**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ   
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Доцент департамента программной инженерии факультета компьютерных наук, кандидат техн. Наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Д. Брейман  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г. |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. |  | | **ПРИЛОЖЕНИЕ МУЗЫКАЛЬНЫЙ ПЛЕЕР ДЛЯ ANDROID  Пояснительная записка   ЛИСТ УТВЕРЖДЕНИЯ** **RU.17701729.04.01**-**01** **81** **01-1-ЛУ**  Исполнитель  Студент группы БПИ172  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ О.В.Омельчишина /  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г. | |  |

**УТВЕРЖДЕН**

**RU.17701729.04.01**-**01** **81** **01-1 ЛУ**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. |  | | **ПРИЛОЖЕНИЕ МУЗЫКАЛЬНЫЙ ПЛЕЕР ДЛЯ ANDROID  Пояснительная записка RU.17701729.04.01**-**01** **81** **01-1  Листов 22** |  |

**СОДЕРЖАНИЕ**

[1. ВВЕДЕНИЕ 4](#_Toc8735492)

[1.1. Наименование программы 4](#_Toc8735493)

[1.2. Документы на основании которых ведётся разработка 4](#_Toc8735494)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 5](#_Toc8735495)

[2.1. Функциональное назначение 5](#_Toc8735496)

[2.2. Эксплуатационное назначение 5](#_Toc8735498)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_Toc8735499)

[3.1. Постановка задачи на разработку программы 6](#_Toc8735500)

[3.2. Описание функционирования программы 7](#_Toc8735501)

[3.2.2 Операция (Activity) MainActivity 7](#_Toc8735502)

[3.2.3 Сервис MediaPlayerService 7](#_Toc8735503)

[3.2.4 Интерфейс onItemClickListener 7](#_Toc8735504)

[3.2.6 Класс RecyclerView\_Adapter 7](#_Toc8735505)

[3.2.7 Класс SettingsBottomSheet 7](#_Toc8735506)

[3.3 Алгоритм отображения основного UI 8](#_Toc8735507)

[3.4 Описание и обоснование метода организации входных и выходных данных 8](#_Toc8735508)

[3.5 Описание и обоснование выбора технических и программных средств. 9](#_Toc8735509)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 10](#_Toc8735510)

[4.1. Ориентировочная экономическая эффективность 10](#_Toc8735511)

[4.2. Предполагаемая потребность 10](#_Toc8735513)

[4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 10](#_Toc8735515)

[5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 11](#_Toc8735516)

[Приложение 1 12](#_Toc8735517)

[Приложение 2 13](#_Toc8735518)

[Приложение 3 16](#_Toc8735519)

[Приложение 4 17](#_Toc8735520)

# 1. ВВЕДЕНИЕ

## 1.1. Наименование программы

Наименование программы: «Музыкальный плеер на Android» («Music Player for Android»).

## 1.2. Документы на основании которых ведётся разработка

Приложение к приказу Национального исследовательского университета "Высшая школа экономики" № 2.3-02/1012-0 2 от 10.12.18

# 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## 2.1. Функциональное назначение

## Функциональным назначением программы является воспроизведение музыкальных файлов на устройствах с операционной системой Android.

## 2.2. Эксплуатационное назначение

Программа может использоваться для воспроизведения и отображения музыкальных файлов, содержащихся на устройстве, а также поиск среди найденных файлов.

# 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

# 3.1. Постановка задачи на разработку программы

Разрабатываемая программа должна обеспечивать пользователя нижеследующими возможностями:

* Поиск музыкальных файлов на устройстве Android.
* Отображение содержимого файлов с поддержкой тегов ID3v1, ID3v2: Title (наименование музыкального произведения (далее м.п.)), Artist (исполнитель м.п.), Album (наименование альбома м.п.), Album\_id (содержит информацию о нахождении обложки альбома на устройстве); всех иных музыкального файлов: наименование файла, дата добавления на устройство.
* Воспроизведение музыкальных файлов.
* Возможность выбора порядка воспроизведения: линейный порядок, случайный порядок, повторение одной композиции.
* Воспроизведение списка композиций, находящийся на экране в момент нажатия по элементу списка.
* Возможность сортировки музыкальных файлов в интерфейсе приложения: по названию файла, наименованию м.п., исполнителю м.п., альбома м.п., дате добавления.
* Возможность поиска музыкальных произведений по наименованию файла, наименованию трека, исполнителю, альбому.
* Настройка поиска: выбор тегов (содержимого файлов), среди которых осуществлять поиск.
* Список избранного, добавление треков в этот список.
* Сохранение списка избранного после выхода из программы
* Сохранение настроек сортировки списка м.п. и последнего воспроизводимого файла после выхода из программы.
* Прослушивание состояния телефона – необходимо останавливать воспроизведение если другая программа начала воспроизведение или произошел входящий/исходящий вызов.

