СОДЕРЖАНИЕ

[1. ВЫПОЛНЕНИЕ ПРОСТЕЙШИХ ИЗМЕРЕНИЙ С ПОМОЩЬЮ МУЛЬТИМЕТРА И ИНДИКАТОРНОЙ ОТВЕРТКИ. 3](#_uw7q10d6rhy2)

[2. СБОРКА ПК И ТЕСТИРОВАНИЕ ОСНОВНЫХ КОМПОНЕНТОВ. 5](#_fswk1rm4td6k)

[3. МИГРАЦИЯ ДИСКОВ ФОРМАТА MBR В ФОРМАТ GPT С СОХРАНЕНИЕМ ДАННЫХ И ВОССТАНОВЛЕНИЕ ЗАГРУЗКИ ОС. 7](#_r85c0r3t9zn9)

[4. МИГРАЦИЯ ОС НА НОВЫЙ ДИСК И РАЗБИЕНИЕ ДИСКОВ С ПОМОЩЬЮ ПРОГРАММЫ MINITOOL PARTITION WIZARD. 11](#_d65b9r39jwh0)

[5. ОСНОВЫ АДМИНИСТРИРОВАНИЯ WINDOWS SERVER. ЧАСТЬ 3. 14](#_gxyeiay13xa)

[ЗАКЛЮЧЕНИЕ 19](#_6bg451miak9z)

# ВВЕДЕНИЕ.

Введение к проекту приложения «Калькулятор» играет важную роль для определения цели работы, контекста разработки и значимости приложения в повседневной жизни. Калькуляторы являются одними из самых распространенных и необходимых инструментов, используемых как в образовательных учреждениях, так и в профессиональной сфере. С усложнением математических вычислений и ростом потребности в удобстве, мобильные приложения становятся незаменимыми помощниками для пользователей, нуждающихся в быстрой и точной обработке числовых данных.

В современном мире, где время и эффективность имеют первостепенное значение, наличие под рукой практичного калькулятора на смартфоне позволяет выполнять различные вычисления в любой момент. Основная цель нашего проекта заключается в разработке мобильного приложения, которое обеспечит пользователей интуитивно понятным и функциональным интерфейсом для выполнения базовых математических операций. Приложение будет включать традиционные функции, такие как сложение, вычитание, умножение и деление, а также, возможно, более сложные математические вычисления, что позволит расширить его функциональность и сделать его актуальным для более широкого круга пользователей.

Основные задачи, которые мы ставим перед собой в рамках данного проекта, включают анализ потребностей целевой аудитории, выбор подходящих технологий для разработки, создание удобного и привлекательного интерфейса, а также реализацию надежной и эффективной логики расчета. В ходе работы над проектом мы стремимся не только создать функциональное приложение, но и гарантировать его надежность и простоту в использовании.

Таким образом, разработка калькулятора не только обогатит мой опыт в области программирования и мобильной разработки, но и предоставит полезный инструмент, способствующий более удобному выполнению вычислений в повседневной жизни пользователей. В дальнейшем данное приложение может быть улучшено и дополнено новыми функциями, что сделает его еще более востребованным.

# 2. ПЛАНИРОВАНИЕ.

Планирование проекта разработки мобильного приложения калькулятора — это ключевой этап, который включает в себя несколько важных компонентов, обеспечивающих успешное завершение проекта. В этом плане мы будем рассматривать основные этапы разработки, сроки выполнения, распределение задач, а также приоритеты.

Первоначальным шагом в нашем планировании является определение цели и задач проекта. Цель заключается в создании функционального и удобного калькулятора для мобильных устройств, а задачи включают в себя реализацию базовых математических операций, а также добавление более сложных функций.

Следующим этапом является анализ целевой аудитории. Мы проведем исследование, чтобы определить, какие функции будут наиболее востребованными у пользователей. Это может включать в себя опросы, интервью и анализ существующих приложений.

