Колледж Автономной некоммерческой образовательной организации высшего образования «Научно-технологический университет «Сириус»

Доклад по дисциплине «Введение в специальность

(Информационные технологии)»

на тему «Фреймворки JavaScript для веб-разработки. React и Vue.js»

Выполнил:  
Студент группы К0709-24/2  
Малкова Кристина Андреевна

Принял:

Яковлева Софья Вячеславовна

IT-Колледж «Сириус»  
2025

Оглавление

[Введение 2](#_Toc191898953)

[1 Обзор фреймворков 3](#_Toc191898954)

[1.1 Определение фреймворков 3](#_Toc191898955)

[1.2 Краткая история развития JavaScript-фреймворков 4](#_Toc191898956)

[2 React 5](#_Toc191898957)

[2.1 История создания и развития React 5](#_Toc191898958)

[2.2 Сферы применения React 6](#_Toc191898959)

[2.3 Преимущества использования React 7](#_Toc191898960)

[2.4 Недостатки и ограничения React 8](#_Toc191898961)

[2.5 Примеры использования React в проектах 8](#_Toc191898962)

[3 Vue.js 10](#_Toc191898963)

[3.1 История создания и развития Vue 10](#_Toc191898964)

[3.2 Сферы применения Vue 10](#_Toc191898965)

[3.3 Преимущества использования Vue 11](#_Toc191898966)

[3.4 Недостатки и ограничения Vue 12](#_Toc191898967)

[3.5 Примеры использования Vue в проектах 12](#_Toc191898968)

[4 Фреймворки React и Vue 14](#_Toc191898969)

[4.1 Сравнение React и Vue 14](#_Toc191898970)

[4.2 Выбор между React и Vue 15](#_Toc191898971)

[Заключение 16](#_Toc191898972)

[Литературные источники 1](#_Toc191898973)7

# Введение

В задачи этого доклада входит: изучить историю создания и развития JavaScript фреймворков, исследовать конкретные примеры фреймворков React и Vue.js и сравнить их.

Изучение фреймворков JavaScript является актуальной темой, потому что они автоматизируют работу с пользовательским интерфейсом, обеспечивая удобную разработку динамических страниц, а также служат основными инструментами в современной веб-разработке.

JavaScript — это язык программирования, который позволяет реализовывать сложные элементы на веб-странице. Каждый раз, когда на странице происходит что-то большее, чем простое статичное отображение, можно с уверенностью сказать, что в этом задействован JavaScript. Он является третьим уровнем (среди HTML и CSS) в структуре стандартных веб-технологий. С ростом популярности JavaScript, ставшего одним из ведущих инструментов при разработке веб-приложений, возникла необходимость в разработке специализированных фреймворков для JavaScript. Эти инструменты направлены на упрощение управления сложностью кода и оптимизацию процесса разработки, что стало ключевым фактором в современном веб-разработке.

# 

# 1 Обзор фреймворков

## 1.1 Определение фреймворков

Фреймворк — это структура или основа в программировании, представляющая собой готовую модель, которая позволяет быстро разрабатывать приложения, добавляя к ней собственный код. Создание и использование фреймворков представляет собой каркасный подход. Понятие фреймворка иногда путают с понятием библиотеки, хотя они имеют различные значения. В отличие от фреймворка, библиотека может быть представлена в проекте как набор подпрограмм с похожей функциональностью и не влияет на архитектуру проекта, не накладывая на неё никаких ограничений. Фреймворк же определяет правила для построения архитектуры приложения, устанавливая поведение по умолчанию уже на начальном этапе разработки. Он служит основой, которую необходимо будет расширять и модифицировать в соответствии с заданными требованиями.

## 1.2 Краткая история развития JavaScript-фреймворков

JavaScript появился в 1995 году благодаря Брендану Эйху из компании Netscape. Вскоре Microsoft создала JScript для Internet Explorer 3 (1996), что вызвало конкуренцию между браузерами и привело к необходимости стандартизации.

В 2005 году технология AJAX изменила веб-разработку, позволив обновлять данные на страницах без перезагрузки. В 2006 году вышла библиотека jQuery, которая упростила работу с JavaScript. В 2010-х годах начали активно развиваться JavaScript-фреймворки.

В 2010 году появились Backbone.js и AngularJS, обеспечивающие двустороннее связывание данных. В 2013 году Эван Ю создал Vue.js, который быстро стал популярным благодаря своей простоте и скорости.

Современный этап развития начался в 2016 году с выпуском Angular 2, поддерживающего TypeScript. В 2018 году React внедрил хуки, упростив процесс разработки. В 2020-х годах появились новые фреймворки, такие как Svelte и Solid.js, которые сделали код еще более эффективным.

