

АНАЛИЗ ТЕХНОЛОГИЙ И ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ПРИ СОЗДАНИИ ГРАФИЧЕСКИХ ВЕБ-ИНТЕРФЕЙСОВ

Нгуен Т.З.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Дик С.К. – к. ф.-м. н., доцент, зав. кафедрой ИКТ

Аннотация. В статье рассматриваются особенности использования технологий HTML/CSS/JavaScript, а также языков программирования TypeScript и Dart для создания графического пользовательского интерфейса веб-приложений. Проведен сравнительный анализ функциональных возможностей, архитектурных подходов, удобства разработки, производительности и масштабируемости решений. Определены преимущества и ограничения каждого из рассматриваемых инструментов в контексте проектирования современных веб-интерфейсов.

Ключевые слова: веб-интерфейс, HTML, CSS, JavaScript, TypeScript, Dart, веб-разработка, фронтенд

Введение. В условиях стремительного развития информационных технологий и цифровых платформ веб-сайты и веб-приложения стали неотъемлемой частью жизни современного общества. Современные информационные системы требуют удобного, адаптивного и функционального графического интерфейса, обеспечивающего эффективное взаимодействие человека с программным обеспечением. Для создания высококачественного веб-интерфейса требуется использование различных языков программирования и технологий, каждая из которых имеет свои особенности и область применения. Наиболее распространёнными языками программирования для разработки графических интерфейсов и веб-приложений в настоящее время являются HTML/CSS/JavaScript, TypeScript и Dart благодаря их качеству, удобству использования и значительным преимуществам, которые они предоставляют программистам, компаниям и пользователям. Однако каждый язык программирования имеет свои собственные преимущества и недостатки. Каждый язык программирования подходит для веб-сайтов с различными требованиями и задачами. Поэтому анализ и сравнение этих инструментов с целью определения их возможностей и целесообразности использования при создании графического веб-интерфейса является весьма важным.

Основная часть. В программировании и проектировании веб-интерфейсов HTML, CSS и JavaScript являются наиболее распространённым набором технологий. HTML представляет собой язык разметки, предназначенный для построения структуры и формирования содержимого веб-страницы. Он включает различные логические элементы, такие как заголовки, абзацы, таблицы, формы ввода данных, изображения и другие элементы. Благодаря этому разработчики могут создавать полноценную структурированную основу веб-интерфейса. CSS используется для описания внешнего вида и оформления веб-страниц. Он определяет стиль элементов, управляет цветовой гаммой, шрифтами, блочной компоновкой, а также позволяет адаптировать интерфейс веб-страницы к различным устройствам пользователя (например, интерфейс веб-страницы на ноутбуке будет отличаться от интерфейса при открытии на мобильном телефоне). Для обеспечения интерактивности и создания анимаций, а также различных динамических действий используется JavaScript. Этот язык позволяет системе оперативно реагировать на действия пользователя и выполнять логические алгоритмы различной сложности за короткое время. Структура веб-страницы, разработанной с использованием технологий HTML, CSS и JavaScript, проиллюстрирована на рисунке 1.

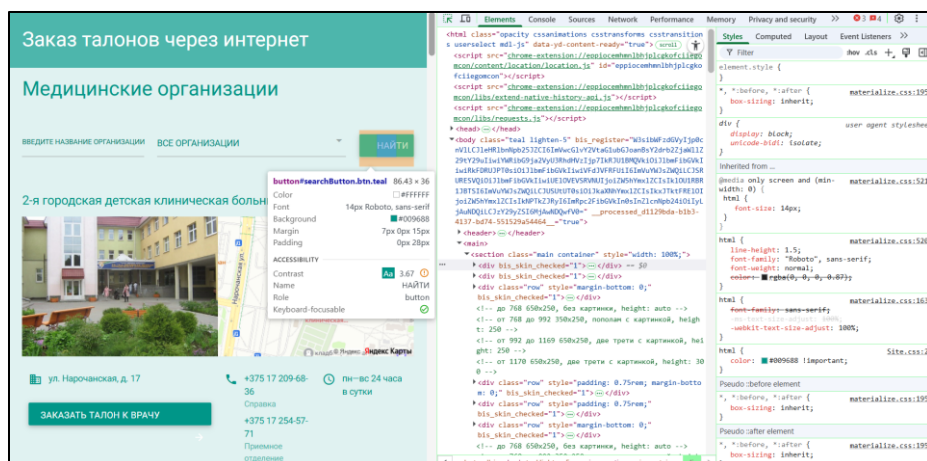


Рисунок 1 – Структура веб-страницы на основе HTML CSS и JavaScript

Сочетание HTML, CSS и JavaScript формирует пользовательский интерфейс веб-приложения. Основным преимуществом данных технологий является их универсальность и широкая поддержка всеми современными веб-браузерами. Благодаря этому пользователям не требуется устанавливать дополнительное программное обеспечение для работы с веб-приложениями, что значительно упрощает их использование и повышает доступность. Веб-приложения, созданные на основе HTML, CSS и JavaScript, могут функционировать на различных операционных системах и устройствах, включая настольные компьютеры, планшеты и смартфоны. Это делает данные технологии востребованными в условиях стремительного развития интернета и увеличения числа пользователей мобильных устройств.