После проведения анализа мы перейдём к выбору технологий для разработки приложения. Мы определим, будет ли приложение разработано для iOS, Android или обеих платформ, и выберем соответствующие языки программирования и фреймворки, такие как Swift для iOS или Kotlin для Android. Также важно будет рассмотреть, использовать ли кроссплатформенные решения, такие как Flutter или React Native, чтобы оптимизировать время разработки.

На следующем этапе мы будем работать над дизайном интерфейса пользователя (UI) и пользовательского опыта (UX). Это включает в себя создание рейтингов, прототипов и тестирование концепций, чтобы убедиться, что пользовательский интерфейс будет интуитивно понятным и эстетически привлекательным.

Затем начинается непосредственно процесс разработки. Этот этап можно разделить на несколько подвигов: реализация базовых функций (сложение, вычитание, умножение, деление), тестирование на наличие ошибок, реализация более продвинутых функций (например, работа с дробями, процентами и т.д.), а также интеграция дополнительных требований на основе анализа потребностей.

Параллельно с разработкой мы будем организовывать регулярные встречи команды для отслеживания прогресса, обсуждения проблем и корректировки планов по мере необходимости. Это позволяет всем участникам проекта быть в курсе событий и вносить предложения по улучшению.

Перед запуском приложения важно провести бета-тестирование, чтобы получить обратную связь от реальных пользователей. Это позволит выявить недоработки и улучшить качество конечного продукта.

Наконец, после завершения всех этапов мы подготовим приложение к публикации в магазинах приложений, создав маркетинговую стратегию для привлечения пользователей и повышения видимости продукта.

# 3. ПЛАНИРОВАНИЕ.

Разработка интерфейса пользователя (UI) является ключевым аспектом создания любого приложения, поскольку именно он определяет, как пользователи будут взаимодействовать с вашим продуктом. Этот процесс включает в себя несколько этапов, начиная с исследования и анализа потребностей пользователей и заканчивая созданием прототипов и визуальной частью интерфейса.

Первым шагом в разработке интерфейса является изучение целевой аудитории. Понимание кто ваши пользователи – это основа для создания интерфейса, который будет интуитивен и удобен в использовании. Вам следует провести опросы, фокус-группы или наблюдения за пользователями, чтобы собрать данные о том, какие функции им наиболее нужны и как они обычно взаимодействуют с аналогичными приложениями.

После того как вы соберете первичную информацию о ваших пользователях, следующим шагом будет создание пользовательских персонажей. Эти персонажи представляют собой обобщенные профили целевой аудитории и помогают направить ваш дизайн, учитывая различные потребности и сценарии использования. Персонажи позволят команде лучше понимать контекст, в котором будут использоваться приложения, а также поможет принимать более обоснованные решения в процессе разработки интерфейса.

Следующим этапом является создание пользовательских сценариев, которые описывают, как разные типы пользователей будут взаимодействовать с вашим приложением. Этот процесс включает в себя описание различных задач, которые пользователи должны выполнить, и шаг за шагом объясняет, как они зайдут в приложение, какие функции они будут использовать и каким образом будут достигать своих целей.

После того как у вас есть четкие пользовательские сценарии, можно переходить к созданию прототипа интерфейса. Прототип – это нечто среднее между эскизом и готовым интерфейсом; он позволяет визуализировать структуру приложения и основные элементы дизайна, такие как кнопки, формы и меню. В этом этапе можно использовать инструменты, такие как Figma, Sketch или Adobe XD для создания интерактивных прототипов. Интерактивные прототипы позволяют вашим пользователям тестировать интерфейс и предоставлять обратную связь на каждом этапе, что помогает выявить потенциальные проблемы или недочеты на раннем этапе.

