

## MD5.4 Работа с сервисами

Репозиторий с проектом:

[https://github.com/bitcoineazy/Android\\_Apps/tree/main/MD54\\_Services](https://github.com/bitcoineazy/Android_Apps/tree/main/MD54_Services)

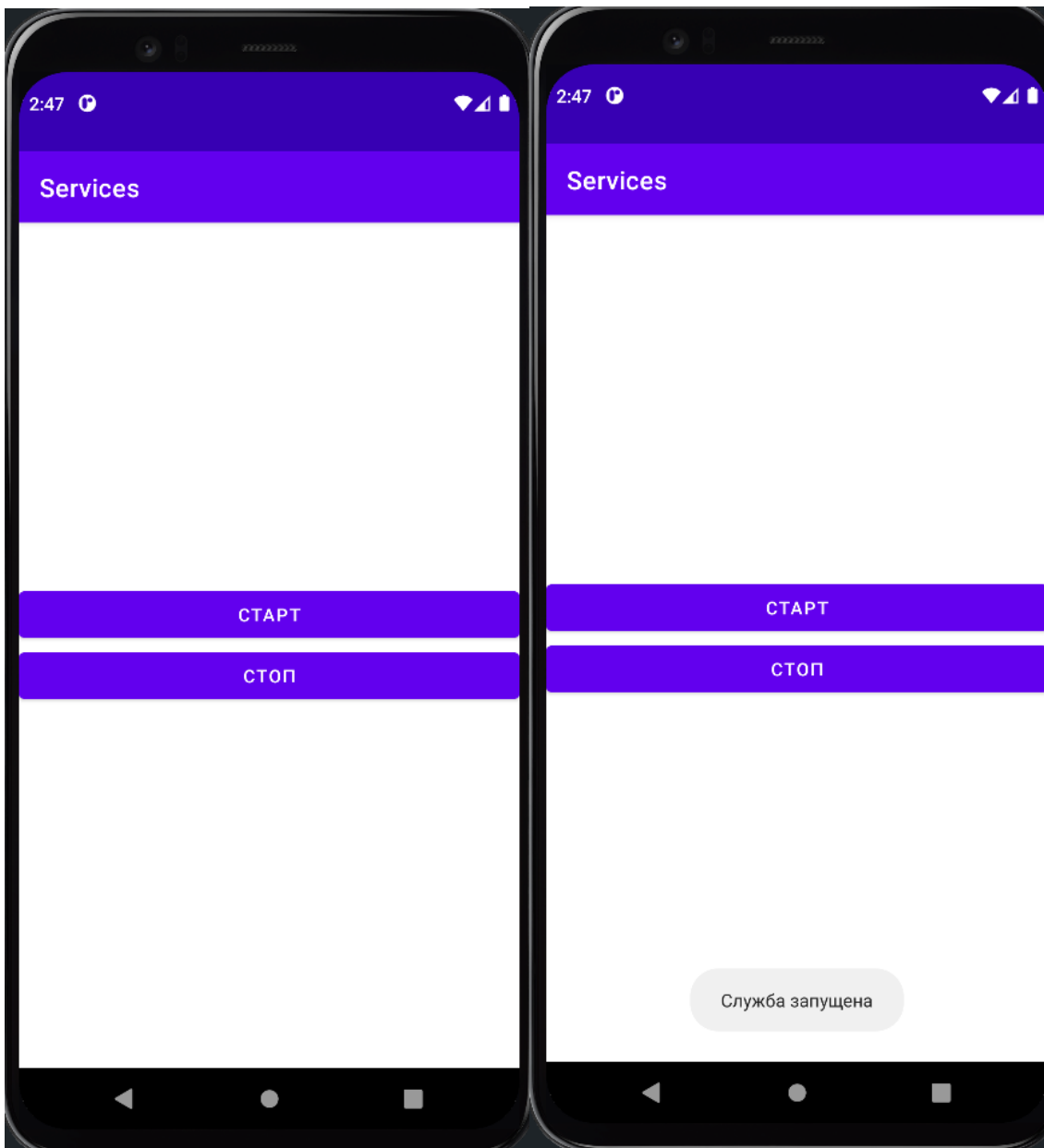
### Цель работы

Познакомится с сервисами Android Studio и научиться создавать приложения принимающие и отправляющие сообщения.

