

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и
информатики»

Кафедра телекоммуникационных систем и вычислительных средств
(ТС и ВС)

Расчетно-графическая работа
по дисциплине
«Визуальное программирование»

по теме:
MP3 PLAYER

Студент:
Группа ИА-332

В.П. Шепталин

Преподаватель:

Р.В. Ахнашев

Новосибирск 2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	4
1.1 Формат МР3 и его особенности.....	4
1.2 Основные компоненты МР3 плеера на Kotlin	4
1.3 Архитектура приложения.....	4
1.4 Обработка разрешений.....	5
1.5 Обработка событий и состояния приложения	5
2 ПРАКТИЧЕСКАЯ ЧАСТЬ	6
2.1 Создание интерфейса пользователя	6
2.2 Загрузка и обработка музыкальных файлов	6
2.3 Инициализация и воспроизведение аудио	7
2.4 Реализация управления воспроизведением.....	7
2.5 Обработка разрешений.....	8
2.6 Тестирование и отладка.....	9
ЗАКЛЮЧЕНИЕ	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	11

ВВЕДЕНИЕ

В современном мире мобильные устройства и программное обеспечение для воспроизведения мультимедиа играют важную роль в повседневной жизни. MP3-плееры — это одно из самых популярных средств для прослушивания аудиофайлов различных форматов. Цель данной работы — разработка простого и функционального MP3-плеера на языке программирования Kotlin, который широко используется для создания приложений под платформу Android.

В ходе выполнения работы рассматриваются основные принципы воспроизведения аудио, управление воспроизведением (воспроизведение, пауза, остановка, переход между треками), а также организация пользовательского интерфейса с помощью современных средств разработки Android-приложений. Особое внимание уделяется интеграции с системными ресурсами устройства и обеспечению удобства использования.

Данный проект позволяет закрепить практические навыки программирования на Kotlin, познакомиться с особенностями работы с аудиоданными и интерфейсом Android, а также получить опыт реализации мультимедийных приложений, востребованных в современных технологиях.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Формат MP3 и его особенности

MP3 — это формат аудиосжатия с потерями, который позволяет уменьшить размер звуковых файлов, сохраняя приемлемое качество воспроизведения. Сжатие осуществляется за счёт удаления избыточных данных, которые человеческое ухо практически не воспринимает.

Для воспроизведения MP3 файлов требуется декодер, который преобразует сжатый поток данных в PCM (Pulse Code Modulation) формат, пригодный для воспроизведения через динамики.

1.2 Основные компоненты MP3 плеера на Kotlin

MP3 плеер реализован на языке Kotlin и включает следующие основные компоненты:

- **MediaPlayer** — встроенный класс Android SDK для воспроизведения аудио.
- **AudioManager** — для управления громкостью и аудиопотоками.
- **Handler** — для обновления позиции воспроизведения и синхронизации с SeekBar.
- **ContentResolver** — для доступа к музыкальным файлам, хранящимся на устройстве.

1.3 Архитектура приложения

Приложение представляет собой одноактивити с XML-интерфейсом и функциональными кнопками управления:

- **Play/Pause** — старт и пауза воспроизведения.
- **Next/Back** — переключение между треками.
- **Cycle** — включение и отключение циклического воспроизведения.
- **Volume Up/Down** — управление уровнем громкости через AudioManager.

Для организации воспроизведения MP3 файлов используется локальный список песен, формируемый из директории /Music и медиа-хранилища устройства.

1.4 Обработка разрешений

Для доступа к музыкальным файлам приложение запрашивает разрешение на чтение данных. В зависимости от версии Android это может быть READ_EXTERNAL_STORAGE или READ_MEDIA_AUDIO.

Система реализована через `ActivityResultContracts.RequestPermission()`, что позволяет гибко управлять процессом запроса разрешений.

1.5 Обработка событий и состояния приложения

В приложении реализована система управления состояниями:

- **Инициализация и подготовка** — загрузка треков и настройка MediaPlayer.
- **Воспроизведение и пауза** — обновление кнопки Play/Pause в зависимости от состояния MediaPlayer.
- **Переключение треков** — корректное обновление индекса трека и перезапуск воспроизведения.
- **Завершение воспроизведения** — автоматический переход к следующему треку или возврат к первому треку в списке.

Таким образом, теоретическая часть охватывает основные компоненты и архитектуру MP3 плеера, реализованного на Kotlin с использованием Android SDK.

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Создание интерфейса пользователя

В приложении реализован простой и интуитивно понятный интерфейс, состоящий из следующих компонентов:

- **TextView** для отображения текущего трека.
- **SeekBar** для визуального отображения и управления прогрессом воспроизведения.
- **Button** для управления воспроизведением: Play/Pause, Next, Back, Cycle.
- **Button** для управления громкостью: Volume Up, Volume Down.

Все компоненты интерфейса размещены в XML-файле `activity_mp3.xml`. Вёрстка выполнена с использованием `ConstraintLayout` для гибкости и адаптивности интерфейса.

2.2 Загрузка и обработка музыкальных файлов

В проекте реализовано два метода для загрузки музыкальных файлов:

1. **Из хранилища устройства:** Используется `ContentResolver` для сканирования медиафайлов в директории `/Music`. Файлы фильтруются по расширению `.mp3`, после чего их пути сохраняются в списке `songs`.
2. **Из директории `/Music`:** Реализована проверка на существование директории `/Music` и считывание файлов напрямую через `File API`.

Пример загрузки файла из медиа-хранилища:

```
1 private fun loadMusicFromMediaStore() {  
2     val uri: Uri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI  
3     val projection = arrayOf(  
4         MediaStore.Audio.Media.DATA,  
5         MediaStore.Audio.Media.TITLE  
6     )
```

```

7      val selection = MediaStore.Audio.Media.IS_MUSIC + "!= 0"
8
9      contentResolver.query(uri, projection, selection, null,
10         null)?.use { cursor ->
11         while (cursor.moveToNext()) {
12             val path = cursor.getString(0)
13             val title = cursor.getString(1)
14             songs.add(path)
15             songTitles.add(title)
16         }
17     }

```

Листинг 2.1 — Загрузка музыки из MediaStore

2.3 Инициализация и воспроизведение аудио

Для воспроизведения MP3 используется класс MediaPlayer. В процессе инициализации создаётся новый экземпляр MediaPlayer и устанавливается путь к текущему треку:

```

1 private fun initializeMediaPlayer() {
2     if (songs.isNotEmpty()) {
3         mediaPlayer = MediaPlayer()
4         mediaPlayer.setDataSource(songs[currentSongIndex])
5         mediaPlayer.prepare()
6         updateSongTitle()
7         setupSeekBar()
8         setupButtons()
9     }
10 }

```

Листинг 2.2 — Инициализация MediaPlayer

2.4 Реализация управления воспроизведением

Основные функции управления воспроизведением:

- **Play/Pause:** Переключает состояние воспроизведения и обновляет текст кнопки.

- **Next/Back:** Изменяет индекс текущего трека и перезапускает воспроизведение.
- **Cycle:** Включает/выключает режим циклического воспроизведения.
- **SeekBar:** Отслеживает текущую позицию трека и позволяет перематывать воспроизведение.

Пример реализации кнопки Play/Pause:

```

1 play.setOnClickListener {
2     if (!mediaPlayer.isPlaying) {
3         mediaPlayer.start()
4         play.text = "Pause"
5     } else {
6         mediaPlayer.pause()
7         play.text = "Play"
8     }
9 }

```

Листинг 2.3 — Обработка нажатия кнопки Play/Pause

2.5 Обработка разрешений

Для доступа к музыкальным файлам приложение запрашивает разрешение на чтение внешнего хранилища:

```

1 private fun checkPermission() {
2     val permission = if (Build.VERSION.SDK_INT >=
3         Build.VERSION_CODES.TIRAMISU) {
4         Manifest.permission.READ_MEDIA_AUDIO
5     } else {
6         Manifest.permission.READ_EXTERNAL_STORAGE
7     }
8     when {
9         ContextCompat.checkSelfPermission(this, permission) ==
10             PackageManager.PERMISSION_GRANTED -> {
11             loadMusic()
12         }
13         else -> {
14             requestPermissionLauncher.launch(permission)
15         }
16     }
17 }

```


Листинг 2.4 — Проверка разрешений

2.6 Тестирование и отладка

В ходе тестирования было проверено:

- Корректность загрузки музыкальных файлов из /Music.
- Корректная работа кнопок управления воспроизведением.
- Отображение текущего трека и его позиции на SeekBar.
- Обработка ошибок при отсутствии файлов или отсутствии разрешений.

Таким образом, приложение реализует все основные функции MP3 плеера, включая воспроизведение, переключение треков, регулировку громкости и цикличное воспроизведение.

ЗАКЛЮЧЕНИЕ

В рамках данной расчетно-графической работы было разработано мобильное приложение MP3-плеер на языке программирования Kotlin с использованием фреймворка Android. Основной функционал приложения включает в себя воспроизведение музыкальных файлов из хранилища устройства, управление треками (воспроизведение, пауза, переход к следующему и предыдущему треку), регулировку громкости и активацию режима циклического воспроизведения.

Приложение успешно реализует задачи, поставленные в техническом задании: 1. Загрузка музыкальных файлов из директории /Music и медиахранилища устройства. 2. Управление воспроизведением с помощью кнопок Play/Pause, Next и Back. 3. Отображение информации о текущем треке и его состоянии на SeekBar. 4. Реализация простого и интуитивно понятного интерфейса на основе ConstraintLayout. 5. Обработка системных разрешений для доступа к аудиофайлам.

В ходе выполнения работы изучены основные компоненты Android SDK для работы с мультимедиа, такие как MediaPlayer и AudioManager. Также были применены базовые подходы к организации пользовательского интерфейса и управления разрешениями.

Основным направлением для дальнейшего улучшения приложения может стать реализация плейлиста с возможностью выбора треков пользователем, создание уведомлений о текущем треке, а также расширение форматов воспроизводимых аудиофайлов.

Таким образом, выполненная работа позволила закрепить знания по разработке мобильных приложений на Kotlin и продемонстрировала базовые навыки создания мультимедийных приложений на платформе Android.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Android Developers*. Media Playback Using MediaPlayer // Android Developer Documentation. — 2025. — URL: <https://developer.android.com/guide/topics/media/mediaplayer>.
2. *Kotlin Documentation*. Android Development with Kotlin // Kotlin Official Site. — 2025. — URL: <https://kotlinlang.org/docs/android-overview.html>.
3. *Android Developers*. AudioManager API // Android Developer Documentation. — 2025. — URL: <https://developer.android.com/reference/android/media/AudioManager>.
4. *Android Developers*. Handling Permissions in Android // Android Developer Documentation. — 2025. — URL: <https://developer.android.com/training/permissions/requesting>.
5. Григорьев С.В. Основы верстки документов в системе ЛАТ_EX. — Инфра-М, 2023.
6. *Material Design*. Material Design Guidelines // Material Design Documentation. — 2025. — URL: <https://material.io/design>.
7. Построение графиков в LaTeX/PGFPlots // Habr. — 2015. — URL: <https://habr.com/ru/articles/250997/>.