После окончания тестирования прототипа и внесения всех необходимых исправлений, можно переходить к визуальному дизайну интерфейса. Это включает в себя выбор цветовой палитры, шрифтов и общей стилистики приложения. Важно, чтобы все элементы дизайна были согласованы и соответствовали общей концепции бренда. Пользовательский интерфейс должен быть не только красивым, но и функциональным, обеспечивая легкий доступ к информации и интуитивное взаимодействие с элементами управления.

На финальном этапе необходимо протестировать интерфейс, чтобы убедиться, что все элементы работают корректно на разных устройствах и разрешениях экранов. Также стоит обратить внимание на доступность приложения для пользователей с ограниченными возможностями. Это можно сделать путем использования подходящих контрастов, читаемых шрифтов и поддержания логической навигации.

# 4. РЕАЛИЗАЦИЯ ЛОГИКИ КАЛЬКУЛЯТОРА.

Реализация логики калькулятора — это ключевой аспект разработки программного обеспечения, который позволяет пользователям выполнять арифметические и логические операции. В данном случае мы поговорим о создании базового калькулятора, который способен выполнять простые операции: сложение, вычитание, умножение и деление. При реализации логики калькулятора стоит уделить внимание основным компонентам, таким как обработка ввода пользователя, выполнение расчетов и отображение результатов.

Первый шаг в реализации логики калькулятора — это создание интерфейса для ввода данных. Пользователь должен иметь возможность вводить числа и выбирать операции. Это можно сделать с помощью кнопок и текстовых полей. Интерфейс должен быть интуитивно понятным, чтобы пользователи могли легко вводить свои данные и получать результаты. Например, текстовое поле может быть использовано для отображения ввода пользователя, а кнопки — для выбора арифметических операций.

После того как интерфейс готов, следует переходить к обработке ввода. На этом этапе важно создать функцию, которая будет принимать пользовательские данные и определять, какие операции необходимо выполнить. Это можно сделать с помощью базового условного оператора или конструкции switch-case. Например, если пользователь нажимает кнопку «+», необходимо сохранить текущее число, ожидать следующий ввод и затем выполнить сложение.

Далее необходимо реализовать функции для выполнения арифметических операций. Например, можно создать отдельные функции для каждой операции: для сложения, вычитания, умножения и деления. Каждая из этих функций будет принимать два аргумента (числа) и возвращать результат операции. Важно также учитывать обработку ошибок: например, при делении нельзя делить на ноль, поэтому стоит предусмотреть соответствующую проверку и уведомление пользователя о неправильно введенных данных.

Когда операции реализованы, следует подумать о выводе результатов. После того как пользователь закончил вводить данные и выбрал операцию, калькулятор должен отобразить результат в текстовом поле или в отдельной области интерфейса. Это позволяет пользователю быстро увидеть результат своих действий без лишних усилий.

Необходимо также предусмотреть возможность сброса ввода. Кнопка «Сброс» или «Очистить» должна очищать все данные, которые были введены пользователем, и возвращать калькулятор в исходное состояние. Это важно для создания удобного и эффективного взаимодействия.

# 5. ТЕСТИРОВАНИЕ И ОТЛАДКА

Тестирование и отладка — это ключевые этапы в разработке программного обеспечения, которые обеспечивают его корректность, надежность и качество работы. Это особенно актуально в контексте разработки калькулятора, где точность результатов и интуитивно понятный интерфейс имеют первостепенное значение. Важным аспектом является систематический подход к этим процессам, который позволяет выявить и устранить возможные недочеты на ранних этапах.

Тестирование калькулятора может быть разделено на несколько видов, каждый из которых играет свою уникальную роль в обеспечении качества приложения. Первым этапом является модульное тестирование, которое предполагает проверку каждой отдельной функции калькулятора на корректность исполнения. На этом этапе разрабатываются тестовые случаи для основных арифметических операций — сложения, вычитания, умножения и деления. Тестирования должны учитывать не только стандартные входные данные, такие как целые и дробные числа, положительные и отрицательные значения, но также и краевые условия. Например, необходимо протестировать обработку деления на ноль и работу с очень большими или очень маленькими числами, чтобы убедиться, что программа не выдает ошибочных результатов или не зависает.