### Задания для выполнения

1. Создайте проигрыватель `PlayService.java` для воспроизведения и остановки одной песни.

Работа приложения:



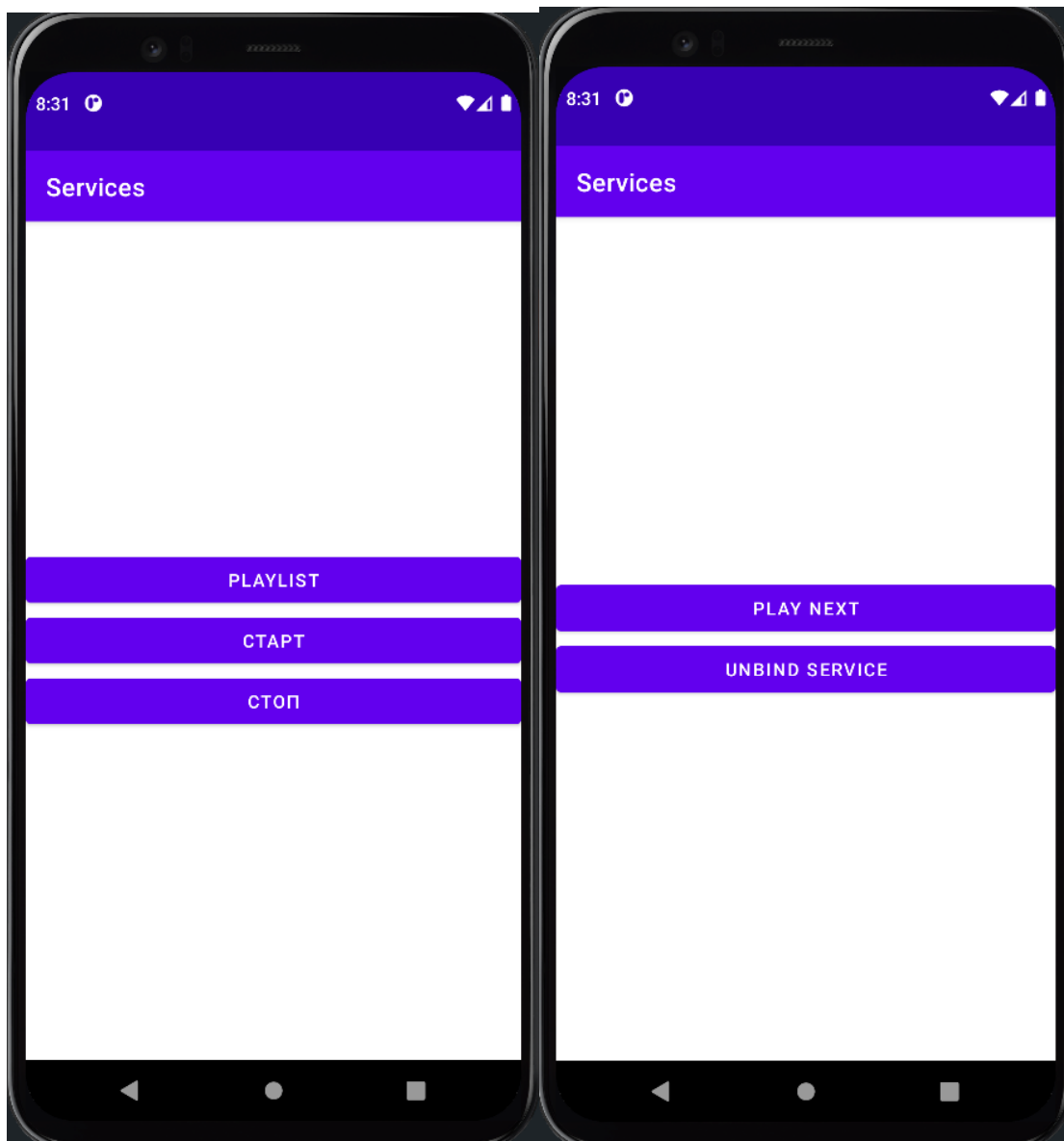


При нажатии на кнопку СТАРТ проигрыватель включает аудиозапись и выводит всплывающее уведомление Служба запущена, при нажатии на кнопку СТОП MainActivity останавливает сервис stopService и в самом сервисе в методе onDestroy() выводится всплывающее уведомление Служба остановлена и происходит освобождение проигрывателя через mPlayer.release()

### **Дополнительные задания**

2. Создайте проигрыватель PlayService.java для воспроизведения списка песен из плей-листа.

Работа приложения:



При нажатии на кнопку **PLAYLIST** открывается новая активность и запускается связанный сервис **PlayService**, который в методе **onCreate()** создаёт массив с 7-ю композициями и запускает медиапроигрыватель с первой композицией, после создается объект таймера и вызывается метод **nextSong()**, который ждёт чтобы композиция закончилась и потом воспроизводит следующую, если такая имеется.

```

33     @Override
34     public void onCreate() {
35         super.onCreate();
36
37         //Массив с треками
