

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО»

УТВЕРЖДЕНО
Решением
Ученого совета ННГУ
протокол №14
от «28» декабря 2022г.

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

Программирование на JavaScript

108 часов

Нижний Новгород
2022-2023 учебный год

1. ОБЛАСТЬ ПРИМЕНЕНИЯ

1.1. Цель программы

Совершенствование и получение новых компетенций, повышение профессионального уровня по основам программирования.

1.2. Нормативные документы для разработки программы повышения квалификации:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- профессиональный стандарт 06.001 Программист. Утвержден приказом Министерства труда и социальной защиты Российской Федерации от «18» ноября 2013г. № 679н.

1.3. Категории слушателей на обучение которых рассчитана программа повышения квалификации (далее – Программа):

Программа рассчитана на слушателей с высшим или средним профессиональным образованием, студентов любых курсов.

1.4. Входные требования к обучающимся (в случае необходимости):

Минимальных требований нет

1.5. Сфера применения слушателями полученных профессиональных компетенций, умений и знаний.

Знания, полученные в рамках изучения программы могут быть использованы предприятиях малого и среднего бизнеса, в частном секторе, а также при реализации навыков программирования на фрилансе.

2. ХАРАКТЕРИСТИКА ПОДГОТОВКИ ПО ПРОГРАММЕ

2.1. Нормативный срок освоения программы 108 часов.

2.2. Срок обучения 12 недель

2.3. Общая трудоемкость 3 ЗЕ¹

2.4. Режим обучения 9 часов в неделю

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Слушатель, освоивший программу, должен:

3.1 обладать профессиональными компетенциями, включающими в себя способность:

ПК-1. Способен осуществлять разработку требований и проектирование программного обеспечения.

3.2. знать:

- возможности существующей программно-технической архитектуры
- возможности современных и перспективных средств разработки программных продуктов, технических средств
- методы разработки программного обеспечения и технологии программирования
- методы и технологии проектирования и использования баз данных

3.3. уметь:

- использовать существующие типовые решения и шаблоны проектирования программного обеспечения
- применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов
- решать практические задачи на JavaScript

3.4. владеть:

- разработкой, изменением и согласованием архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения
- проектированием структур данных
- проектированием баз данных

¹ 1 ЗЕ = 36 ак. часов

- проектированием программных интерфейсов
- основами языка JavaScript

3.5. Сфера применения слушателями полученных профессиональных компетенций, умений и знаний.

Знания, полученные в рамках изучения программы могут быть использованы предприятиях малого и среднего бизнеса, в частном секторе, а также при реализации навыков программирования на фрилансе.

4. ТРЕБОВАНИЯ К СТРУКТУРЕ ПРОГРАММЫ

Программа предусматривает изучение следующих модуля Программирование на JavaScript, включающий следующие темы

1. Введение в JavaScript. Переменные и типы данных
2. GitHub. Типы данных. Преобразование типов. Операторы сравнения
3. Преобразование типов. Операторы сравнения. Массивы
4. Условные и логические операторы, циклы
5. Введение в функции
6. Примитивы. Объекты
7. Методы массивов и объектов
8. Продвинутая работа с функциями
9. Прототипы и классы
10. DOM
11. Асинхронность в JavaScript. EventLoop
12. Обработка ошибок. Fetch
13. Введение в продвинутый курс
14. Знакомство с React и TypeScript. Инициализация приложения
15. Знакомство с webpack
16. Компоненты, props, state
17. Хуки и обработка событий
18. Архитектура приложения, key - prop, стилизация компонентов
19. Жизненный цикл компонентов
20. Роутинг в реакт приложении
21. Redux toolkit и работа с API
22. Продвинутая работа с хуками
23. Основы Node.js

Учебный план программы повышения квалификации представлен в Приложении №1 к Макету программы повышения квалификации.

Календарный учебный график программы повышения квалификации представлен в Приложении №2 к Макету программы повышения квалификации.

5. ТРЕБОВАНИЯ К ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

«Программирование на JavaScript — полный курс»

Итоговая аттестация проходит в форме проверки выполненного итогового задания на компьютере. Текущий контроль осуществляется в форме контроля подготовленных слушателем заданий.

Таблица 1

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 прт 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 монтирования, 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 компонентов. Применить хуки useMemo и useCallback для оптимизации рендера компонента
11.	Текущий контроль. Тема 11.	Владеет основами Node.js	Добавить API к текущему проекту

	Асинхронность в JavaScript. EventLoop		
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 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
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

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

6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ ОБЕСПЕЧЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Руководитель программы повышения квалификации:

Усков Алексей Владимирович – ведущий специалист по организационной работе
Института экономики и предпринимательства ННГУ им.Н.И.Лобачевского.

Разработчики программы повышения квалификации:

Усков Алексей Владимирович – ведущий специалист по организационной работе
Института экономики и предпринимательства ННГУ им.Н.И.Лобачевского.

Составители учебно-тематического плана программы повышения квалификации:

Усков Алексей Владимирович – ведущий специалист по организационной работе
Института экономики и предпринимательства ННГУ им.Н.И.Лобачевского.

Сведения о педагогических (научно-педагогических) работниках, участвующих в реализации программы повышения квалификации, и лицах, привлекаемых к реализации дополнительной образовательной программы на иных условиях, представлены в таблице 2.

Таблица 2.

Преподаватели программы повышения квалификации
«Программирование на JavaScript»

№ п/п	Наименование модулей (тем, разделов)	Фамилия, имя, отчество,	Ученая степень, ученое звание	Основное место работы, должность	Место работы и должность по совместительству (если есть)
1.	Модуль. Программирование на JavaScript	Куликов Дмитрий Игоревич	-	Инженер-программист, ООО «Студия Т Г»	

		Широков Алексей Михайлович		Программист ООО «Лад Ай Ти»	
--	--	----------------------------------	--	--------------------------------	--