Следующим шагом является интеграционное тестирование. Этот этап направлен на проверку взаимодействия различных компонентов системы. В случае калькулятора это подразумевает тестирование взаимодействия пользовательского интерфейса с логикой вычислений. Важно убедиться, что данные, введенные пользователем, правильно обрабатываются и передаются между различными модулями приложения. Например, необходимо проверить, что после выполнения арифметической операции пользователь получает правильный результат, отображаемый в интерфейсе, и что интерфейс корректно реагирует на новые входные данные, обеспечивая бесшовный опыт использования.

Кроме того, полезно проводить функциональное тестирование, которое фокусируется на проверке всех функций калькулятора в соответствии с заранее установленными требованиями спецификации. Это включает в себя не только базовые операции, но и дополнительные функции, если они предусмотрены, такие как вычисление процентов, выполнение квадратных корней и использование памяти для хранения чисел. В этом контексте важно учитывать реальное поведение пользователя и разрабатывать сценарии использования, которые могут охватывать различные варианты ввода и последовательности операций.

Одним из важных аспектов тестирования является работа с пользовательским интерфейсом, что позволяет удостовериться, что все элементы, такие как кнопки, текстовые поля и меню, функционируют корректно и это соответствует ожиданиям конечного пользователя. Необходимо оценить как визуальность интерфейса, так и его функциональность. Проблемы на этапе пользовательского интерфейса могут включать в себя неправильное отображение элементов, неочевидное расположение кнопок или трудные для понимания всплывающие подсказки, которые могут негативно повлиять на общий опыт использования.

Отладка является неотъемлемой частью процесса разработки. Она подразумевает выявление и исправление ошибок, которые могут возникнуть во время тестирования. Для этого может быть полезным осуществление мониторинга и ведение логов, что позволит разработчикам гибко реагировать на обнаруженные проблемы. Логирование предоставляет важную информацию о том, где и почему произошли сбои, что упрощает процесс диагностики и последующего исправления ошибок.

Кроме того, привлечение независимых тестировщиков к процессу тестирования и отладки может значительно повысить качество конечного продукта. Свежий взгляд на приложение позволяет выделить проблемы, которые могли быть упущены разработчиками. Благодаря тщательному тестированию и отладке калькулятор может не только удовлетворить потребности пользователей, но и соответствовать высоким стандартам качества, что является особенно важным в условиях конкурентного рынка программного обеспечения.

# ЗАКЛЮЧЕНИЕ

В заключение отчета о моей учебной практике следует отметить, что выполненная работа по разработке калькулятора оказалась весьма продуктивной и насыщенной. В ходе практики я успешно освоил ключевые аспекты программирования, включая проектирование пользовательского интерфейса и реализацию основных арифметических операций, что является важным навыком в области разработки программного обеспечения.

Процесс разработки позволил мне на практике применить теоретические знания, полученные в учебном заведении. Я столкнулся с различными техническими вызовами, что способствовало углублению понимания принципов работы программ и важности тестирования. Особенно полезным был опыт по исправлению ошибок и оптимизации программы, что подчеркнуло необходимость тщательной проверки всех компонентов системы.

Кроме того, взаимодействие с командой разработчиков позволило мне развить навыки коммуникации и учиться работать в условиях совместной деятельности. Это взаимодействие было важным для понимания важности командной работы и распределения задач для достижения общего результата.

Обобщая, можно сказать, что опыт, который я получил в ходе своей учебной практики, является неоценимым. Он не только поспособствовал углублению профессиональных навыков, но и открыл новые горизонты для дальнейшего обучения и развития в сфере информационных технологий. Я уверен, что знания и практический опыт, полученные в ходе этой практики, будут полезны в моей будущей профессиональной деятельности.