

# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, РАЗПОЗНАЮЩЕЕ КОМПОЗИЦИИ МУЗЫКАЛЬНОЙ БИБЛИОТЕКИ ПОЛЬЗОВАТЕЛЯ

Комаров И.Н., Горюнова М.А., Шерстнев В.С.

Томский политехнический университет  
mag8@tpu.ru

## Введение

В наше время пользователи скачивают музыкальные файлы из различных источников и загружают их на свой телефон, не заботясь о названиях композиций. Следовательно, иногда музыкальные файлы названы некорректно, а иногда название и вовсе не несет никакого смысла (случайная последовательность символов).

Поэтому целью нашей работы являлось написание программного обеспечения, упорядочивающего музыкальную библиотеку на телефоне пользователя, путем распознавания и переименования композиций.

Программа получила название МААКИ, что является «перевертышем» слова, состоящего из первых букв каждого слова из фразы «I Know All About Music».

## Проектирование.

Приложение состоит из двух *Activity (форм)* – *Main Activity* и *File Manager Activity* (Рисунок 1).

*Main Activity* содержит два элемента управления – кнопку перехода к выбору композиций с карты памяти (переход к файловому менеджеру) и кнопку, отправляющую выбранные композиции на распознавание.

*File Manager Activity* содержит один элемент управления (кнопка *Выбрать*).

При нажатии кнопки перехода к файловому менеджеру (кнопка *Выбрать*) в *Main Activity* происходит переход на *File Manager Activity*, где пользователю предоставляется возможность выбрать композиции для распознавания. При нажатии кнопки *Выбрать* в *File Manager Activity* происходит передача массива выбранных для распознавания композиций в *Main Activity* и переход к *Main Activity*, где отображается процесс распознавания композиций.

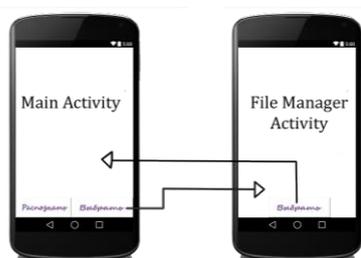


Рис. 1. Структура приложения

## Реализация.

Для реализации приложения была выбрана среда «Android Studio». Распознавание

композиций происходит с использованием сервиса «The Echo Nest».

«The Echo Nest» – музыкальная энциклопедия, позволяющая найти, описать и объединить все когда-либо выпущенные композиции, своего рода музыкальная Wikipedia. Это открытый проект, широко используемый разработчиками при написании приложений.

Сервис знает более 36 миллионов песен и около 3 миллионов исполнителей.

Обращение к сервису происходило с помощью библиотеки jEN. Библиотека jEN предоставляет доступ к методам API, таким как:

- Исполнители – поиск исполнителей по имени. Метод возвращает подробную информацию об артисте, включая аудио, подобных исполнителей, адреса сайтов и блогов, видео.
- Песни – поиск песни по исполнителю, названию, описанию или атрибуту (темп, длительность и т.д.).
- Плейлисты – создавать собственные плейлисты на основе большого спектра параметров.

Для возможности выбора пользователем музыкальных файлов был реализован файловый менеджер, предоставляющий доступ к карте памяти телефона пользователя. Для упрощения выбора композиций, которые необходимо распознать, файловый менеджер отображает только директории и mp3 файлы.

Список файлов карты памяти и список выбранных для распознавания композиций хранятся в элементах массива (*ArrayList*).

Файловый менеджер позволяет выбирать как отдельные композиции, так и папки. При выборе папки в массив файлов для распознавания помещаются все mp3 файлы из этой и любой из вложенных папок (для всех уровней вложенности).

При реализации файлового менеджера была перегружена стандартная кнопка *назад*:

- При нажатии кнопки, находясь в корневом каталоге, происходит переход к стартовому экрану и очищение списка выбранных файлов.
- При нажатии кнопки, находясь не в корневом каталоге, происходит переход к каталогу, находящемуся на уровень выше.

Работа с сервисом происходит в отдельном от GUI потоке. Для этого используется стандартный класс *AsyncTask*. *AsyncTask* создает, синхронизирует потоки, а также управляет ими,

что позволяет создавать асинхронные задачи, состоящие из операций, выполняющихся в фоновом режиме, и обновлять пользовательский интерфейс по их завершении.

Файлы, выбранные пользователем, передаются на сервис для распознавания. Элемент *Progress Bar* в *Main Activity* отражает прогресс распознавания композиций (Рисунок 2).



Рис. 2. Процесс распознавания композиций

Ответ от сервиса приходит в формате JSON. После обработки этого ответа библиотека jEN выдает объект типа Task, содержащий всю информацию о распознанном файле. С использованием этой информации происходит переименование композиций.

#### Тестирование программного продукта.

Для оценки работы программного продукта было проведено его тестирование на следующих сценариях:

1. Распознавание одного файла (Рисунки 3 и 4).

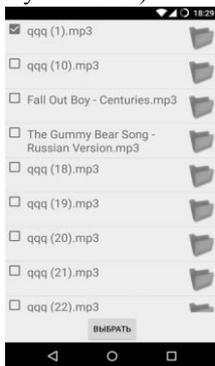


Рис. 3. Распознавание одного файла



Рис. 4. Результат распознавания одного файла

В результате работы программы тестовая музыкальная композиция была распознана.

2. Распознавание нескольких файлов из одной директории (Рисунки 5 и 6).

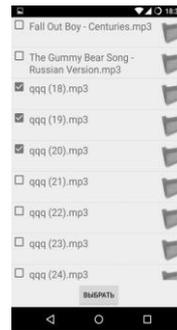


Рис. 5. Распознавание нескольких файлов



Рис. 6. Результат распознавания нескольких файлов

В результате работы программы были распознаны две из трех тестовых композиций.

3. Распознавание файлов из выбранной папки (Рисунки 7 и 8).

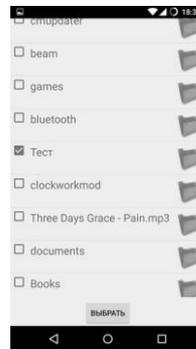


Рисунок 7 – Распознавание папки с файлами



Рисунок 8 – Результат распознавания папки с файлами

При распознавании папки с музыкальными файлами было распознано 10 из 12 композиций.

#### Заключение

По результатам тестирования можно сделать вывод, что количество распознанных композиций полностью зависит от работы сервиса «The Echo Nest». Если музыкальной композиции, отправленной приложением на распознавание к сервису, нет в базе «The Echo Nest», она распознана не будет.

Написанное приложение предоставляет пользователю возможность упорядочить его музыкальную библиотеку, а удобный и понятный интерфейс облегчает эту задачу.

#### Список литературы

1. «The Echo Nest». [Электронный ресурс]. – URL: <http://the.echonest.com> (Дата обращения: 10.04.2015)
2. Учебник по Android. [Электронный ресурс]. – URL: <http://startandroid.ru/ru/> (Дата обращения: 05.05.2015)