# 

# 2 React

Фреймворки могут содержать в себе библиотеки. Например, фреймворк Next.js строится на библиотеке React, но за счет того, что React очень масштабная библиотека, его зачастую используют как полноценный фреймворк, пусть он и не имеет некоторых свойств присущих фреймворкам (строгая файловая структура, маршрутизации и т.п.)

## 2.1 История создания и развития React

В 2011 году Джордан Уолк, инженер Facebook, создал прототип React для улучшения работы с DOM. Официально React был представлен в мае 2013 года на JSConf. Первая версия сосредоточилась на динамических пользовательских интерфейсах и представила JSX, позволяющий интегрировать HTML в JavaScript. Основной особенностью стал виртуальный DOM (объектная модель документа), обеспечивающий быструю производительность.

В 2016 году в React 15 была оптимизирована обработка обновлений, улучшив отзывчивость веб-приложений. В 2017 году появилось ядро React Fiber, которое ускорило сложные обновления. В 2019 году с React 16.8 были введены хуки, а в 2022 году React 18 предложил новые функции, такие как параллельный рендеринг и автоматическая пакетная обработка для повышения производительности.

## 2.2 Сферы применения React

React.js – компонент, который используется разработчиками при написании одностраничных и многостраничных приложений, а также в процессе разработки веб-страниц (преимущественно крупных).

Рассматриваемый компонент разработки используется для:

1. Создания функциональных интерактивных веб-интерфейсов, а также для разработки отдельных компонентов и страниц. С ними не нужно будет в процессе работы постоянно обновлять страницы
2. Простейшей разработки сложных программных структур. Благодаря React можно описать их просто и быстро.
3. Доработки новых функциональных возможностей проектов с любым изначальным технологическим стеком. Это связано с тем, что библиотека будет прекрасно работать, независимо от того, на чем был изначально написан код программы или сайта.
4. Разработки многостраничных и одностраничных (PWA и SPA) приложений. Так называют проекты, которые работают подобно обычным программам, веб-сервисы.
5. Работы с серверной частью проекта клиент-серверного типа.

## 2.3 Преимущества использования React

1. React JS композиционный — код разбивается на компоненты, делая его удобным и гибким. JSX (синтаксическое расширение для JavaScript), сочетая JavaScript и HTML-подобный синтаксис, упрощает создание шаблонов.
2. React декларативен — UI обновляется при изменении состояния без прямого взаимодействия с DOM, что упрощает разработку и отладку. Можно добавлять новые функции без переписывания кода. Он поддерживает серверный рендеринг (Node) и мобильные приложения (React Native), позволяя разрабатывать сразу для IOS, Android и веба.
3. React включает готовые шаблоны, ускоряя создание масштабируемых проектов. В отличие от Angular и Ember, достаточно базовых знаний HTML и CSS.
4. React оптимизирован для SEO (поисковая оптимизация), помогая разработчикам обеспечивать доступ к нужному контенту для пользователей. Когда пользователь ищет что-то, поисковые системы, такие как Google или Bing, ищут наиболее релевантную страницу.
5. React JS обеспечивает стабильность кода благодаря однонаправленному потоку данных. Это означает, что изменения в дочерних компонентах не влияют на родительские.
6. React имеет сильное сообщество поддержки (Facebook и сообщество разработчиков)

## 2.4 Недостатки и ограничения React

Несмотря на то, что у React много преимуществ, есть и недостатки. Вот некоторые из них:

1. Отсутствие качественной документации. Документация React на официальном сайте охватывает основные концепции, но не предоставляет глубоких и подробных объяснений с множеством примеров, а также статьи по разным уровням сложности не отфильтрованы по категориям.
2. Нет единого стандарта для разработки приложений. React представляет собой библиотеку для пользовательского интерфейса, которая включает в себя различные концепции и рекомендации, однако не существует универсального метода создания приложений.
3. Быстрый темп развития. Разработчики должны быстро осваивать новые концепты, чтобы поддерживать стабильность приложения, так как React каждый год добавляет новые функции, изменяя старые.
4. Продвинутое использование JavaScript. Основные концепции React просты, но более сложные требуют знаний продвинутых возможностей JavaScript.
5. React — это только UI-библиотека, а не полноценный фреймворк. Для хорошей архитектуры приложения необходимо правильно выбрать сторонние библиотеки, так как React не навязывает конкретную архитектуру приложения.

## 2.5 Примеры использования React в проектах

React.js широко применяется в различных проектах благодаря своей гибкости и производительности.

