

## Тема: Звуковое сопровождение приложения.

Мария Александровна  
Сокольская



# Воспроизведение аудио в проекте

Фоновая  
музыка

Звуки событий,  
короткие звуки  
(нажатие на  
элемент, сдвиг,  
взрыв и т.п.)

Класс  
MediaPlayer

Класс  
SoundPool



# Ресурсы

1. Для добавления в проект ресурсов, *не требующих специальной обработки системой построения приложений Android* необходим каталог **raw** в папке **res**.
2. Добавьте в каталог **res/raw** желаемый звуковой файл.
3. Если необходимы дополнительные строки – отредактируйте файл **strings.xml**

# Программный код. Создание класса для воспроизведения фонового аудио

1. Для воспроизведения аудио и видео файлов в Андроид предусмотрен класс **MediaPlayer**. Для его использования рекомендуется написать собственный класс.

Например:

```
public class AudioPlayer {  
    private MediaPlayer mPlayer;  
    public void stop() {  
        if (mPlayer != null){  
            mPlayer.release();  
            mPlayer = null;  
        }  
    }  
}
```

Правило: удерживайте одновременно **только один объект** класса MediaPlayer (или его потомков) и **только на время воспроизведения**. Правило реализуется в методе play().

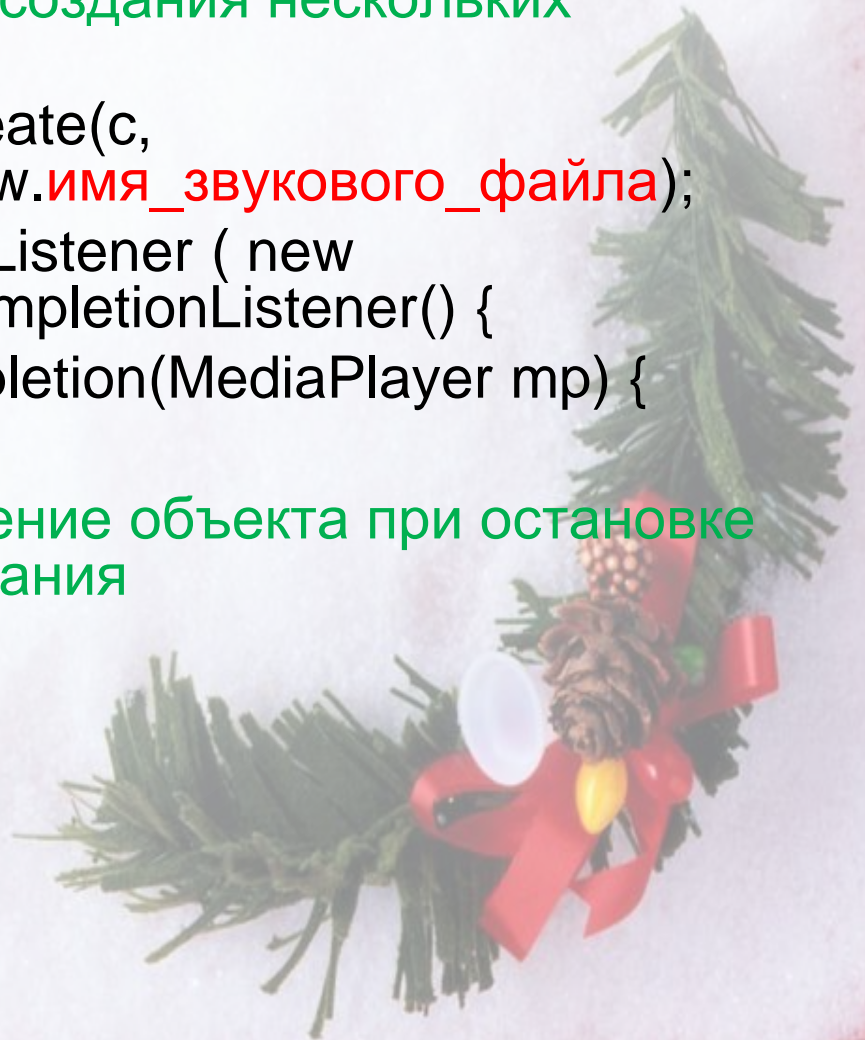


# Программный код. Создание класса для воспроизведения аудио



```
public void play(Context c){
    stop(); //предотвращение создания нескольких
           //объектов класса
    mPlayer = MediaPlayer.create(c,
                                R.raw.ИМЯ_звукового_файла);
    mPlayer.setOnCompletionListener ( new
        MediaPlayer.OnCompletionListener() {
        public void onCompletion(MediaPlayer mp) {
            stop();
            //освобождение объекта при остановке
            //использования

        }
    });
    mPlayer.start();
}
```