38         playlist = new ArrayList<>();
39         playlist.add(R.raw.hiphopbeat);
40         playlist.add(R.raw.afrobeat);
41         playlist.add(R.raw.ambientbeat);
42         playlist.add(R.raw.comedybeat);
43         playlist.add(R.raw.knights_academy);
44         playlist.add(R.raw.logopreview);
45         playlist.add(R.raw.shortlogo);
46
47         mediaPlayer = MediaPlayer.create( context: this, playlist.get(0));
48         mediaPlayer.start();
49
50         timer = new Timer();
51         if (playlist.size() > 1) nextSong();
52     }
53
54     //Следующий трек по таймеру (после завершения предыдущего)
55     public void nextSong() {
56         timer.schedule(new TimerTask() {
57             @Override
58             public void run() {
59                 mediaPlayer.reset();
60                 mediaPlayer = MediaPlayer.create( context: PlayService.this, playlist.get(++i));
61                 mediaPlayer.start();
62                 if (playlist.size() > i+1) {
63                     nextSong();
64                 }
65             }
66         }, mediaPlayer.getDuration());
67     }

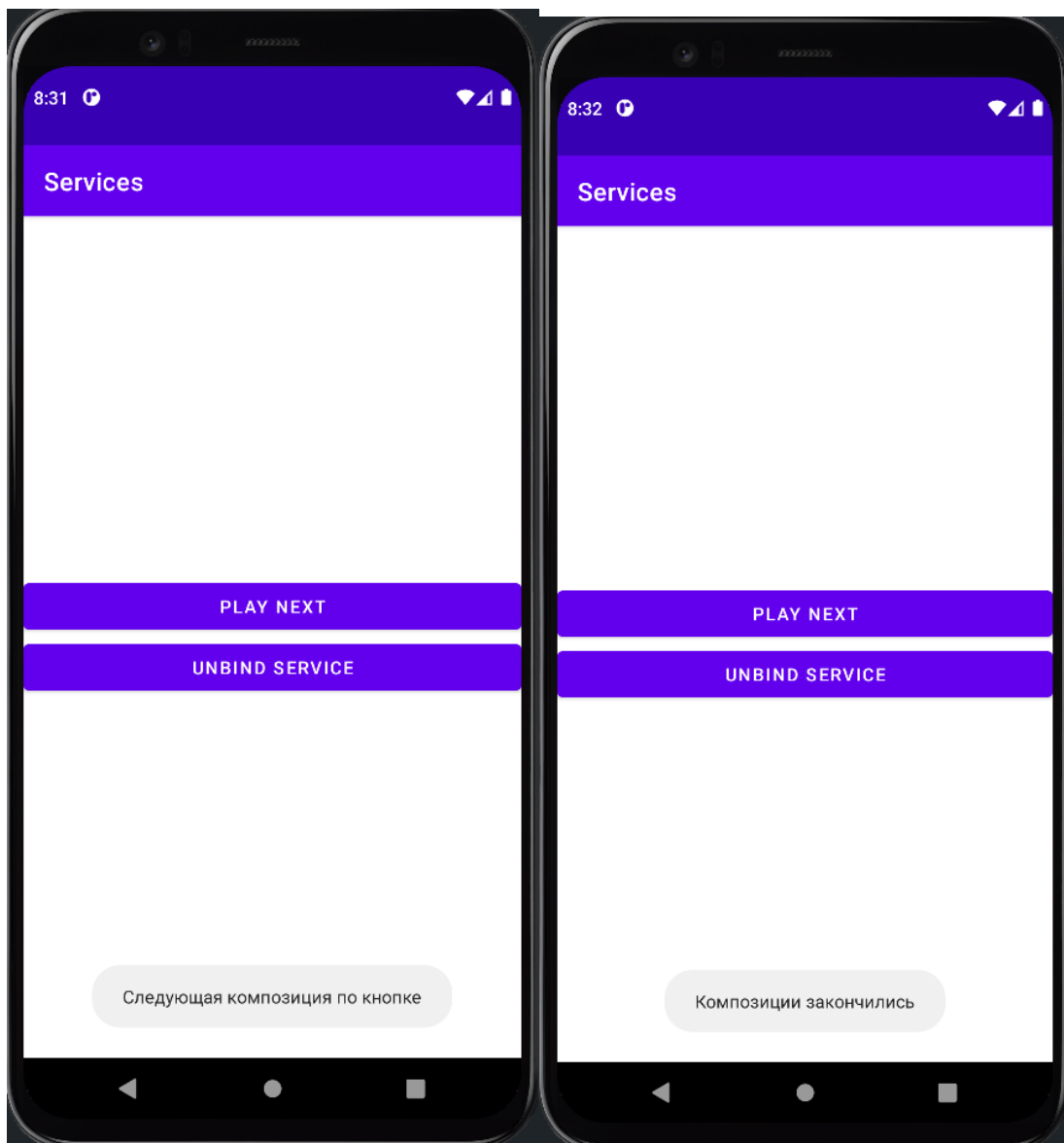
```