1. React был разработан инженерами Facebook и активно используется в интерфейсе этой социальной сети. Он обеспечивает динамическое обновление ленты новостей и других компонентов без перезагрузки страницы.
2. Netflix использует React для обеспечения быстрого и плавного взаимодействия пользователей с интерфейсом, что особенно важно при работе с большим объемом контента.
3. Платформа для бронирования жилья Airbnb применяет React для создания интерактивных элементов, таких как карты и фильтры поиска, что улучшает удобство использования сервиса.
4. Веб-версия популярного мессенджера WhatsApp Web использует React для обеспечения быстрого и надежного обмена сообщениями через браузер.

# 

# 3 Vue

## 3.1 История создания и развития Vue

Vue.js создан Эваном Ю в 2013 году как более легкая и простая альтернатива AngularJS. Он решил объединить лучшие аспекты существующих технологий.

Первая версия Vue была выпущена в феврале 2014 года и с тех пор фреймворк начал активно развиваться. В октябре 2015 года вышла версия 1.0, которая предоставила основные функции для создания одностраничных веб-приложений, загружающихся на одной HTML-странице. В сентябре 2016 года была выпущена версия 2.0, которая добавила поддержку серверного рендеринга (SSR), что расширило возможности создания приложений с улучшенной производительностью и оптимизацией для поисковых систем (SEO). 18 сентября 2020 года была представлена версия Vue 3.0, ставшая важным этапом в развитии фреймворка. В этой версии был оптимизирован размер пакетов, что сделало Vue.js более легким и быстрым. Также улучшилась интеграция с TypeScript, что повысило совместимость и удобство работы с типами данных. Кроме того, были добавлены новые API, позволяющие более эффективно решать задачи при разработке крупных и сложных приложений. Существенные улучшения в производительности позволили Vue.js лучше работать на мобильных устройствах и с большими объемами данных.

## 3.2 Сферы применения Vue

Vue — прогрессивный JavaScript-фреймворк, широко используемый для разработки различных веб-приложений и интерфейсов. Основные сферы его применения включают:

1. Одностраничные приложения. Vue.js отлично подходит для разработки динамичных одностраничных сайтов с интерактивными интерфейсами.
2. Высоконагруженные приложения. Этот фреймворк используется для создания ресурсов с высокой нагрузкой, таких как интернет-магазины и информационные порталы.
3. Интеграция с CMS (система управления содержимым). Vue.js применяется для разработки интерфейсов сайтов, которые затем интегрируются с популярными системами управления контентом.
4. Отдельные функциональные блоки. Фреймворк подходит для создания отдельных компонентов, таких как формы авторизации, заявки, чаты и другие интерактивные элементы.
5. Адаптивные интерфейсы. Vue.js позволяет разрабатывать интерфейсы, которые подстраиваются под различные устройства и разрешения экранов.

Благодаря своей гибкости и простоте, Vue.js продолжает набирать популярность и широко применяется как в небольших проектах, так и в крупных корпоративных решениях

## 3.3 Преимущества использования Vue

Основная характеристика Vue.js — его реактивность. Разработчик получает не только шаблоны, но и возможность отслеживать изменения в реальном времени. Это исключает обновление страницы, и новые данные мгновенно доступны пользователю.

У Vue есть и другие достоинства:

1. Низкий порог входа. Фреймворк легко освоить, зная JavaScript, CSS и HTML. Также хорошая и подробная документация, однако недостаточно полная на русском языке.
2. Компактный размер. Всего 20 Кб, благодаря чему приложения загружаются быстро и работают эффективно.
3. Модульное внедрение. Vue можно внедрять для отдельных компонентов, например: интерфейс чата или форма авторизации. Он легко интегрируется с кодом на других фреймворках, позволяя плавно переходить с устаревших технологий.

## 3.4 Недостатки и ограничения Vue

Выделяют несколько недостатков Vue:

1. Ограниченное количество профессиональных ресурсов. По сравнению с другими фреймворками, помимо официального сайта Vue.js, найти материалы для изучения, особенно на русском языке, довольно сложно.
2. Проблемы с гибкостью. Сложности возникают при интеграции Vue в крупные проекты. Из-за отсутствия оптимального решения разработчикам приходится использовать нестандартные подходы. Однако этот нюанс редко затрагивает начинающих программистов, так как Vue чаще применяется в небольших и средних проектах, где подобных проблем не возникает.

## 3.5 Примеры использования Vue в проектах

Vue.js широко применяется в различных проектах благодаря своей гибкости и простоте. Ниже представлены конкретные примеры использования Vue.js:

1. Chess.com — ведущий мировой онлайн-сервис для игры в шахматы. Разработчики ресурса заменили AngularJS на Vue, что позволило повысить удобство работы с кодом и скорость разработки. Благодаря этому, основатель платформы смог эффективнее взаимодействовать с удаленной командой, а реализация новых функций заняла меньше времени.
2. Codeship — облачная платформа для непрерывной интеграции, которую используют Red Bull, CNN и Product Hunt. Переход с jQuery на Vue.js помог устранить сбои, повысить стабильность и упростить поддержку интерфейса.
3. Behance — один из самых популярных сайтов для дизайнеров, на котором они демонстрируют свои работы. Vue.JS используют в качестве языка программирования для фронтенда.