# 3.2. Описание функционирования программы

Программа включает в себя классы: CustomTouchListener, MainActivity, MediaPlayerService, FirstRun, RecyclerView\_Adapter, SettingsBottomSheet, StorageUtil, Track и интерфейс onItemCliclListener

3.2.1 Класс CustomTouchListener

Вспомогательный класс. Перехватчик и обработка нажатий

### 3.2.2 Операция (Activity) MainActivity

Основной пользовательский интерфейс программы. При запуске подключает (связывает) MediaPlayerService и устанавливает с ним связь.

### 3.2.3 Сервис MediaPlayerService

Содержит в себе техническую реализацию плеера, а также обработку реакций на широковещательные сообщения.

* Получение информации о состоянии телефона происходит с помощью BroadcastReceiver.
* Запуск воспроизведения происходит с помощью отправки намерений (Intent) через sendBroadcast и получение этих намерений через BroadcastReceiver
* Случайное воспроизведение заполняется 1 раз (при запросе из MainActivity) и что позволяет возвращаться на предыдущий/следующий воспроизводимый файл после его окончания с помощью кнопки играть\_предыдущее/играть\_следующее

### 3.2.4 Интерфейс onItemClickListener

Вспомогательный интерфейс. Используется в CustomTouchListener

### 3.2.6 Класс RecyclerView\_Adapter

Составляющая обновляемых списков файлов. Содержит реализацию поиска по списку, а также сортировку.

* Поиск настраивается через общие поля этого класса и описан в filterOptions
* Сортировка выполнена с помощью настраиваемых сортировок языка Java

### 3.2.7 Класс SettingsBottomSheet

Реализация UI окна настроек.

# 3.3 Алгоритм отображения основного UI

3.3.1 Запрос у пользователя необходимых для работы приложения на Anroid API 23+ разрешений на работу с внутренним хранилищем и возможность принимать широковещательные сообщения.

3.3.2 Поиск всех музыкальных файлов на устройстве, сохранение их в отдельный список

3.3.3 Сохранение списка в файл настроек

3.3.4 При наличии сохранённых настроек и списка избранного в файле настроек происходит их погрузка в главный Activity

3.3.5. С учетом наличия старых сохранений запускается пользовательский интерфейс

3.3.6 Ожидание от пользователя запроса на воспроизведение или других действий: поиск, смена настроек

# 3.4 Описание и обоснование метода организации входных и выходных данных

На вход поступают файлы во внутреннем хранилище Android устройства

Выходными данными являтся воспроизведение музыкальных файлов.

MediaStore.Audio.Media используется для изъятия из музыкальных файлов содержимое тегов (при их наличии). Для хранения изображения альбомов (обложек) используется Album\_id, который представляет собой информацию, необходимую для генерации скрытого URI к изображению обложки. Данный способ хранения выбран по причине более быстрой обработки файлов чем декодирование обложки из каждого файла и хранение в Bitmap каждого объекта.

# 3.5 Описание и обоснование выбора технических и программных средств.

Программа Music Player дает доступ к автоматическому воспроизведению музыкальных файлов на устройстве Android

Для работы программы требуется следующий состав программных и технических средств:

* Android устройство или эмулятор устройства, оснащённый двух- и более ядерным процессором с тактовой частотой 2 ГГц или более.
* Сенсорный экран размером 5 и более дюймов
* Рекомендуется 0.5 ГБ ОЗУ или более. Минимальный объём – 300 МБ;
* Не менее 20 МБ свободного места на жёстком диске или носителе, на котором хранится программа;
* Наличие музыкальных файлов во внутреннем хранилище.
* Предоставленные разрешение на прослушивание широковещательных сообщений и чтение/запись внутреннего хранилища.
* Поддерживаемая операционная система:
* Android 9 3.18.120-perf
* MIUI Goval 10.3.10(ODEMIXM) ~ Android 8.0.0
* EMUI 8.0.0 ~ Android 8.0.0

Потенциально поддерживаемые системы:

* Android 7.0 и выше

Необходимое тестирование

# 4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## 4.1. Ориентировочная экономическая эффективность

## В рамках данной работы расчёт экономической эффективности не предусмотрен. Монетизация разработки не предусматривается.

