

## Создание приложения диктофон для Android

*Ульянов Егор Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье рассматривается создание мобильного приложения диктофон для android устройств. Разработка происходит в среде android studio. Практическим результатом является готовое мобильное приложение.

**Ключевые слова:** android, диктофон, приложение

## Creating a voice recorder app for Android

*Ulianov Egor Andreevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

This article discusses the creation of a mobile voice recorder application for android devices. Development takes place in android studio environment. The practical result is a ready-made mobile application.

**Keywords:** android, a voice recorder, app

Диктофоны помогают записать разговор, или слова, которые необходимо запомнить и в нужный момент прослушать. В Android SDK имеется множество возможностей использовать функции смартфонов. Среди них есть возможность записывать и воспроизводить аудио.

Цель данной статьи создать приложение диктофон для android устройств.

С.В. Глазков и А.Л. Ронжин рассмотрели способы формализации и анализа контекста приложений, исполняемых на мобильных гетерогенных устройствах [1]. В своей работе К.В.Рубинов протестировал большое множество интерфейсов в мобильных приложениях и разработал свой подход к тестированию [2]. А.А. Сковорода и Д.Ю. Гамаюнов рассмотрели множество мобильных приложений с вредоносной базой и предложили методы для автоматической классификации приложений [3]. Орлов В.Г. и Шаврин С.С. рассмотрели мобильные приложения для использования в системах мониторинга и диспетчеризации технологических служб [4].

Во-первых, нужно определить пользовательский интерфейс для диктофона. Он состоит из простой компоновки с тремя кнопками: одна для начала записи звука, вторая для остановки записи звука и последняя для воспроизведения записанного звука (рис.1).

```
7 <ImageView
8     android:layout_width="250dp"
9     android:layout_height="250dp"
10    android:layout_centerHorizontal="true"
11    android:layout_marginTop="20dp"
12    android:src="@drawable/logo_ssaurel"
13    android:id="@+id/logo"/>
14 <Button
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:text="Record"
18     android:textSize="20sp"
19     android:id="@+id/record"
20     android:layout_below="@id/logo"
21     android:layout_centerHorizontal="true"
22     android:layout_marginTop="15dp"
23     />
24 <Button
25     android:layout_width="wrap_content"
26     android:layout_height="wrap_content"
27     android:text="Stop"
28     android:textSize="20sp"
29     android:id="@+id/stop"
30     android:layout_below="@id/record"
31     android:layout_marginTop="10dp"
32     android:layout_centerHorizontal="true"
33     />
34 <Button
35     android:layout_width="wrap_content"
36     android:layout_height="wrap_content"
37     android:text="Play"
38     android:textSize="20sp"
39     android:id="@+id/play"
40     android:layout_below="@id/stop"
41     android:layout_marginTop="10dp"
42     android:layout_centerHorizontal="true"
43     />
44 </RelativeLayout>
```

Рисунок 1 – Добавление кнопок

Следующим шагом будет написание кода Java для основного действия. Сначала соединяем кнопки с кодом для реакции на пользовательские события (рис.2).

```
1 public class MainActivity extends AppCompatActivity {
2     private Button play, stop, record;
3     private MediaRecorder myAudioRecorder;
4     private String outputFile;
5
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_main);
10        play = (Button) findViewById(R.id.play);
11        stop = (Button) findViewById(R.id.stop);
12        record = (Button) findViewById(R.id.record);
13        stop.setEnabled(false);
14        play.setEnabled(false);
15        // ...
16    }
17
18 }
```

Рисунок 2 – Добавляем действие кнопок

Нужно отключить кнопку остановки и воспроизведения, когда действие создается, вызывая метод «setEnabled» с параметром «false». Затем нужно определить путь к файлу результатов записи. Он будет храниться в строковой переменной с именем «outputFile» (рис.3).

```
String outputFile = Environment.getExternalStorageDirectory().getAbsolutePath() + "/recording.3gp";
```

