функции и их виды
6.	Тема 6. Примитивы. Объекты	6 часов 1. Примитивы 2. Объекты
	Практическое занятие	1. Понятие примитивов 2. Объекты для использования примитивов
7.	Тема 7. Методы массивов и объектов	6 часов 1. Методы массивов. 2. Методы объектов.
	Практическое занятие	1. Методы массивов 2. Методы объектов
8.	Тема 8. Продвинутая работа с функциями	6 часов 1. Ключевое слово this 2. Работа с контекстом 3. Замыкания 4. Каррирование
	Практическое занятие	1. Работа с функциями 2. Продвинутый уровень работы с функциями
9.	Тема 9. Прототипы и классы	4 часа
	Практическое занятие	1. Понятие прототипов 2. Понятие классов
10.	Тема 10. DOM	4 часа 1. DOM 2. События
	Практическое занятие	1. Понятие DOM 2. Методы работы с DOM
11.	Тема 11. Асинхронность в JavaScript. EventLoop	4 часа 1. event loop 2. axios 3. async-thunk
	Практическое занятие	1. Понятие асинхронности и ее видов 2. EventLoop
12.	Тема 12. Обработка ошибок. Fetch	4 часа 1. Принципы обработки ошибок
	Практическое занятие	Реализация обработки ошибок
13.	Тема 1. Введение в продвинутый курс	4 часа 1. npm. Модули. TypeScript
	Практическое занятие	1. Изучить дополнительный материал
14.	Тема 2. Знакомство с React и TypeScript. Инициализация приложения	4 часа 1. Настройка рабочей среды
	Практическое занятие	1. Выполнить работу с модулями (на примере to do list)
15.	Тема 3. Знакомство с webpack	4 часа 1. webpack
	Практическое занятие	1. Установить и настроить базовую конфигурацию webpack 2. Настроить режим prod и dev для сборки и разработки с помощью webpack
16.	Тема 4. Компоненты, props, state	4 часа 1. Props, React Hooks, useState
	Практическое занятие	1. Необходимо реализовать универсальный

		<p>компонент карточки продукта. Этот компонент разбить на логические составляющие (<code><header><content><button></code> и тд). Компонент должен принимать различные пропсы (название, цена, картинка товара и тд).</p> <p>2. Для работы со state, добавить в карточку состояние отвечающее, за то что данный товар добавлен в корзину по клику на кнопку "Добавить", и добавить возможность выбирать количество товаров добавляемых в корзину.</p>
17.	Тема 5. Хуки и обработка событий	<p>4 часа</p> <ol style="list-style-type: none"> 1. Обработка событий 2. React Hooks
	Практическое занятие	<ol style="list-style-type: none"> 1. Необходимо реализовать задачу с использованием одного и того же хука несколько раз в одном компоненте
18.	Тема 6. Архитектура приложения, key - prop, стилизация компонентов	<p>4 часа</p> <ol style="list-style-type: none"> 1. Архитектура приложения 2. Стилизация компонентов 3. children 4. key-prop
	Практическое занятие	<ol style="list-style-type: none"> 1. Реализовать использование атрибута <code>style = {styleObject}</code> внутри <code>jsx</code>. 2. Составить задачу с использованием специального <code>props children</code>
19.	Тема 7. Жизненный цикл компонентов	<p>4 часа</p> <ol style="list-style-type: none"> 1. Жизненный цикл компонентов 2. <code>useEffect</code> 3. <code>React.memo</code>
	Практическое занятие	<p>4 часа</p> <ol style="list-style-type: none"> 1. Реализовать функционал удаления таймеров, удаления слушателей созданных во время жизни компонента и отмены запросов к серверу. 2. Реализовать поверхностное сравнение вложенных объектов в объекте <code>props</code>. 3. Реализовать валидацию
20.	Тема 8. Роутинг в реакт приложении	<p>4 часа</p> <ol style="list-style-type: none"> 1. Роутинг 2. <code>React-router-dom</code> 3. <code>BrowserRouter</code> 4. <code>Routes</code> 5. <code>Route</code> 6. <code>Link</code>
	Практическое занятие	<ol style="list-style-type: none"> 1. Установить библиотеку <code>React-router-dom</code> 2. Реализовать навигацию по клику, перенаправление, маршрутизацию, работу с историей
21.	Тема 9. Redux toolkit и работа с API	<p>4 часа</p> <ol style="list-style-type: none"> 1. <code>redux</code> — теория 2. <code>redux-toolkit</code>
	Практическое занятие	<ol style="list-style-type: none"> 1. Получение доступа с использованием хука <code>useDispatch()</code> из библиотеки <code>react-redux</code>. 2. Получить доступ к возвращаемому значению с помощью хука <code>useSelector()</code> из библиотеки <code>react-redux</code>
22.	Тема 10. Продвинутая работа с хуками	<p>4 часа</p> <ol style="list-style-type: none"> 1. <code>useRef</code>