# 4 Фреймворки React и Vue

React и Vue.js — это два широко известных фреймворка на JavaScript, предназначенные для создания пользовательских интерфейсов. Хотя у них общие цели, они отличаются методами и характеристиками. В дальнейшем приведено сравнение этих фреймворков, а также советы по их выбору в зависимости от конкретных ситуаций

## 4.1 Сравнение React и Vue

1. Происхождение и поддержка: React был разработан и поддерживается компанией Facebook (теперь Meta), что гарантирует его надежность и регулярные обновления. Vue.js был создан Эваном Ю, который ранее работал в Google, и продолжает развиваться благодаря активному сообществу. Несмотря на отсутствие поддержки от крупной корпорации, Vue получил значительное признание среди разработчиков
2. Архитектура и масштабируемость: React предлагает возможность выбора архитектуры и сторонних библиотек, что делает его подходящим для масштабных и сложных проектов. Однако для этого команде необходимо иметь опыт и знания для принятия архитектурных решений. В свою очередь, Vue.js предоставляет более организованный подход "из коробки", что облегчает разработку небольших и средних проектов. Он позволяет быстро начать работу и подходит для команд с ограниченным опытом
3. Синтаксис и обучение: React применяет JSX, что объединяет JavaScript и HTML в одном документе. Это может показаться необычным для тех, кто только начинает, но предоставляет широкие возможности для разработки. В то же время Vue.js использует шаблоны, основанные на HTML, что делает его более доступным и понятным для начинающих программистов
4. Производительность: Оба фреймворка показывают высокую производительность, и выбор между ними в этом аспекте зависит от специфики проекта и оптимизации кода.

## 4.2 Выбор между React и Vue

Выбирайте React, если:

* 1. Ваш проект крупный и требует высокой масштабируемости.
  2. Необходима гибкость в выборе архитектуры и интеграции с другими библиотеками.
  3. Команда обладает опытом работы с JavaScript и готова к изучению JSX.

Выбирайте Vue.js, если:

* 1. Вы работаете над небольшим или средним проектом с ограниченным временем на разработку.
  2. Требуется быстрое прототипирование и простота интеграции.
  3. Команда состоит из новичков или разработчиков с небольшим опытом в JavaScript.

Также стоит отметить, что окончательный выбор между React и Vue.js зависит от конкретных требований проекта, опыта команды и долгосрочных целей разработки.

# Заключение

Таким образом, была изучена история развития JavaScript фреймворков и разобраны характеристики фреймворков React и Vue.

Фреймворки предоставляют разработчикам полноценную платформу, включающую разнообразные инструменты и функции, необходимые для создания приложений и проведения исследовательских проектов. Каждый фреймворк предназначен для определённых задач и предлагает разные возможности.

Как Vue, так и React являются отличными основами для разработки динамических пользовательских интерфейсов. При принятии решения стоит учитывать такие аспекты, как сценарии использования, бизнес-цели и т.д., чтобы определить, что будет более подходящим для проекта.

# Литературные источники

1. Danielle Adams JavaScript исполнилось 25 лет / Danielle Adams [Электронный ресурс] // Хабр: [сайт]. — URL: https://habr.com/ru/companies/ruvds/articles/534104/ .
2. React: характеристика, особенности, области применения / [Электронный ресурс] // Otus: [сайт]. — URL: <https://otus.ru/journal/react-harakteristika-osobennosti-oblasti-primeneniya/> .
3. What are the advantages of React.js / [Электронный ресурс] // Geeksforgeeks: [сайт]. — URL: https://www.geeksforgeeks.org/what-are-the-advantages-of-react-js/ .
4. ReactJS - Advantages & Disadvantages / [Электронный ресурс] // Tutorialspoint : [сайт]. — URL: <https://www.tutorialspoint.com/reactjs/reactjs_advantages_and_disadvantages.htm> .
5. Vue JavaScript: описание и особенности / [Электронный ресурс] // Otus: [сайт]. — URL: <https://otus.ru/journal/vue-javascript-opisanie-i-osobennosti/> .
6. Денис Бородовский JavaScript-фреймворк Vue.js: особенности и примеры реализации / Денис Бородовский [Электронный ресурс] // Highload: [сайт]. — URL: <https://highload.tech/javascript-vue-js/?utm_source> .
7. React или Vue: что выбрать бизнесу в 2025? / [Электронный ресурс] // Worksolutions: [сайт]. — URL: https://worksolutions.ru/useful/React-ili-Vue-chto-vybrat-biznesu/?utm\_source .