Важную роль в развитии веб-разработки сыграло появление большого количества библиотек и фреймворков, которые значительно расширяют возможности стандартного JavaScript. Среди наиболее известных и широко используемых инструментов можно отметить React, Vue и Angular. Они предоставляют разработчикам удобные механизмы для построения сложных пользовательских интерфейсов, управления состоянием приложения и организации структуры кода. Благодаря применению данных фреймворков становится возможным создание современных одностраничных приложений, которые отличаются высокой скоростью работы, удобством использования и высоким уровнем интерактивности. Кроме того, данные технологии обеспечивают высокую адаптивность пользовательского интерфейса. Это особенно важно в условиях широкого распространения мобильных устройств, когда веб-сайты и веб-приложения должны корректно работать на экранах различных размеров и разрешений. Использование адаптивного дизайна позволяет автоматически подстраивать расположение элементов интерфейса под характеристики устройства пользователя, обеспечивая комфортное взаимодействие с веб-приложением.

Несмотря на многочисленные преимущества, HTML, CSS и JavaScript имеют и определённые ограничения. HTML и CSS не являются полноценными языками программирования, поскольку они предназначены в первую очередь для разметки и оформления веб-страниц и не поддерживают реализацию сложной логики обработки данных. JavaScript, хотя и является полноценным языком программирования, при использовании исключительно на стороне клиента может сталкиваться с определёнными ограничениями, связанными с безопасностью, производительностью и доступом к системным ресурсам. В связи с этим при разработке полноценных веб-приложений обычно используется сочетание клиентских технологий с серверными языками программирования и системами обработки данных, что позволяет создавать более функциональные, безопасные и масштабируемые информационные системы.

TypeScript представляет собой надмножество языка JavaScript, которое расширяет его возможности за счёт введения статической типизации и дополнительных инструментов разработки. Благодаря этому разработчики могут выявлять многие ошибки на этапе компиляции, что повышает надёжность и поддерживаемость программного кода. TypeScript широко используется как при разработке клиентской части веб-приложений, так и на стороне сервера с использованием среды выполнения Node.js. Он играет важную роль в создании

логики пользовательского интерфейса и часто применяется совместно с современными фреймворками и библиотеками, такими как Angular, React и Vue.

Использование TypeScript особенно эффективно при разработке крупных и сложных веб-проектов, где большое количество компонентов и взаимосвязей между ними требует строгой структуры и хорошей организации кода. Благодаря поддержке модульности, интерфейсов и объектно-ориентированного программирования разработчики могут создавать более понятную и масштабируемую архитектуру приложений. В результате применение TypeScript способствует повышению качества программного обеспечения, улучшению командной разработки и упрощению дальнейшего сопровождения и развития веб-приложений. Typescript – это язык программирования, разработанный Microsoft. Он основан на JavaScript и также используется для фронтенд-разработки. Хотя TypeScript имеет сходство с JavaScript, его особенность – статическая система типизации данных. JavaScript – динамически типизированный язык, в котором типы переменных определяются во время выполнения. TS позволяет разработчикам объявлять типы статически. Типизация объектов TypeScript означает, что на этапе разработки можно указать типы данных, параметры функций и свойства объекта. Это помогает создавать более безопасный и надежный код [1]. К недостаткам TypeScript можно отнести необходимость дополнительного этапа компиляции перед выполнением программы, что может усложнять процесс сборки проекта. Также при работе с большими проектами система типизации требует более тщательного проектирования структуры данных, что может увеличивать время первоначальной разработки.

Язык программирования Dart также широко используется в современном процессе веб-разработки. Он активно применяется в экосистеме Flutter, что позволяет создавать кроссплатформенные пользовательские интерфейсы, работающие как в мобильных, так и в веб-приложениях. Одной из целей разработчиков языка было создание, по сути, улучшенной версии JS, которая эффективнее решала бы практические задачи программирования [2]. На рисунке 2 представлена структура веб-страницы, разработанной на языке Dart в экосистеме Flutter.