		2. useCallback 3. useMemo
	Практическое занятие	1. Реализовать управление фокусом, воспроизведение медиа, выделение текста, императивный вызов анимаций
23.	Тема 11. Основы Node.js	4 часа 1. Основы Node.js
	Практическое занятие	1. С использованием Node.js реализовать сбор и обработку данных

3. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ МОДУЛЯ

Итоговая аттестация проходит в форме зачета – выполнения итогового задания на компьютере.

Таблица 2

Формы и методы контроля и оценки результатов освоения модулей

№ п/п	Наименование процедуры	Основные показатели оценки	Формы и методы контроля и оценки
1.	Текущий контроль. Тема 1. Введение в JavaScript. Переменные и типы данных	Владеет основными понятиями программирования	Создать npm проект, установить и обновить необходимые пакеты. Разбить js-код на модули
2.	Текущий контроль. GitHub. Тема 2. Типы данных. Преобразование типов. Операторы сравнения	Владеет характеристиками React, TypeScript, инициализация приложения	Создать react приложение. Провести базовую настройку инструментов разработчика.
3.	Текущий контроль. Тема 3. Преобразование типов. Операторы сравнения. Массивы	Владеет основными понятиями и характеристиками webpack	Провести настройку webpack сборки в prod и dev режимы. Конфигурация babel
4.	Текущий контроль. Тема 4. Условные и логические операторы, циклы	Знает характеристики props, state	Создать react-компоненты. Передать необходимые props, добавить типизацию props
5.	Текущий контроль. Тема 5. Введение в функции	Владеет функцией Хуки и обработки событий	Добавить состояние для компонента с помощью хука useState. Рассказать про условия применения хуков. Добавить базовые обработчики события (onClick, onChange)
6.	Текущий контроль. Тема 6. Примитивы. Объекты	Владеет параметрами архитектуры приложения, key - prop, стилизации компонентов	Рассказать про container-component подход. Стилизовать компонент с помощью scss модулей и styled-components. На основе статического списка объяснить принцип работы key-props
7.	Текущий контроль. Тема 7. Методы массивов и объектов	Понимает особенности жизненного цикла компонентов	С помощью хука useEffect наделить компонент логикой работы с методами жизненного цикла. Затронуть этапы

			монтажирования, обновления и размонтирования компонента.
8.	Текущий контроль. Тема 8. Продвинутая работа с функциями	Знает функции и возможности Роутинг в реакт приложении	Настроить роутинг в приложении. Рассказать базовые принципы навигации в SPA
9.	Текущий контроль. Тема 9. Прототипы и классы	Знает функции и возможности Redux toolkit, документацию API	Подключить и настроить redux-toolkit. Рассказать про основные составляющие redux. С помощью API положить данные в redux store и отрендерить их на странице
10.	Текущий контроль. Тема 10. DOM	Применяет методы продвинутой работы с хуками	С помощью хука useRef получить доступ до DOM и провести различные манипуляции с элементом (добавление/удаление атрибутов). Рассказать про мемоизацию компонентом. Применить хуки useMemo и useCallback для оптимизации рендера компонента
11.	Текущий контроль. Тема 11. Асинхронность в JavaScript. EventLoop	Владеет основами Node.js	Добавить API к текущему проекту
12.	Текущий контроль . Тема 12. Обработка ошибок. Fetch	Владеет основными принципами обработки ошибок	Реализовать перехват и обработку ошибки при передаче неправильной ссылки
13.	Текущий контроль Тема 13. Введение в продвинутый курс	Владеет основными понятиями программирования	Проверка усвоения материала
14.	Текущий контроль Тема 14. Знакомство с React и TypeScript. Инициализация приложения	Владеет характеристиками React, TypeScript, и инициализация приложения	Проверка настройки рабочей среды
15.	Текущий контроль Тема 15. Знакомство с webpack	Владеет основными понятиями и характеристиками webpack	Провести базовую настройку babel
16.	Текущий контроль Тема 16. Компоненты, props, state	Знает характеристики props, state	Выбрать макет с любого ресурса. Создать страницу, с разбиением на логические компоненты (header, content (mainInfo + description + slider + gallery), footer).
17.	Текущий контроль Тема 17. Хуки и обработка событий	Владеет функцией Хуки и обработки событий	Проверка реализации задачи «Карусель»
18.	Текущий контроль Тема 18. Архитектура приложения, key - prop, стилизация компонентов	Владеет параметрами архитектуры приложения, key - prop, стилизации компонентов	Добавить в приложение несколько страниц, следуя заложенному подходу (контейнер-компоненты).
19.	Текущий контроль Тема 19. Жизненный цикл компонентов	Понимает особенности жизненного цикла компонентов	1. Создать страницу + форму регистрации. Реализовать валидацию по аналогии с классной работой. Добавить проверку на равенство введенных пользователем паролей. 2. Опираясь на компонент Input