## 4.2. Предполагаемая потребность

## Данный продукт может быть использован развлекательных целях.

## 4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Полное отсутствие рекламы или иных скрытых функционалов таких как слежение и собирание информации о пользователе, простой интерфейс.

# 5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.301-79 Программа и методика испытаний. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.401-78 Текст программы. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.505-79 Руководство оператора. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.404-79 Пояснительная записка. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. Внутри MP3. [Электронный ресурс]//URL: <https://habr.com/ru/post/103635/>
9. Android и звук: как делать правильно [Электронный ресурс]//URL: <https://habr.com/ru/post/339416/>
10. Guide to Building an Android Audio Player App [Электронный ресурс]//URL: <https://www.sitepoint.com/a-step-by-step-guide-to-building-an-android-audio-player-app/>
11. [www.freepik.com](http://www.freepik.com)
12. Создание музыкального проигрывателя на Android [Электронный ресурс]//URL: <https://code.tutsplus.com/ru/tutorials/create-a-music-player-on-android-project-setup--mobile-22764>
13. Компоненты программного обеспечения Android [Электронный ресурс]//URL: <https://novainfo.ru/article/2143>
14. Уроки по Android Разработке [Электронный ресурс] //URL:http://developer.alexanderklimov.ru/android/

# Приложение 1

**Термины и определения**

Далее указаны необходимые понятия:

1. Теги - ID3 (от англ. Identify a MP3) — формат метаданных, наиболее часто используемый в звуковых файлах в формате MP3. ID3 подпись содержит данные о названии трека, альбома, имени исполнителя и т. д., которые используются мультимедиапроигрывателями и другими программами, а также аппаратными проигрывателями, для отображения информации о файле и автоматического упорядочивания аудиоколлекции.[8]
2. Широковещательные сообщения – Broadcast receivers/ Широковещательный приемник — этот компонент отвечает за распространение общесистемных сообщений, отслеживание и реагирование на действия. Многие оповещения идут от системы, например, сообщения о том, что заряд батареи мал или экран выключен.[13]
3. UI – пользовательский интерфейс

