

3. Военное дело.

Беспилотные летательные аппараты (БПЛА) и наземные роботы активно используются в военных операциях. Распознавание роботов позволяет координировать их действия и предотвращать столкновения.

4. Домашняя робототехника.

Интеллектуальные домашние помощники, такие как роботы-пылесосы и кухонные роботы, должны уметь ориентироваться в домашней обстановке. Распознавание роботов помогает им избегать препятствий и выполнять свои функции безопасно.

Заключение

Алгоритмы распознавания роботов играют важную роль в развитии современной робототехники. Они обеспечивают безопасное взаимодействие между людьми и машинами, повышают точность выполнения задач и открывают новые возможности для автоматизации. Продолжающееся развитие технологий машинного обучения и компьютерного зрения обещает еще больше улучшить эти алгоритмы и расширить область их применения.

Список использованных источников:

1. Колючкин В.Я. Алгоритмы обработки информации в системах технического зрения промышленных роботов / В.Я. Колючкин, Н.К. Минь, Ч.Т. Хай // Инженерный журнал: наука и инновации. – 2013. – № 9 (21). – 42 с.
2. Беликов В.А. Системы машинного зрения мобильных роботов / Беликов В.А. [и др.] // Известия Волгоградского государственного технического университета. – 2014. – № 25. – С. 184–190.
3. Есенбаева Г.Б. Техническое зрение для мобильных роботов / Г.Б. Есенбаева // ББК 3 П27. – 2019. – 73 с.
4. Егорцев М.В. Алгоритмическое обеспечение системы внешнего наблюдения и маршрутизации автономных мобильных роботов / М.В. Егорцев, С.А.К. Диане, Н.Д. Кац // Russian Technological Journal. – 2021. – Т. 9. – № 3. – С. 15–23.
5. Нехороших С.А. Распознавание объектов для решения задачи их сортировки мобильным роботом / С.А. Нехороших, А.В. Востров, Л.М. Курочкин // Математические методы в технике и технологиях-ММТТ. – 2019. – Т. 12. – С. 104–107.

ОСНОВЫ РАЗРАБОТКИ ПРИЛОЖЕНИЙ ДЛЯ ANDROID/iOS

Д.С. Родионова², ученица 9 А класса,

Научный руководитель: Разумников С.В.^{1а}, к.т.н., доц.

¹Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

²МБОУ «СОШ № 14 имени К.С. Федоровского»

652057, Кемеровская обл., г. Юрга, ул. Волгоградская, 3

E-mail: ^arazumnikov@tpu.ru

Аннотация: В статье рассматриваются основы разработки приложений для Android/iOS.

Ключевые слова: разработка, приложения, Android/iOS.

Abstract: This article covers the basics of developing applications for Android/iOS.

Keywords: development, applications, Android/iOS.

Введение. В современном мире мобильные приложения стали неотъемлемой частью нашей жизни. Они помогают нам решать множество задач: от общения до управления финансами, от работы до развлечений. Разработка мобильных приложений – это востребованная область IT-индустрии, которая требует знаний в различных сферах программирования, дизайна и маркетинга. В этой статье мы рассмотрим основы создания приложений для двух самых популярных платформ – Android и iOS.

1. Выбор платформы.

Перед началом разработки необходимо определиться с целевой платформой. Основными игроками на рынке мобильных операционных систем являются Android и iOS.

Каждая платформа имеет свои особенности:

- Android: Операционная система от Google, используемая на большинстве смартфонов и планшетов. Приложения для Android разрабатываются преимущественно на языке Java или Kotlin.

- iOS: Операционная система от Apple, установленная на устройствах компании, таких как iPhone и iPad. Для разработки под iOS используются языки Swift и Objective-C.

Выбор платформы зависит от аудитории вашего приложения, бюджета проекта и технических требований. Если вы хотите охватить максимальное количество пользователей, стоит рассмотреть возможность кросс-платформенной разработки, о которой поговорим позже.

2. Инструменты и среды разработки.

Для разработки приложений вам понадобятся специальные инструменты и среды разработки:

Android

- Android Studio: Официальная интегрированная среда разработки (IDE) от Google. Поддерживает работу с языками Java и Kotlin.

- Gradle: Система сборки проектов, встроенная в Android Studio.

- SDK (Software Development Kit): Набор инструментов и библиотек для разработки приложений под Android.

iOS

- Xcode: Официальная IDE от Apple для разработки приложений под iOS. Работает исключительно на macOS.

- SwiftUI/UIKit: Фреймворки для создания пользовательских интерфейсов. – CocoaPods/Carthage/Swift Package Manager: Менеджеры зависимостей для iOS-проектов.

3. Языки программирования.

Как уже упоминалось выше, выбор языка программирования зависит от выбранной платформы:

Android

- Java: Традиционный язык для разработки под Android. Несмотря на появление Kotlin, многие проекты продолжают использовать Java.