При нажатии на кнопку PLAY NEXT Playlist активности вызывает метод playNext() у сервиса PlayService и происходит моментальное переключение трека на следующий, если такой имеется, иначе выводится всплывающее уведомление “Композиции закончились”, сбрасывается счётчик и проигрывание снова начинается с первой композиции.

```

69     //Следующий трек по кнопке
70     public void playNext() {
71         if (mediaPlayer.isPlaying())
72             mediaPlayer.stop();
73         timer.cancel();
74         mediaPlayer.reset();
75         if (playlist.size() > i+1) {
76             Toast.makeText(PlayService.this, this, getApplicationContext(),
77                 text: "Следующая композиция по кнопке",
78                 Toast.LENGTH_SHORT).show();
79             mediaPlayer = MediaPlayer.create( context: PlayService.this, playlist.get(++i));
80             mediaPlayer.start();
81             timer = new Timer();
82             nextSong();
83         } else {
84             Toast.makeText(PlayService.this, this, getApplicationContext(),
85                 text: "Композиции закончились",
86                 Toast.LENGTH_SHORT).show();
87             i = -1;
88         }
89     }
90
91     @Override
92     public void onDestroy() {
93         if (mediaPlayer.isPlaying())
94             mediaPlayer.stop();
95         timer.cancel();
96         super.onDestroy();
97     }
98 }

```



При нажатии на кнопку UNBIND SERVICE происходит разрыв соединения со связанным сервисом PlayService, если плеер в этот момент проигрывал композицию, то она останавливается, также отменяется таймер и сервис завершает свою работу.