## Программный код. Изменения в коде активности

2. В активности, использующей MediaPlayer, объявите переменную – объект созданного класса AudioPlayer:

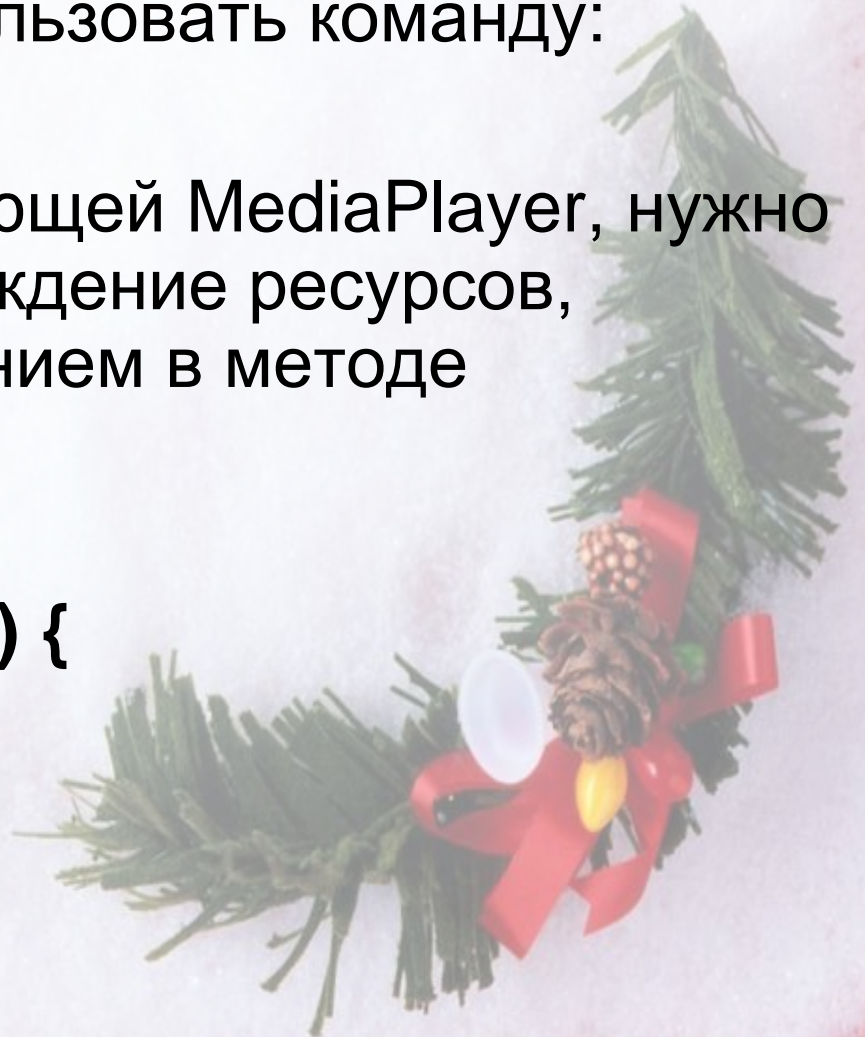
```
private AudioPlayer mPlayer = new  
    AudioPlayer();
```

3. Для запуска звука нужно в желаемом месте программы разместить команду:  
**mPlayer.play(this);**



# Программный код. Изменения в коде активности

4. Для остановки звука нужно в соответствующем месте программы использовать команду:  
**mPlayer.Stop();**
5. В активности, использующей MediaPlayer, нужно предусмотреть освобождение ресурсов, занятых воспроизведением в методе onDestroy()  
**@Override**  
**public void onDestroy() {**  
    **super.onDestroy();**  
    **mPlayer.stop();**  
**}**



# Задание.

Напишите приложение –  
поздравительную открытку



**с Новым годом!**





# Класс SoundPool

1. Создать объект SoundPool

**SoundPool sPool = new SoundPool(fCount,  
audioType, sample-rate);**

**fCount** – количество одновременно  
проигрываемых файлов

**audioType** – тип аудиопотока:

***STREAM\_MUSIC***, STREAM\_ALARM,  
STREAM\_DTMF, STREAM\_NOTIFICATION,  
STREAM\_RING, STREAM\_SYSTEM,  
STREAM\_VOICE\_CALL

**sample-rate** – сейчас значения не имеет,  
поэтому =0

# Класс SoundPool

**SoundPool sPool = new SoundPool(fCount, audioType, sample-rate);**

**fCount** – количество одновременно проигрываемых файлов

**audioType** – тип аудиопотока:

***STREAM\_MUSIC***, STREAM\_ALARM, STREAM\_DTMF, STREAM\_NOTIFICATION, STREAM\_RING, STREAM\_SYSTEM, STREAM\_VOICE\_CALL