# Приложение 2

**Диаграмма классов**

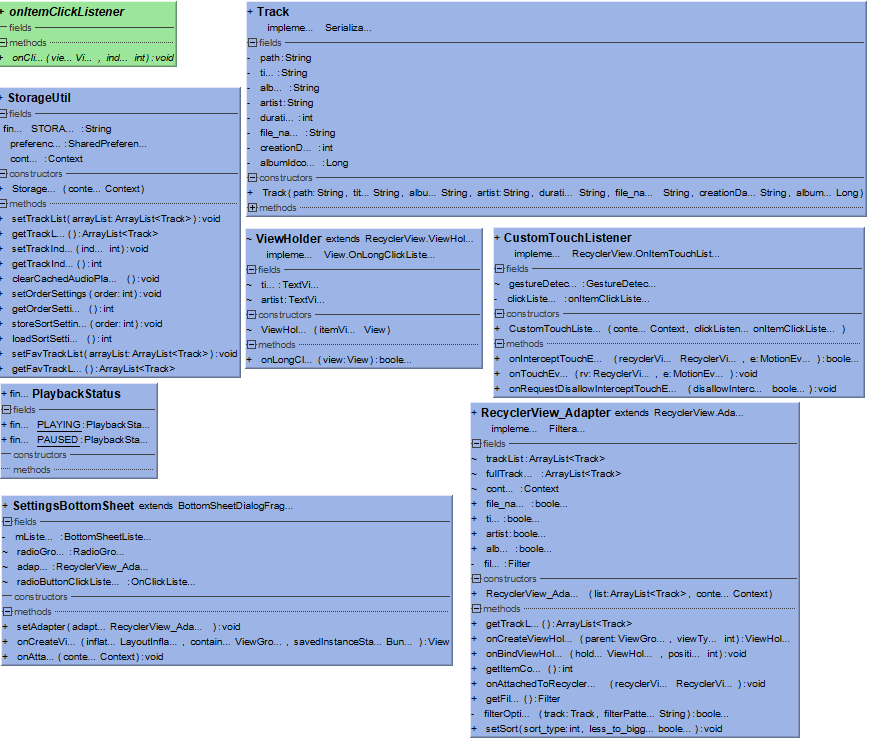


Рис. 1 Диаграмма классов ч.1

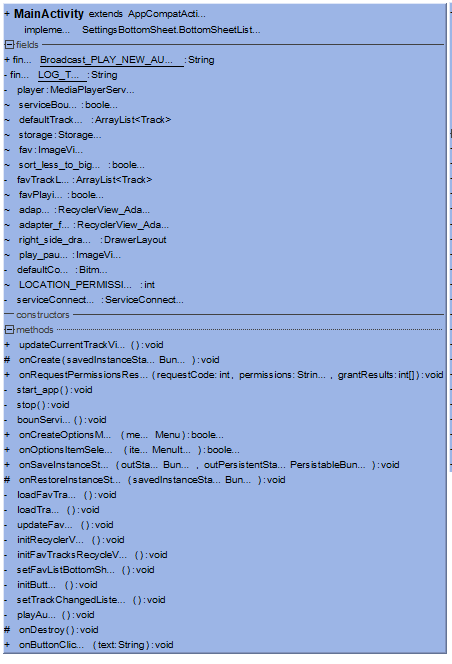


Рис. 2 Диаграмма классов 2

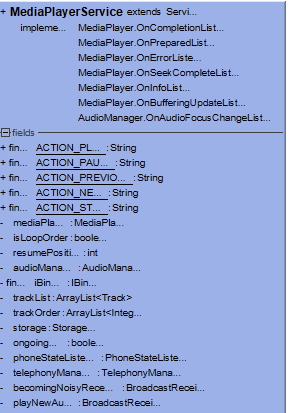
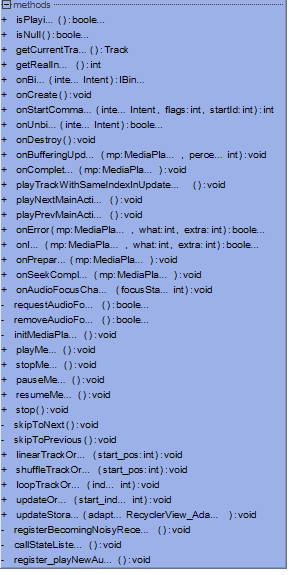
 

Рисунок 3. Диграмма классов 3

# Приложение 3

**Описание и функциональное назначение классов**

|  |  |
| --- | --- |
| Класс | Назначение |
| MainActivity | Основной пользовательский интерфейс программы. При запуске подключает MediaPlayerService и устанавливает с ним связь. |
| MediaPlayerService | Содержит в себе техническую реализацию плеера а так же обработку на широковещательные сообщения. |
| CustomTouchListener | Вспомогательный класс. Перехватчик и обработка нажатий |
| RecyclerView | Составляющая обновляемых списков файлов. Содержит реализацию поиска по списку. |
| SettingsBottomSheet | Реализация UI настроек. |