Рисунок 3 – Отключаем кнопку паузы

Далее определяем источник звука, микрофон, выходной формат, кодировщик звука и выходной файл, в котором будет сохраняться записанный звук. Код, используемый для начала записи звука, определен в реализации «OnClickListener», установленной для кнопки записи (рис.4).

```
1 record.setOnClickListener(new View.OnClickListener() {
2     @Override
3     public void onClick(View v) {
4         try {
5             myAudioRecorder.prepare();
6             myAudioRecorder.start();
7         } catch (IllegalStateException ise) {
8             // make something ...
9         } catch (IOException ioe) {
10            // make something
11        }
12        record.setEnabled(false);
13        stop.setEnabled(true);
14        Toast.makeText(getApplicationContext(), "Recording started", Toast.LENGTH_LONG)
15    }
16 });
```

Рисунок 4 – определяем устройства

Следующим шагом перед тем, как начать запись звука, необходимо вызвать метод подготовки объекта «MediaRecorder», после чего можно вызвать метод «start». Делаем то же самое, чтобы установить реализацию «OnClickListener» на кнопку остановки (рис.5).

```
1 stop.setOnClickListener(new View.OnClickListener() {
2     @Override
3     public void onClick(View v) {
4         myAudioRecorder.stop();
5         myAudioRecorder.release();
6         myAudioRecorder = null;
7         record.setEnabled(true);
8         stop.setEnabled(false);
9         play.setEnabled(true);
10        Toast.makeText(getApplicationContext(), "Audio Recorder stopped", Toast.LENGTH_LONG)
11    }
12 });
```

Рисунок 5 - OnClickListener

Далее останавливаемся для записи звука, вызывая метод «stop» объекта «MediaRecorder». Затем нужно вызвать метод выпуска. Последнее, что нужно сделать - это управлять состоянием кнопок. Отключаем кнопку остановки и включаем кнопки записи и воспроизведения. Теперь есть записанный аудиофайл, хранящийся по пути «outputFile» (рис.6).

```
1 play.setOnClickListener(new View.OnClickListener() {
2     @Override
3     public void onClick(View v) {
4         MediaPlayer mediaPlayer = new MediaPlayer();
5         try {
6             mediaPlayer.setDataSource(outputFile);
7             mediaPlayer.prepare();
8             mediaPlayer.start();
9             Toast.makeText(getApplicationContext(), "Playing Audio", Toast.LENGTH_LONG).show();
10        } catch (Exception e) {
11            // make something
12        }
13    }
14 });
```

Рисунок 6 –Управление кнопками

Чтобы воспроизвести аудиофайл, воспользуемся «MediaPlayer API». Создаем экземпляр «MediaPlayer» и устанавливаем путь к аудиофайлу для воспроизведения. Перед воспроизведением аудиофайла нужно вызвать метод подготовки объекта MediaPlayer. Наконец можно воспроизвести файл, вызвав метод «start». Теперь проверим на эмуляторе работу данного диктофона (рис.7).



Рисунок 7 – Работа диктофона

В данной статье был описан метод создания мобильного приложения диктофон на android-устройства.

### **Библиографический список**

1. Глазков С.В., Ронжин А.Л. Методы анализа контекста приложений в мобильных гетерогенных устройствах// Автоматика. Вычислительная техника. 2018. №3. С. 34-39
2. Рубинов К.В. Подход к тестированию программных интерфейсов приложений мобильных устройств // Труды Международного симпозиума «Надежность и качество». 2019. №5. С. 16-22.
3. Сковорода А.А., Гамаюнов Д.Ю. Анализ мобильных приложений с использованием моделей привилегий и api-вызовов вредоносных приложений// Современные научные исследования и инновации. 2018. № 3-5 (63). С. 42-44.
4. Орлов В.Г., Шаврин С.С. Беспроводные мобильные приложения в системах мониторинга и диспетчеризации технологических служб // Автоматика. Вычислительная техника. 2019. №1. С. 67-71