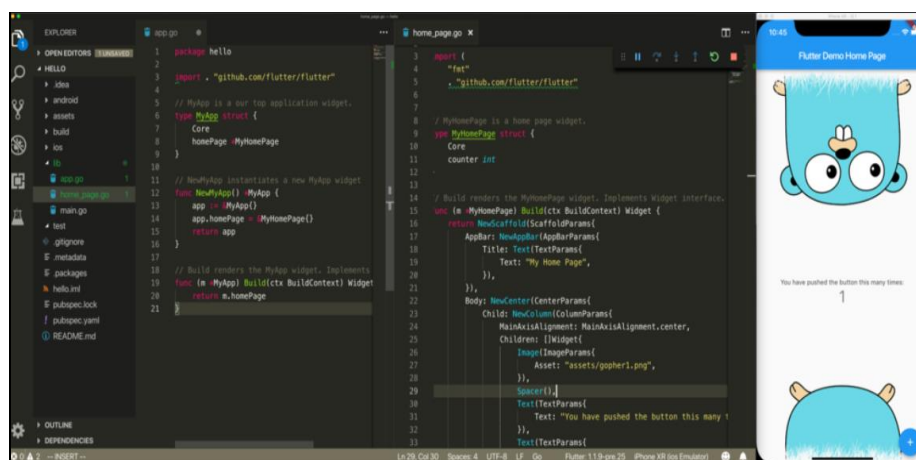


Рисунок 2 – Структура веб-страницы на основе Dart

В архитектуре веб-приложений Dart может использоваться для реализации логики приложения, обработки данных и взаимодействия с серверными сервисами. К преимуществам Dart относятся высокая производительность, стабильность работы при больших нагрузках и наличие современных механизмов безопасности. Кроме того, строгая система типизации и объектно-ориентированная модель программирования способствуют созданию структурированного, надёжного и масштабируемого программного кода. Это делает данный язык подходящим для разработки крупных информационных систем и сложных программных решений. Однако разработка на Dart может требовать более глубокого понимания его экосистемы, а настройка среды разработки и процесс развёртывания приложений иногда оказываются более сложными по сравнению с некоторыми другими языками программирования.

При сравнении HTML, CSS и JavaScript, а также TypeScript и Dart следует учитывать их роль в архитектуре веб-приложений. HTML, CSS и JavaScript непосредственно формируют структуру, внешний вид и интерактивность пользовательского интерфейса и являются

основными технологиями клиентской части веб-приложений. TypeScript используется для разработки более сложной логики интерфейса, обеспечивая статическую типизацию и улучшенную структуру программного кода. Это способствует повышению надёжности и упрощает сопровождение крупных проектов. Dart, в свою очередь, широко применяется в экосистеме Flutter для создания кроссплатформенных пользовательских интерфейсов, включая веб-приложения. Благодаря высокой производительности и развитой системе типизации Dart позволяет эффективно разрабатывать масштабируемые программные решения и сложные пользовательские интерфейсы.

Таким образом, различия между данными технологиями в основном заключаются в области их применения и архитектурной роли. HTML, CSS и JavaScript служат базовой основой для построения веб-интерфейсов, тогда как TypeScript и Dart выступают инструментами для повышения эффективности разработки, структурирования программного кода и создания более сложных и функциональных пользовательских приложений.

Заключение. В рамках проведенного анализа было подробно рассмотрено применение веб-языков программирования HTML/CSS/JavaScript, TypeScript и Dart при разработке пользовательских интерфейсов веб-приложений и программных систем. Каждая из рассмотренных технологий обладает своими преимуществами и недостатками и предоставляет программистам и разработчикам программного обеспечения различные возможности выбора в зависимости от конкретных задач и целей разработки. HTML/CSS/JavaScript представляют собой набор технологий, используемых для создания структуры веб-страниц, их визуального оформления и организации взаимодействия с пользователем. В свою очередь, TypeScript и Dart предоставляют разработчикам дополнительные инструменты для более удобной организации программного кода, а также уделяют внимание вопросам надежности и безопасности программных решений. Выбор языков программирования определяется требованиями проекта, его масштабом, бюджетом и уровнем сложности разрабатываемой системы. В современной практике разработки веб-приложений часто применяется комбинированное использование различных языков и инструментов программирования, что позволяет эффективно использовать их преимущества и минимизировать существующие недостатки, обеспечивая создание программных продуктов высокого качества.

Список литературы

1. Язык программирования TypeScript: его преимущества и недостатки, отличия от JavaScript [Электронный ресурс]. – Режим доступа : <https://gitverse.ru/blog/articles/development/190-yazyk-programmirovaniya-typescript-ego-preimushestva-i-nedostatki-otlichiya-ot-javascript>
2. Язык программирования Dart [Электронный ресурс]. – Режим доступа : <https://appfox.ru/blog/yazyk-programmirovaniya-dart/>

UDC 004.047

ANALYSIS OF TECHNOLOGIES AND PROGRAMMING LANGUAGES IN THE CREATION OF GRAPHIC WEB INTERFACES

Nguyen T.D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Dik S.K. – Cand. of Sci., Associate Professor, Chair of the department of ECG

Annotation. This article examines the use of HTML/CSS/JavaScript technologies, as well as the programming languages TypeScript and Dart, to create graphical user interfaces for web applications. A comparative analysis of the solutions' functionality, architectural approaches, development ease, performance, and scalability is provided. The advantages and limitations of each of the tools under consideration are identified in the context of designing modern web interfaces.

Keywords: web interface, HTML, CSS, JavaScript, TypeScript, Dart, web development, frontend