# Приложение 4

**Описание и функциональное назначение полей и методов классов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс** MainActivity | | | | | |
| **Поля** | | | | | |
| Имя | | Модификатор доступа | Тип | Аргументы | Описание |
| MediaPlayerService | | private | MediaPlayerService | – | Элемент Service связанный с текущим Activity |
| defaultTrackList | | ArrayList<Track> | Список музыкальных файлов на устройстве, заполняемый при запуске |
| Adapter/ adapter\_fav | | RecyclerView\_Adapter | Адаптер для основного списка и списка избранных файлов |
| defaultCover | | Bitmap | Изображение, отображаемое в случае отсутствия обложки у файла |
| Методы | | | | | |
| Имя | Модификатор доступа | | Тип | Аргументы | Описание |
| updateCurrentTrackView | Public | | void | – | Обновляет содержимое отображаемое в UI «текущий файл» |
| setTrackChangedListener | private | | Устанавливает Listener на изменение текущего воспроизводимого файла в связанном MediaPlayerService |
| initButtons | Устанавливает Listener на кнопки, отображаемые в UI а так же подгружает UI в соответствии с сохранённым состоянием |
| playAudio | С помощью BroadCast передает сообщение о необходимости воспроизвести следующий файл в связанный MediaPlayerService |
| updateFavList | Обновляет содержимое списка избранного в файле настроек |
| initRecyclerView/ initFavTracksRecycleView | Инициализация отображаемых списков файлов, установка Listener на нажатия по элементам списка |
| loadTracks | Поиск всех музыкальных файлов на устройстве и заполнение defaultTrackList |
| loadFavTracks | Подгрузка списка избранных файлов из файла настроек |
| onRequestPermissionsResult | public | | void |  | Запрашивает у пользователя необходимые разрешения для работы приложения |
| start\_app | Private | |  | Вызов методов, подгружающих файлы, UI |
| bounService | Private | |  | Устанавливает взаимную связь с сервисом воспроизведения |
| Stop | Private | |  | Выход из программы |
| onCreateOptionsMenu | Public | | Menu | Установка функционала Menu в UI |
| onSaveInstanceState/ onRestoreInstanceState | Private | | Bundle | Обработка и сохранение текущего состояния программы |
| onResume | Private | |  | Восстановление UI после закрытия приложения (установки на паузу) |
| setFavListBottomSheet | Private | |  | Установка UI выдвигаемого списка избранного |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Класс MediaPlayerService** | | | | | | | |
| **Поля** | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Описание | |
| ACTION\_\* | | Public | | String | | Содержит информацию о намерениях воспроизведения – пауза, стоп, воспроизвести, следующий, предыдущий | |
| mediaPlayer | | Private | | MediaPlayer | | Контролирует воспроизведение файлов | |
| resumePosition | | Private | | Integer | | Содержит информацию о времени, когда воспроизведение было поставлено на паузу | |
| ongoingCall | | Private | | Boolean | | Флаг что в текущий момент идет звонок | |
| audioManager | | Private | | AudioManager | | Управление аудио-фокусом | |
| new\_trackListener | | Public | | Интерфейс | | Отслеживает изменение воспроизводимого файла | |
| TrackState | | Public | | Класс | | Предоставляет информацию о произощедших изменениях – поменялся воспроизводимый файл | |
| Storage | | Private | | StorageUtil | | Предоставляет доступ для взаимодействия с файлом настроек | |
| LocalBinder | | Public | | Класс | | Возвращает себя, чтобы можно было вызывать public методы извне | |
| trackOrder | | Private | | ArrayList<Integer> | | Содержит в себе информацию о порядке воспроизведения объектов в trackList | |
| trackList | | ArrayList<Track> | | Содержит информацию о текущем воспроизводимом наборе файлов | |
| becomingNoisyReceiver | | BroadcastReceiver | | При изменении способы вывода звука ставит воспроизведение на паузу | |
| playNewAudio | | BroadcastReceiver | | При получении broadcast сообщения воспроизводит текущий файл | |
| **Методы** | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Описание |
| getCurrentTrack | public | | Track | |  | | Возвращает текущий воспроизводимый файл |
| getRealIndex | int | |  | | Возвращает индекс текущего воспроизводимого файл |
| onCreate | void | |  | | Подготавливает список файлов и прочую информацию для воспроизведения файлов с помощью MediaPlayer а так же запускает слушателей Broadcast сообщений |
| onStartCommand | Int | | Intent, int, int | | Запускается, когда происходит запрос от связанного с сервисом Activity. |
| onDestroy | Void | |  | | Останавливает всех слушателей и останавливает воспроизведение |
| onCompletion | Void | | MediaPlayer | | Вызывается когда воспроизведение файла завершилось |
| playTrackWithSameIndexInUpdatedList | Void | |  | | Вызывается при изменении размера списка воспроизведения |
| playNextMainActivity/ playPrevMainActivity | Void | |  | | Вызывается из UI если был запрос на новое воспроизведение |
| onAudioFocusChange | Void | |  | | Обработчик изменений аудиофокуса |