			<p>реализовать компонент PasswordInput, добавить в компонент логику просмотра пароля.</p> <p>3. Модифицировать компоненты Input и PasswordInput, что бы они меняли свои стили в зависимости от корректности ввода, валидация поля должна происходить при снятии фокуса с элемента</p>
20.	Текущий контроль Тема 20. Роутинг в реакт приложении	Знает функции и возможности Роутинг в реакт приложении	Настроить роутинг для страниц и навигацию
21.	Текущий контроль Тема 21. Redux toolkit и работа с API	Знает функции и возможности Redux toolkit, документацию API	Настроить подключение redux и реализовать кейс сохранения информации с регистрации пользователя (email, password) в отдельные поля внутри стора. И после нажатия на кнопку регистрации, если все данные валидны, редиректить пользователя на страницу авторизации (заполнять поля информацией из стора email и password)
22.	Текущий контроль Тема 22. Продвинутая работа с хуками	Применяет методы продвинутой работы с хуками	Реализовать приложение со списком дел используя hook
23.	Текущий контроль Тема 23. Основы Node.js	Владеет основами Node.js	Реализовать сбор статистики и создание отчетов на Node.js
	Итоговая аттестация	Реализация способностей в сфере разработки, отладки, проверки работоспособности, модификации программного обеспечения	Зачет/Проверка выполненного итогового задания на компьютере

Практические задания

Задание 1.

Создать npm проект, установить и обновить необходимые пакеты. Разбить js-код на модули

Задание 2

Создать react приложение. Провести базовую настройку инструментов разработчика.

Задание 3

Провести настройку webpack сборки в prod и dev режимы. Конфигурация babel

Задание 4

Создать react-компоненты. Передать необходимые props, добавить типизацию props

Задание 5

Добавить состояние для компонента с помощью хука useState. Рассказать про условия

применения хуков. Добавить базовые обработчики события (onClick, onChange)

Задание 6

Рассказать про container-component подход. Стилизовать компонент с помощью scss модулей и styled-components.

На основе статического списка объяснить принцип работы key-props

Задание 7

С помощью хука useEffect наделить компонент логикой работы с методами жизненного цикла. Затронуть этапы монтирования, обновления и размонтирования компонента.

Задание 8

Настроить роутинг в приложении. Рассказать базовые принципы навигации в SPA

Задание 9

Подключить и настроить redux-toolkit. Рассказать про основные составляющие redux. С помощью API положить данные в redux store и отрендерить их на странице

Задание 10

С помощью хука useRef получить доступ до DOM и провести различные манипуляции с элементом (добавление/удаление атрибутов). Рассказать про мемоизацию компонентов. Применить хуки useMemo и useCallback для оптимизации рендера компонента

Задание 11

Добавить API к текущему проекту

Задание 12

Затронуть этапы монтирования, обновления и размонтирования компонента. Реализовать перехват и обработку ошибки при передаче неправильной ссылки

Задание 13

Дать определение строгой типизации (привести примеры), интерфейсам, модификаторам доступа, абстрактным классам. Рассказать про новые типы в TypeScript. Рассказать про плюсы и минусы TypeScript

Задание 14

Настроить рабочую среду. Подключить утилиты ESLint и Prettier

Задание 15

Провести базовую настройку babel

Задание 16

Выбрать макет с любого ресурса. Создать страницу, с разбиением на логические компоненты (header, content (mainInfo + description + slider + gallery), footer).

Задание 17

Реализовать задачу «Карусель» – ленту изображений, которую можно листать влево-вправо нажатием на стрелочки.

Задание 18

Добавить в приложение несколько страниц, следуя заложенному подходу (контейнер-компоненты).

Задание 19

1. Создать страницу + форму регистрации. Реализовать валидацию по аналогии с классной работой. Добавить проверку на равенство введенных пользователем паролей.
2. Опираясь на компонент Input реализовать компонент PasswordInput, добавить в компонент

логику. просмотра пароля.
3. Модифицировать компоненты Input и PasswordInput, что бы они меняли свои стили в зависимости от корректности ввода, валидация поля должна происходить при снятии фокуса с элемента.

Задание 20

Настроить роутинг для страниц и навигацию.

Задание 21

Настроить подключение redux и реализовать кейс сохранения информации с регистрации пользователя (email, password) в отдельные поля внутри стора. И после нажатия на кнопку регистрации если все данные валидны, редиректить пользователя на страницу авторизации (заполнять поля информацией из стора email и password).

Задание 22

Реализовать приложение со списком дел используя hook

Задание 23

Реализовать сбор статистики и создание отчетов на Node.js

Критерии оценки итоговой аттестации (текущего контроля)

Результат итоговой оценки практического задания	Критерии оценки
Не зачтено	Практическое задание выполнено с большим количеством недочетов
Зачтено	Практические задания выполнены без недочетов

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ МОДУЛЯ

4.1. Учебно-методическое и информационное обеспечение программы:
Слушателю будут доступны презентации и конспекты лекций, записи занятий

4.2. Содержание комплекта учебно-методических материалов.

В период обучения слушателю доступны:

– учебные материалы в виде учебников и учебных пособий, указанных в 4.3.

– учебные материалы для работы в аудитории, размещаемые на портале a-generation.unn.ru

по мере реализации учебной программы по данному модулю.

4.3. Используемые образовательные технологии.

Практические занятия проводятся по схеме мастер-классов с последующим контролем выполнения заданий.

4.4. Литература.

Основная литература по дисциплине

1. <https://learn.javascript.ru/>

2. <https://doka.guide/js/>

3. <https://developer.mozilla.org/ru/docs/Web/JavaScript> - JavaScript Учебные материалы

4. <https://www.youtube.com/watch?v=Bluxbh9CaQ0> - JavaScript Основы для

Начинающих - Полный Курс за 6 часов

5. [https://www.youtube.com/watch?v=Glux40jBsZ8&list=PLcvhF2Wqh7DPD5sRK3lw4bj](https://www.youtube.com/watch?v=Glux40jBsZ8&list=PLcvhF2Wqh7DPD5sRK3lw4bjBsKdgY2bPi)

[BsKdgY2bPi](https://www.youtube.com/watch?v=Glux40jBsZ8&list=PLcvhF2Wqh7DPD5sRK3lw4bjBsKdgY2bPi) - Уроки JavaScript с нуля

6. <https://ru.reactjs.org>

7. <https://learn-reactjs.ru/home>

8. <https://indepth.dev/react>

9. <https://redux.js.org>

10. <https://www.youtube.com/watch?v=GNrdg3PzpJQ>

11. https://www.youtube.com/watch?v=xJZa2_aldDs

4.5. Материально-техническая база

№ п. п.	Наименование модулей (тем, разделов)	Материально-технические условия для реализации программ (наличие лабораторий, производственных участков и т.п. по профилю программы повышения квалификации)
1.	Модуль 1. Программирование на JavaScript (полный курс)	<p>Процессор: Pentium G5xxx (или аналог от AMD) или выше; Оперативная память: 4Gb RAM или выше; Свободное дисковое пространство 200 Мб или выше; Разрешение дисплея 1280x1024 точек или выше; Глубина цвета дисплея 16 бит (65536 цветов) или выше; Колонки, наушники или встроенный динамик;</p> <p>Графическая операционная система (Microsoft Windows 8.1 и выше); Браузер (рекомендуется Microsoft Internet Explorer 11 и выше; Mozilla Firefox (актуальная версия); Google Chrome (актуальная версия)) При использовании других браузеров или более ранних версий, доступ к части функций сервиса может быть ограничен, либо осуществляться некорректно.</p>