- Kotlin: Современный язык программирования, который стал официальным языком для разработки под Android. Обладает множеством преимуществ перед Java, включая лаконичный синтаксис и лучшую совместимость с современными фичами.

iOS

- Objective-C: Один из первых языков, используемых для разработки под iOS. Сейчас постепенно уступает позиции Swift.

- Swift: Новый язык программирования от Apple, который быстро завоевал популярность благодаря своей простоте и производительности.

4. Создание пользовательского интерфейса.

Создание удобного и интуитивно понятного интерфейса – одна из ключевых задач при разработке мобильного приложения. Оба Android и iOS предоставляют мощные инструменты для этого:

Android

- XML Layouts: Основной способ описания интерфейсов в Android-приложениях. XML-файлы содержат информацию о расположении элементов интерфейса.

- Material Design: Концепция дизайна от Google, включающая рекомендации по созданию современного и привлекательного интерфейса.

iOS

- Storyboards/XIB Files: Визуальные редакторы интерфейсов в Xcode. Позволяют создавать интерфейс без необходимости писать код вручную.

- Auto Layout: Механизм для автоматического размещения элементов интерфейса с учетом разных размеров экранов.

5. Кросс-платформенная разработка.

Если вы хотите создать приложение сразу для обеих платформ, то стоит обратить внимание на кросс-платформенные решения:

- React Native: Популярная платформа от Facebook, позволяющая создавать нативные приложения на базе JavaScript.

- Flutter: Платформа от Google, использующая язык Dart. Flutter позволяет создавать красивые и высокопроизводительные приложения.

- Xamarin: Решение от Microsoft, основанное на C#. Подходит для разработчиков, знакомых с .NET.

Кросс-платформенная разработка может значительно сократить затраты времени и ресурсов, но важно учитывать возможные ограничения и компромиссы.

6. Тестирование и публикация.

После завершения разработки приложения необходимо провести тщательное тестирование. Это включает в себя функциональное тестирование, тестирование юзабилити, а также тестирование на различных устройствах и версиях операционной системы.

Когда все ошибки исправлены, можно приступить к публикации приложения в магазинах:

- Google Play Store: Магазин приложений для Android.

- App Store: Магазин приложений для iOS.

Процесс публикации включает в себя подготовку необходимых материалов (иконки, скриншоты, описание), а также прохождение модерации.

Заключение

Разработка мобильных приложений – это увлекательный процесс, требующий глубоких знаний в области программирования, дизайна и маркетинга. Важно понимать особенности каждой платформы и выбирать подходящие инструменты и технологии. Независимо от того, какую платформу вы выберете, главное – создать качественное и полезное приложение, которое будет радовать ваших пользователей.

Список использованных источников:

1. Васильев Н.П. Универсальные технологии разработки мобильных приложений / Н.П. Васильев // Системы и технологии: теория и практика. – 2017. – 26 с.

2. Ахметов А.К. Операционная система Android: история создания и развития. Разработка приложений для платформы Android / А.К. Ахметов // Скиф. Вопросы студенческой науки. – 2017. – № 9. – С. 108–115.

3. Федотов В.А. Разработка мобильных приложений / В.А. Федотов // Научный электронный журнал Меридиан. – 2021. – № 2. – С. 30–32.

4. Аксенов К.В. Обзор современных средств для разработки мобильных приложений / К.В. Аксенов // Новые информационные технологии в автоматизированных системах. – 2014. – № 17. – С. 508–513.

5. Мытников А.Н. Технологии разработки мобильных приложений / Мытников А.Н. [и др.] // Теория и практика современной науки. – 2016. – № 4 (10). – С. 504–507.

ОСНОВНЫЕ АЛГОРИТМЫ СОРТИРОВКИ И ПОИСКА

Т.И. Селедков, ученик 9 класса,

Научный руководитель: Разумников С.В.^{1а}, к.т.н., доц.

¹Юргинский технологический институт (филиал)

Национального исследовательского Томского политехнического университета

652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26

²МБОУ «СОШ № 14 имени К.С. Федоровского»

652057, Кемеровская обл., г. Юрга, ул. Волгоградская, 3

E-mail: ^arazumnikov@tpu.ru

Аннотация: В статье рассматриваются основные алгоритмы сортировки и поиска.

Ключевые слова: алгоритм, поиск, сортировка.

Abstract: The article discusses the basic sorting and searching algorithms.

Keywords: algorithm, search, sorting.

Введение. Алгоритмы сортировки и поиска занимают центральное место в информатике и вычислительной технике. Они применяются практически во всех областях, начиная от баз данных и заканчивая обработкой больших массивов данных.

В этой статье мы рассмотрим основные алгоритмы сортировки и поиска, их принципы работы, сложность и практическое применение.