| requestAudioFocus | Boolean | |  | | Запрос аудиофокуса у системы |
| removeAudioFocus | Boolean | |  | | Освобождение аудифокуса |
| initMediaPlayer | Void | |  | | Подготовка файла к проигрыванию |
| Play/stop/pause/resume - media | Void | |  | | Установление состояния поля mediaPlayer |
| skipToNext/skipToPrevious | Void | |  | | Установка индексов следующего воспроизводимого файла |
| linearTrackOrder/ shuffleTrackOrder/ loopTrackOrder | void | | Int | | Устанавливает порядок воспроизведения trackList начиная с int и сохраняет его в trackOrder |
| updateOrder/ updateStorage | Void | | Int/ RecyclerView\_Adapter | | Обновляет содержимое trackList и trackOrder в зависимости от текущего состояния в файле настроек |
| callStateListener | void | |  | | Обработка события входящего вызова и его прекращения |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Класс StorageUtil** | | | | | | |
| **Поля** | | | | | | |
| Имя | Модификатор доступа | Тип | | Описание | | |
| STORAGE | private | String | | Имя файла настроек | | |
| preferences | SharedPreferences | | Экземпляр класса общих настроек | | |
| context | Context | | Экземпляр класс Context (в текущей программе исключительно контекст MainActivity) | | |
| **Методы** | | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Описание |
| StorageUtil | Public | Конструктор | Context | | Установка поля context |
| setTrackList |  | ArrayList<Track> | | Сохраняет аргумент TrackList в общий файл настроек |
| getTrackList | ArrayList<Track> |  | | Подгружает из общего файла настроек TrackList |
| setTrackIndex |  | int | | Сохраняет аргумент trackIndex в общий файл настроек |
| getTrackIndex | int |  | | Подгружает из общего файла настроек trackIndex |
| setRealIndex |  | int | | Сохраняет аргумент realIndex в общий файл настроек |
| getRealIndex | int |  | | Подгружает из общего файла настроек realIndex |
| clearCachedAudioPlaylist |  |  | | Очищает в файле настроек информацию о TrackList |
| setOrderSettings |  | int | | Сохраняет аргумент OrderIndex в общий файл настроек |
| getOrderSettings | int |  | | Подгружает из общего файла настроек OrderIndex |
| storeSortSettings |  | int | | Сохраняет аргумент SortIndex в общий файл настроек |
| loadSortSettings | int |  | | Подгружает из общего файла настроек SortIndex |
| setFavTrackList |  | ArrayList<Track> | | Сохраняет аргумент FavoriteTrackList в общий файл настроек |
| getFavTrackList | ArrayList<Track> |  | | Подгружает из общего файла настроек FavoriteTrackList |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс Track** | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | | Описание | |
| path | private | String | | Содержит абсолютный путь до файла | |
| title | Содержит наименование трека | |
| album | Содержит наименование альбома | |
| artist | Содержит наименование исполнителя | |
| duration | Int | | Содержит длительность воспроизведения | |
| file\_name | String | | Содержит отображаемое имя в системе | |
| creationDate | Int | | Содержит относительную дату создания на устройстве | |
| albumIdcover | Long | | Содержит ID альбома, по которому можно получить информацию о расположении изображения альбома на устройстве | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Описание |
| Track | public | Конструктор | Все поля | | Конструктор элемента класса |
| Get/Set для каждого из Int/String поля класса | Public | Метод |  | | Предоставляет доступ к Int/String полям класса без изменений |
| getCoverImage | public | Bitmap | ContentResolver | | Получает из albumIdcover изображение |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс RecyclerView\_Adapter** | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | | Описание | |
| trackList | private | ArrayList<Track> | | Содержит текущий отображаемый список треков | |
| fullTrackList | Private | ArrayList<Track> | | Содержит полный список треков на устройстве | |
| context | private | Context | | Экземпляр класс Context (в текущей программе исключительно контекст MainActivity) | |
| file\_name | public | boolean | | Параметр фильтра – имя файла | |
| title | Public | boolean | | Параметр фильтра – наименование трека | |
| artist | Public | boolean | | Параметр фильтра – исполнитель | |
| album | Public | boolean | | Параметр фильтра – альбом | |
| filter |  | Filter | | Описывает поведение фильтра | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Описание |
| Методы базовой реализации интерфейса | Public |  |  | |  |
| filterOptions | private | Booleans | Track, String | | В зависимости от состояния полей file\_name, title, album, artist возвращает если ли в информации о Track строка String |
| setSort | Public | Void | int sort\_type, boolean less\_to\_bigger | | Сортирует отображаемое содержимое (trackList) в зависимости от аргументов |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| изменённых | заменённых | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |