



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №12

по дисциплине «Разработка мобильных приложений»

Выполнили:

Студенты группы ИКБО-12-22

Солобай А.П.

Проверил:

Преподаватель

Степанов П.В.

Москва 2024 г.

Выполнение практической работы

Ссылка на GitHub со всеми работами: <https://github.com/Eckorezze/Mobil>

1. Реализовать приложение для работы с видео материалами в стандартном наборе виджетов применяя класс `VideoView`

Листинг 1 – код в файле `MainActivity`

```
private VideoView videoPlayer;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    videoPlayer = findViewById(R.id.videoPlayer);
    Uri myVideoUri = Uri.parse( "android.resource://" +
    getPackageName() + "/" + R.raw.spongebob);
    videoPlayer.setVideoURI(myVideoUri);
}
public void play(View view){
    videoPlayer.start();
}
public void pause(View view){
    videoPlayer.pause();
}
public void stop(View view){
    videoPlayer.stopPlayback();
    videoPlayer.resume();
}
```

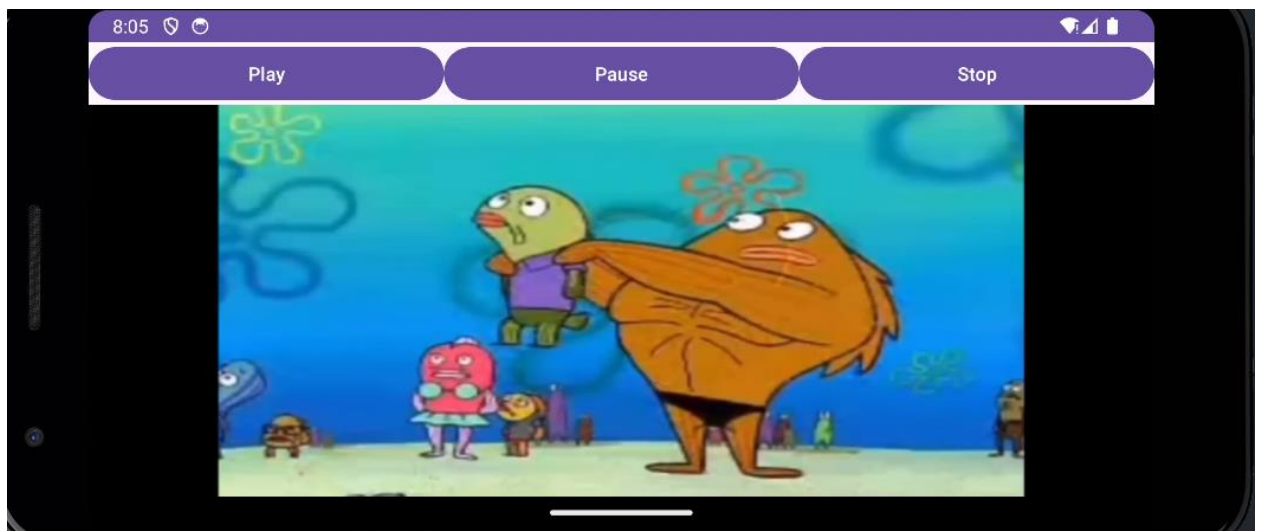


Рисунок 1 – тестирование кода

2. С помощью класса `MediaController` добавить к `VideoView` дополнительные элементы управления.

Листинг 2 – код в файле `MainActivity`

```
private VideoView videoPlayer;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    videoPlayer = findViewById(R.id.videoPlayer);
    Uri myVideoUri= Uri.parse( "android.resource://" +
getPackageName() + "/" + R.raw.spongebob);
    videoPlayer.setVideoURI(myVideoUri);
    MediaController mediaController = new MediaController(this);
    videoPlayer.setMediaController(mediaController);
    mediaController.setMediaPlayer(videoPlayer);
}
public void play(View view){
    videoPlayer.start();
}
public void pause(View view){
    videoPlayer.pause();
}
public void stop(View view){
    videoPlayer.stopPlayback();
    videoPlayer.resume();
}
```

