МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению лабораторных работ

по дисциплине «Программирование мобильных устройств»

для студентов направлений 09.03.03 «Прикладная информатика» и 09.03.02 «Информационные системы и технологии».

Ставрополь

2021

# **Лабораторная работа 8. Ресурсы. Медиа-элементы.**

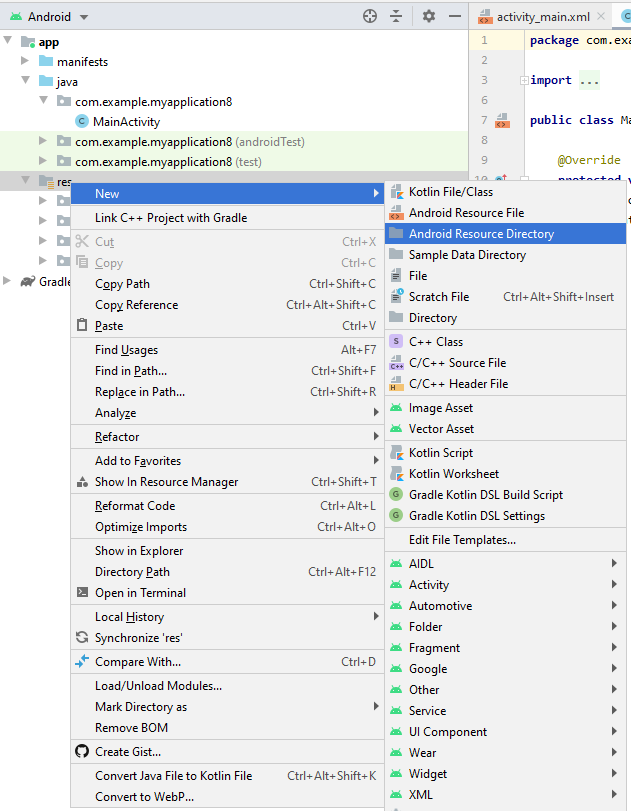
**Цель работы:** изучить способы отображения и манипулирования внешними файлами.

**Формируемые компетенции:** ПК-7

**Теоретическая часть:**

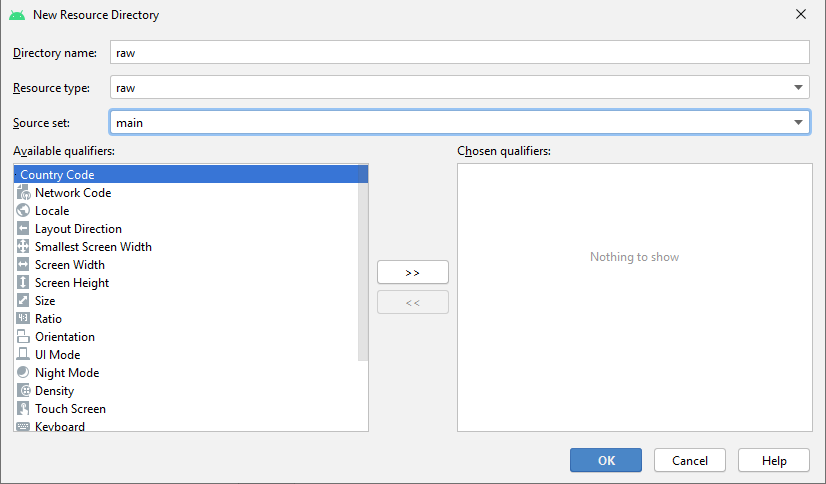
Разумеется, не всё нужное можно нарисовать и отобразить штатными средствами AndroidStudio. Поэтому, в ней есть средства для отображения внешних файлов. Начнём с простых изображений. Чтобы отобразить внешнюю картинку в проекте есть два способа, в зависимости от того, является ли он частью проекта или нет.

Если Вы хотите использовать заранее подготовленные изображения, можно смело использовать папку «Res». Как Вы помните, все файлы в данной папке распределены по категориям. Например, всё, что Вы нарисовали в AndroidStudio, располагается в папке drawable. Общие строковые значения хранятся в папке values и тд. Для внешних файлов же следует использовать папку, которую Вы сами создадите (кроме png-картинок). Она должна называться raw:

****

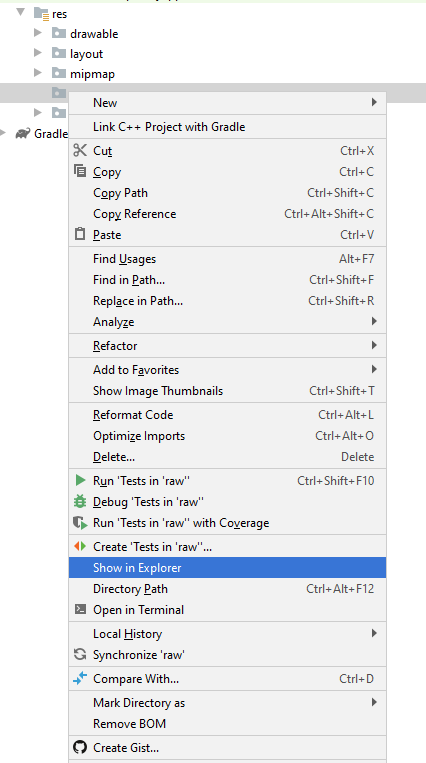
**Рисунок 1 – Добавление директории**

Назвать папку можно как угодно, но раз уж она хранит в себе raw-файлы, то пусть так и называется.

****

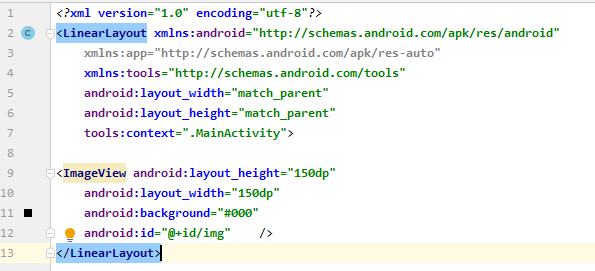
**Рисунок 2 – Указание типа и названия директории**

Скачаем любую картинку из интернета (приличную). Найдём изображение в формате png, чтобы посмотреть простой способ добавления и поместим её внутрь папки drawable. Чтобы быстро найти эту папку, нажмём правой кнопкой по ней и выберем Show in Explorer:



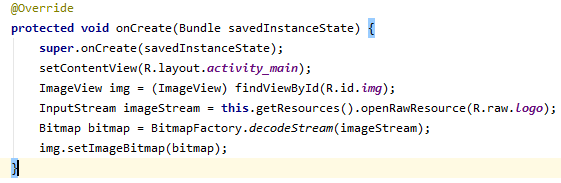
**Рисунок 3 – Контекстное меню созданной папки**

Переместим скачанное изображение в папку. Для отображения изображений используется знакомый Вам тэг – ImageView. Поместим код ниже в activity\_main.xml:



**Рисунок 4 – activity\_main.xml**

Теперь присвоим нашему ImageView в качестве источника – картинку, скачанную из интернета. Для этого перейдём в MainActivity.java и добавим следующий код:



**Рисунок 5 – Заполнение содержимого из файла**

Запустив приложение, убедимся, что всё работает:

****

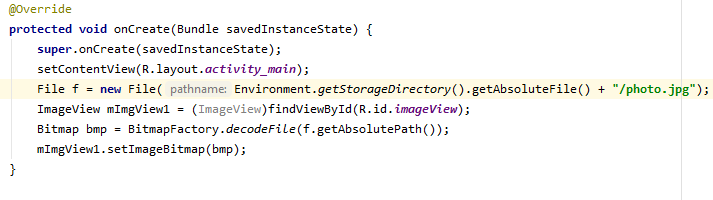
**Рисунок 6 – Результат добавления изображения**

Теперь рассмотрим вариант, когда у Вашего приложения есть доступ к устройству пользователя и Вы точно знаете, где лежит нужный Вам файл. Чтобы получить доступ к устройству, необходимо настроить соответствующее разрешение. Вообще, с разрешениями Вы познакомитесь в будущих лабораторных занятиях. Сейчас же мы познакомимся с теми, что понадобятся для выполнения именно этой. Открываем файл AndroidManifest (расположение помним из первой лабораторной) и добавляем следующую строку:

****

**Рисунок 7 – Добавление разрешения на чтение**

Данная строка позволяет получить доступ к хранилищу телефона, чтобы можно было прочитать файлы. Итак, предположим, что пользователь сохранил изображение на флешке в папке Lab в корне:



**Рисунок 8 – Открытие внешнего файла**

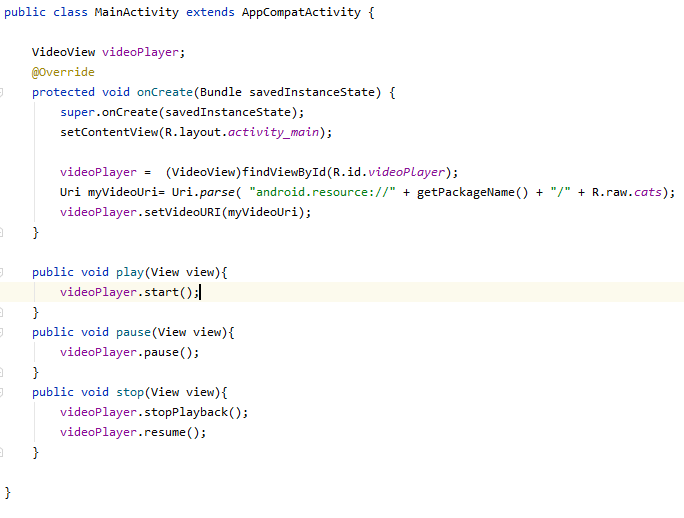
Запустим и убедимся, что всё работает.

Для добавления видео и аудио принцип не очень отличается. Рассмотрим для начала видеофайл. Скачайте и поместите видеофайл в формате .mp4 в папку raw. Добавьте в activity\_main.xml следующий код:

****

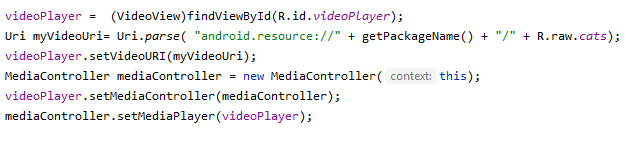
**Рисунок 9 – добавление разметки для управления пллером**

А в MainActivity это:

****

**Рисунок 10 – Добавление кода по управлению плеером**

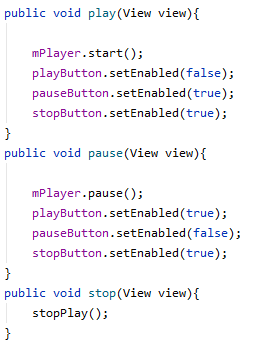
Естественно, кнопки Play и Pause можно убрать, и добавить вместо них встроенный компонент:

****

**Рисунок 11 – Добавление контроллера для плеера**

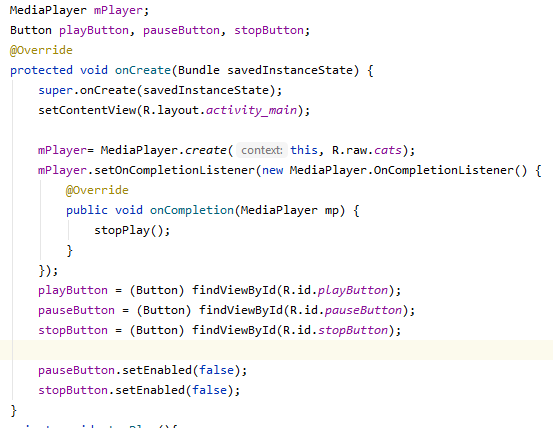
Естественно, стоит выбирать файлы, форматы которых телефон сможет прочитать. Так, распространённый формат «.mkv» не сможет быть открыт данным плеером, так как на телефоне не установлены требуемые кодеки.

Для добавления аудио принцип схож, как и с видео. Однако, для отображения аудиозаписи нам не всегда нужен интерфейс. Для простого воспроизведения воспользуемся следующим кодом. Изменим те 3 кнопки (Play, Pause, Stop) для работы с аудио:

****

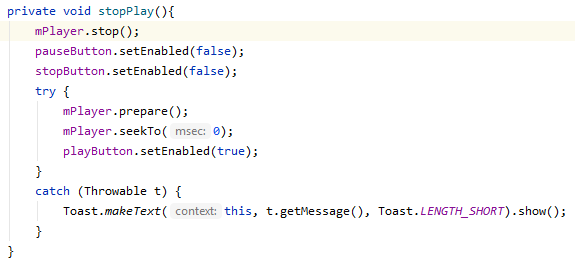
**Рисунок 12 – изменённый код обработки кнопок**

И добавим инициализацию:

****

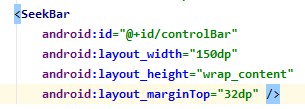
**Рисунок 13 – Инициализация MediaPlayer**

Метод stopPlay будет выглядеть следующим образом:

****

**Рисунок 14 – Метод для сброса MediaPlayer**

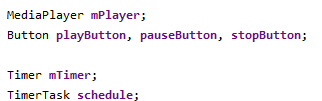
Такой подход (фоновая музыка) подходит, если есть необходимость запустить фоновую музыку (например, в главном меню игры или во время её). Если же мы хотим добавить возможность пользователю самостоятельно выбирать с какого момента воспроизводить аудио, мы можем добавить эту информацию в интерфейс. Прежде всего добавим полосу, которая будет показывать, на каком этапе :

****

**Рисунок 15 – Добавление полосы отображения текущей позиции аудиофайла**

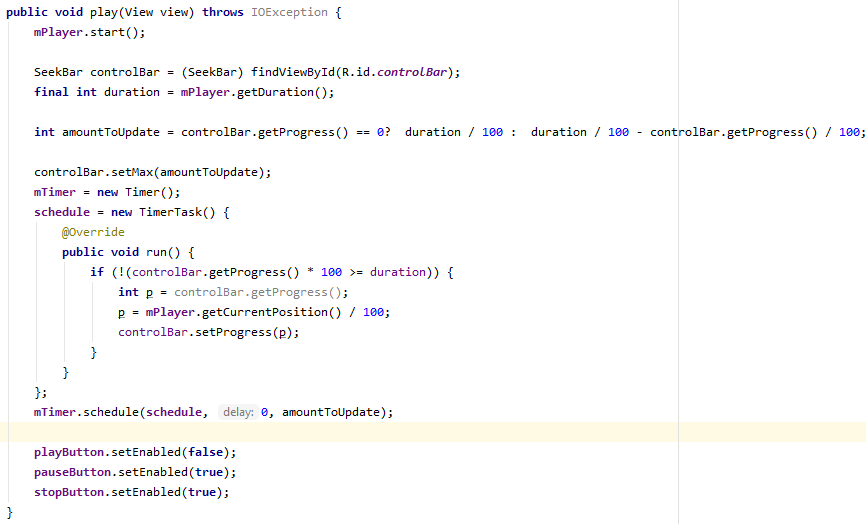
А в MainActivity:

Изменим содержимое Main activity на следующее. Добавляем две переменные (таймер), которые будут обновлять интерфейс при воспроизведении:

****

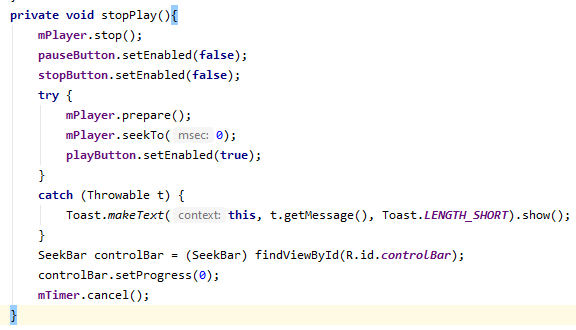
**Рисунок 16 – Определение Timer-а и расписания**

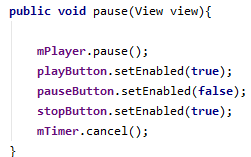
Меняем метод play:

****

**Рисунок 17 – Изменение кнопки воспроизведения**

Чтобы получить длительность аудиофайла необходимо вызвать метод mPlayer.getDuration(). amountToUpdate – вспомогательная переменная, которая будет показывать, сколько раз необходимо обновлять интерфейс. Вначале он будет равен duration/100. Когда пользователь нажимает на «Pause», то работу таймера необходимо приостановить. Timer в Java не поддерживает паузу, поэтому, необходимо его просто остановить. Остановку таймера реализуем в методах Pause и stopPlay:

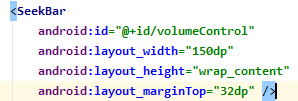
****

****

**Рисунок 18 – Изменение методов паузы и остановки**

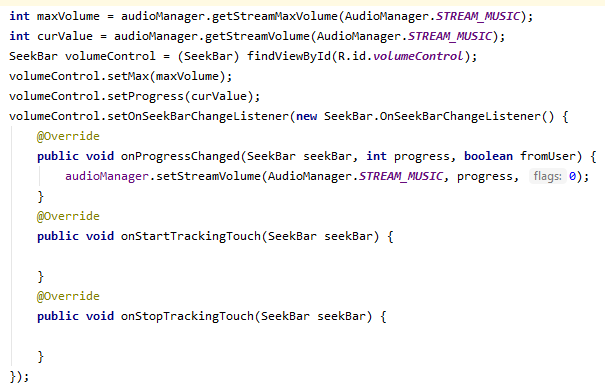
Когда пользователь снова нажимает Play после паузы, то нужно продолжить воспроизведение с момента остановки. Следовательно, мы должны правильно посчитать, сколько осталось «тиков» до полной остановки таймера.

Добавим полосу для регулирования громкости. В разметке укажем:

****

**Рисунок 19 – Добавление полосы управления звуком в разметке**

А в коде:

****

**Рисунок 20 – Обработка изменения полосы звука**

Похожим образом можно изменить и способ перемещения по controlBar. Но в данной лабораторной работе это рассмотрено не будет.

Естественно, мы не всегда хотим открывать только то, что сами добавили в проект. Пользователю тоже можно предоставить открыть нужный файл. Однако, данный пример будет рассмотрен в лабораторной работе №10.

Задание:

1. Добавьте несколько изображений в папку res. Сделайте так, чтобы при нажатии кнопок, изображение менялось на другое по очереди.
2. Добавьте в папку res видеозапись. Отобразите полосу громкости для видео и MediaController.
3. Добавьте аудиозапись в папку res. При запуске приложения данная аудиозапись должна воспроизводиться фоном и ставиться на паузу, если пользователь воспроизводит видео. При паузе видео, аудиозапись должна воспроизводиться спустя 1.5 секунды. После окончания аудио, оно должно воспроизвестись заново.

**Содержание отчета:** отчет по лабораторной работе должен быть выполнен в редакторе MSWord и оформлен согласно требованиям. Требования по форматированию: Шрифт TimesNewRoman, интервал – полуторный, поля левое – 3 см., правое – 1,5 см., верхнее и нижнее – 2 см. Абзацный отступ – 1,25. Текст должен быть выравнен по ширине.

Отчет должен содержать титульный лист с темой лабораторной работы, цель работы и описанный процесс выполнения вашей работы. В конце отчета приводятся выводы о проделанной работе.

В отчет необходимо вставлять скриншоты выполненной работы и добавлять описание к ним. Каждый рисунок должен располагаться по центру страницы, иметь подпись (Рисунок 1 – Создание подсистемы) и ссылку на него в тексте.

**Контрольные вопросы:**

1. ImageView.
2. VideoView.
3. Timer.

**Список литературы, рекомендуемый к использованию по данной теме:**

1. Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. Основы интернет-технологий: учебное пособие / Пархимович М. Н. , Липницкий А. А. , Некрасова В. А. – М.: Архангельск: ИПЦ САФУ, 2013 – 366 с.; То же [Электронный ресурс]. - URL:http://biblioclub.ru/index.php?page=book\_red&id=436379&sr=1
2. Соколова В. В. Разработка мобильных приложений: учебное пособие/ Соколова В. В. Томск: Издательство Томского политехнического университета, 2015 – 176с. То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book\_red&id=442808&sr=1