**sample-rate** – сейчас значения не имеет, поэтому =0

**private SoundPool sPool = new SoundPool(4, AudioManager.STREAM\_MUSIC,0);**



# Класс SoundPool

2. Загружаем звуки в память

//проверка, что все загрузилось

```
sPool.setOnLoadCompleteListener(new  
    SoundPool.OnLoadCompleteListener() {  
        @Override  
        public void onLoadComplete(SoundPool soundPool, int  
            sampleId, int status) {  
            loaded = true;  
        }  
    });
```

//загрузка звуков

```
soundID1 = sPool.load(this, R.raw.имя_файла1, 1);  
soundID2 = sPool.load(this, R.raw. имя_файла2, 1);
```



# Класс SoundPool

3. В нужном месте программы используем метод `play()` с параметрами:

- *soundID* переменная с номером сэмпла. Этот номер возвращается в результате выполнения метода `load`.
- *leftVolume* значение громкости левого канала (от 0.0 до 1.0)
- *rightVolume* значение громкости правого канала (от 0.0 до 1.0)
- *priority* приоритет потока (0 - самый низкий приоритет)
- *loop* сколько раз нужно повторить сэмпл (0 не повторять, -1 - зациклить)
- *rate* скорость воспроизведения (может изменяться от 0.5 до 2.0, 1 - нормальная скорость)

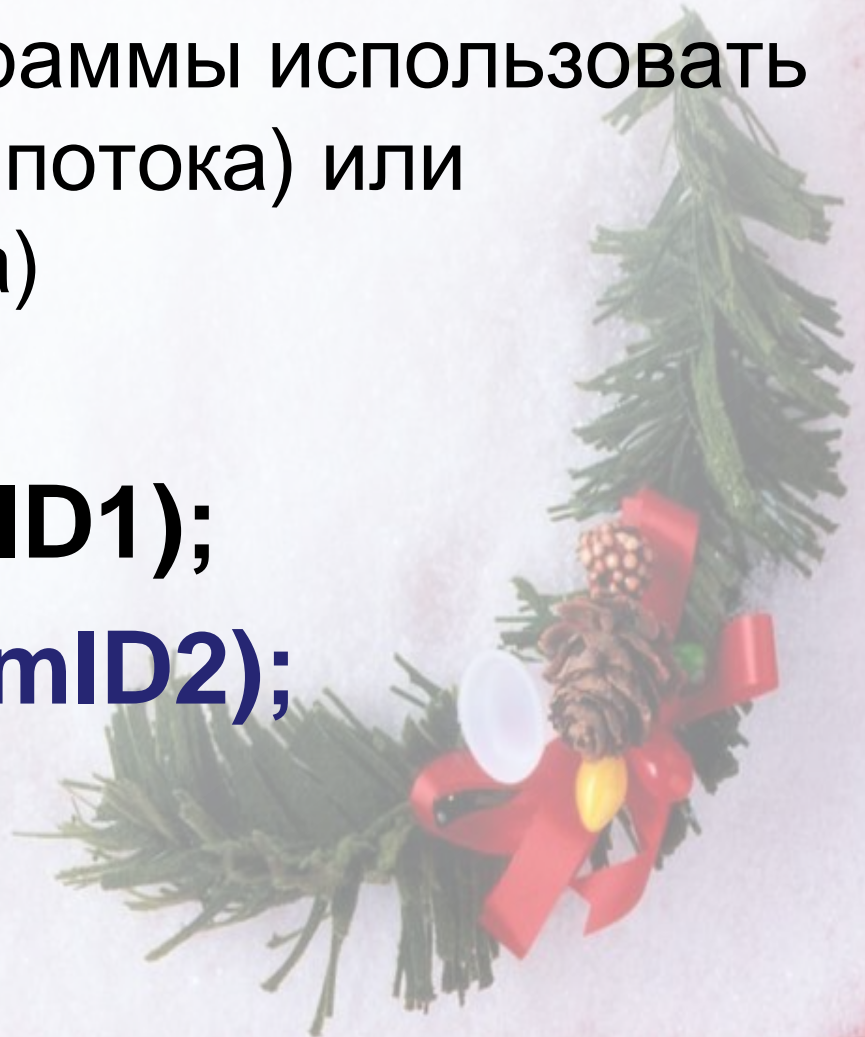
```
if(loader) {  
    streamID1 = sPool.play(soundID1,1.0f,1.0f,1,0,1f);  
    streamID2 = sPool.play(soundID2,1.0f,1.0f,1,0,1f);  
}
```

# Класс SoundPool

3. Для остановки воспроизведения в нужном месте программы использовать методы `stop(номер потока)` или `pause(номер потока)`

**`sPool.stop(streamID1);`**

**`sPool.pause(streamID2);`**





# С Новым Годом!!!



Следующее  
занятие - 12  
января

=)