Оглавление

[Глоссарий 2](#_Toc158680649)

[Введение 3](#_Toc158680650)

[Актуальность и проблематика 4](#_Toc158680651)

[Цель и задачи 5](#_Toc158680652)

[Постановка цели работы 5](#_Toc158680653)

[Задачи 5](#_Toc158680654)

[Исследовательская часть 6](#_Toc158680655)

[Анализ требований к приложению и интерфейсу 6](#_Toc158680656)

[Выбор языка программирования для создания приложения 6](#_Toc158680657)

[Архитектурная часть 7](#_Toc158680658)

[Разработка интерфейса и функционала приложения 7](#_Toc158680659)

[Тестирование и отладка 7](#_Toc158680660)

[Трудности разработки 7](#_Toc158680661)

[Заключение 8](#_Toc158680662)

[Список литературы 9](#_Toc158680663)

# Глоссарий

**ПО** – Программное обеспечение. Какая-либо выполняемая компьютером программа.

ОС – программное обеспечение, управляющее компьютерами и позволяющее запускать на них различное ПО.

**Интерфейс** – граница между двумя объектами. В общем смысле это инструмент для взаимодействия человека и программы. В области программирования интерфейс – это устройство или код, обеспечивающий совместимость элементов.

**Тестирование** – процесс исследования и испытания ПО, имеющий целью проверку соответствия между реальным и ожидаемым поведением программы.

**Баг** – жаргонное слово в программировании, означающее ошибку в программе.

**Кроссплатформенное** **приложение** – это приложение, способное работать на нескольких ОС.

**Документация** – документы, которые сопровождают некоторое ПО и описывают то, как работает программа и/или то, как её использовать.

# Введение

Приложение для заметок очень полезно в повседневной работе или организации информации. Оно может использоваться как электронный блокнот, где пользователи могут сохранять свои мысли, задачи или идеи без необходимости использования бумаги. В зависимости от пользователя и его умения пользоваться подобными приложениями использование приложения для заметок может положительно сказаться на общей продуктивности пользователя. Так же в данном случае уменьшается используемая бумага, что может положительно сказаться на экологии и денежных затратах на бумажные носители.

## Актуальность и проблематика

Современные приложения для заметок могут столкнуться с различными проблемами. Одна из таких проблем может быть связана с нежелательным сбором пользовательских данных без достаточного осведомления пользователей об этом процессе. Некоторые приложения могут неявно запрашивать доступ к персональной информации или даже передавать ее третьим лицам без разрешения. Кроме того, некоторые приложения для заметок могут содержать функционал или сервисы, которые не всегда нужны конкретному пользователю. Это может усложнять использование самого приложения, создавать путаницу в интерфейсе пользователя или даже мешать пользоваться приложением, в случае всплывающих окон с просьбой приобрести премиум-подписку на сервис[[1]](#_Список_литературы_2).

# Цель и задачи

## Постановка цели работы

Целью данной работы является разработка приложения для заметок. Приложение должно выглядеть современно и иметь базовый функционал.

## Задачи

Для достижения цели были выявлены следующие задачи:

* Анализ требований к приложению и интерфейсу
* Выбор языка программирования для создания приложения
* Разработка интерфейса и функционала приложения
* Тестирование и отладка приложения

# Исследовательская часть

## Анализ требований к приложению и интерфейсу

Приложение должно выглядеть современно, не иметь отталкивающий интерфейс и цветовую палитру. Использование приложения должно быть интуитивным. Также оно должно работать стабильно и плавно. Не иметь багов и проблем, как визуальных, так и негативно влияющих на опыт использования.

## Выбор языка программирования для создания приложения

Для приложения было принято использовать язык Swift[[2]](#_Список_литературы_2). С его помощью можно сделать приложение как кроссплатформенное приложение iOS, iPadOS, macOS, tvOS, visionOS и watchOS, так и приложение только для некоторых из них. Они все по своей структуре используют схожий принцип и инструменты, что упрощает разработку, перенос и адаптацию интерфейса для каждой операционной системы из вышеперечисленных.

Из достоинств Swift также можно отметить то, что при удачном применении технологий и дизайна оно позволит использовать дизайн Apple – одной из крупнейших компании мира[[3]](#_Список_литературы_2).

# Архитектурная часть

## Разработка интерфейса и функционала приложения

Процесс разработки интерфейса используя Swift очень схож с разработкой интерфейса, используя Dart и Flatter. Система из “view” (рус. “видов”) позволяет собрать интерфейс приложения, словно сложный конструктор из базовых деталей.

Разработку функционала приложения было решено осуществить по Чистой Архитектуре[[4]](#_Список_литературы_2) (англ. “Clean Architecture”), что для начала осложнило процесс разработки, но упростило его в процессе добавления новых функций и отладки кода. В случае необходимости изменения кода по любой причине это можно будет осуществить проще, чем без использования Чистой Архитектуры.

## Тестирование и отладка

В процессе разработки тестирование и отладка являются важнейшей частью перед выпуском приложения. Тестирование приложения в процессе разработки позволяет улучшить понимание языка и держать код в чистоте.

## Трудности разработки

В процессе разработки приложения, как и функционала, возникла трудность – несмотря на обширную и доступную документацию Apple для разработки приложений используя Swift, вся документация обновляется, как и сам Swift. Но ограниченная железная часть не позволяет использовать новейшую версию Swift, что существенно ограничивает использование новых, и усложняет применение старых инструментов в случае их обновления в новой версии языка и необратимому изменению их документации.

Последствиями данной трудности стало серьезное ограничение приложения. В финальной версии функционал приложения был сильно ограничен, несмотря на то что такой же функционал доступен для других языков разработки, например Flutter.

# Заключение

Таким образом в работе проведен анализ требований к приложению, выбран наиболее удачный язык программирования, разработан интерфейс и функционал, а также проведено тестирование приложения для заметок.

Все задачи цели были достигнуты. Были выполнены все этапы перед реализацией приложения. Приложение и его исходный код находятся в открытом доступе на платформе GitHub[[5]](#_Список_литературы_2).

# Список литературы

1. Премиум-заметки // Notion – URL: <https://www.notion.so/pricing>
2. Swift // Apple Developer – URL: <https://developer.apple.com/swift/>
3. Компания Apple // Wikipedia – URL: <https://ru.wikipedia.org/wiki/Apple>
4. Чистая Архитектура // Blog – URL: <https://blog.cleancoder.com/uncle-bob/2011/11/22/Clean-Architecture.html>
5. Приложение для заметок // GitHub – URL: <https://github.com/KanaDvfu/MeList-PP2->