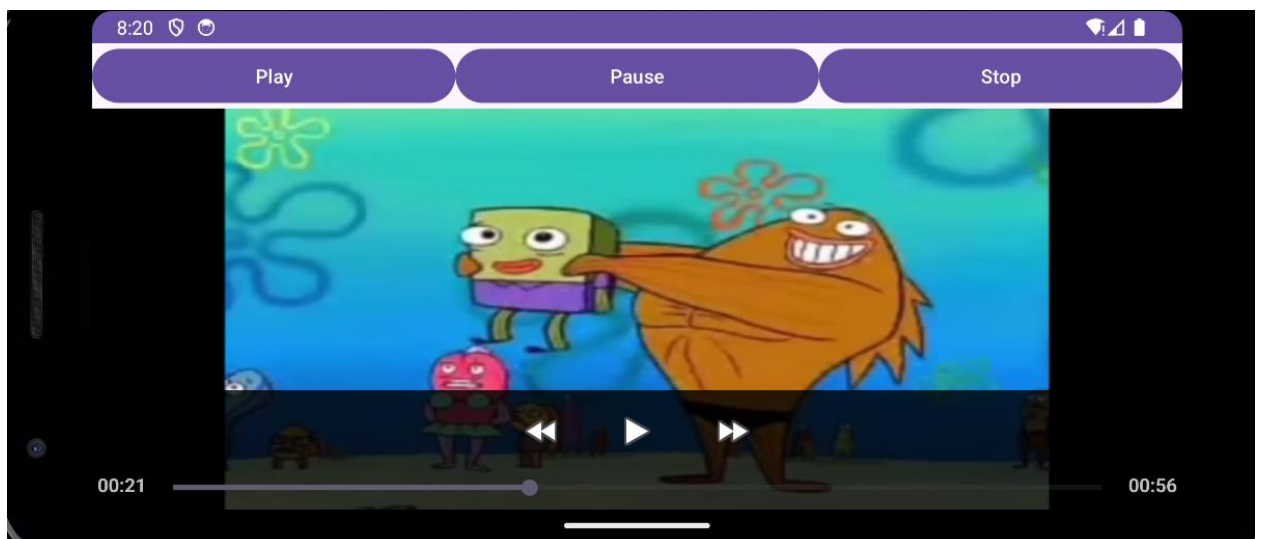


Рисунок 2 – тестирование кода

3. Реализовать приложение для работы с аудио материалами, применяя класс MediaPlayer.

Листинг 3 – код в файле MainActivity

```
MediaPlayer mPlayer;
Button playButton, pauseButton, stopButton;

SeekBar volumeControl;
AudioManager audioManager;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mPlayer=MediaPlayer.create(this, R.raw.matteblack);
    mPlayer.setOnCompletionListener(mp -> stopPlay());
    playButton = findViewById(R.id.playButton);
    pauseButton = findViewById(R.id.pauseButton);
    stopButton = findViewById(R.id.stopButton);
    audioManager = (AudioManager)
        getSystemService(Context.AUDIO_SERVICE);
    int maxVolume =

audioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC);
    int curValue =
        audioManager.getStreamVolume(AudioManager.STREAM_MUSIC);
    volumeControl = findViewById(R.id.volumeControl);
    volumeControl.setMax(maxVolume);
    volumeControl.setProgress(curValue);
    volumeControl.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SearchBar seekBar, int progress,
boolean fromUser) {
            audioManager.setStreamVolume(AudioManager.STREAM_MUSIC,
progress, 0);
        }
        @Override
        public void onStartTrackingTouch(SearchBar seekBar) {
        }
        @Override
        public void onStopTrackingTouch(SearchBar seekBar) {
        }
    });
    pauseButton.setEnabled(false);
    stopButton.setEnabled(false);
}
private void stopPlay() {
    mPlayer.stop();
    pauseButton.setEnabled(false);
    stopButton.setEnabled(false);
    try {
        mPlayer.prepare();
        mPlayer.seekTo(0);
        playButton.setEnabled(true);
    }
```

```

    }
    catch (Throwable t) {
        Toast.makeText(this, t.getMessage(),
Toast.LENGTH_SHORT).show();
    }
}
public void play(View view){
    mPlayer.start();
    playButton.setEnabled(false);
    pauseButton.setEnabled(true);
    stopButton.setEnabled(true);
}
public void pause(View view){
    mPlayer.pause();
    playButton.setEnabled(true);
    pauseButton.setEnabled(false);
    stopButton.setEnabled(true);
}
public void stop(View view){
    stopPlay();
}
@Override
public void onDestroy() {
    super.onDestroy();
    if (mPlayer.isPlaying()) {
        stopPlay();
    }
}
}

```



Рисунок 3 – тестирование кода

4. Реализовать приложение для работы с анимацией, применяя технологию Cell animation.

Листинг 4 – код примера

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    ImageView img = findViewById(R.id.animationView);  
    img.setBackgroundResource(R.drawable.rab);  
    AnimationDrawable frameAnimation = (AnimationDrawable)  
        img.getBackground();  
    // по нажатию на ImageView  
    img.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            // запускаем анимацию  
            frameAnimation.start();  
        }  
    });  
}
```

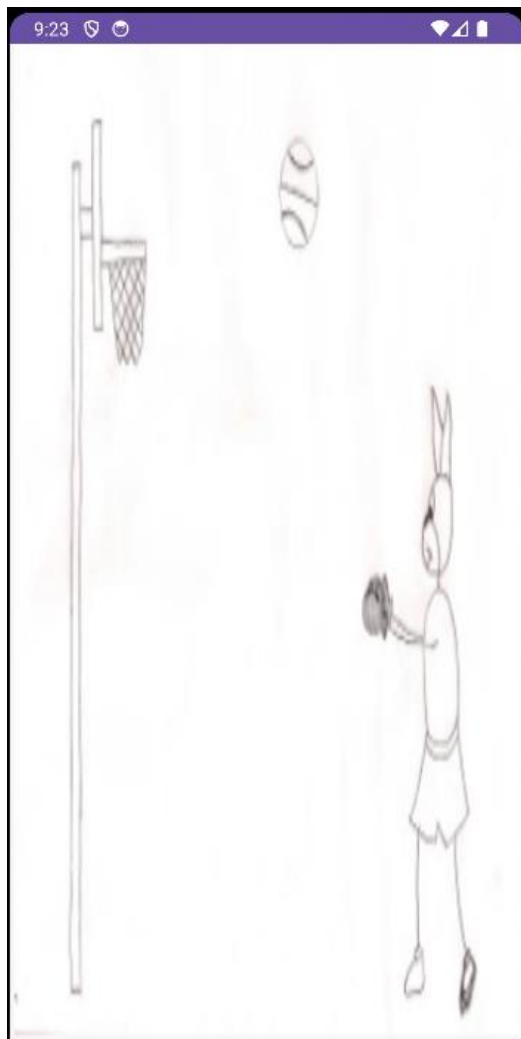


Рисунок 4 – тестирование кода

5. Реализовать приложение для работы с анимацией, применяя технологию Tween-animation. Создать приложение с сервисом со всеми необходимыми этапами жизненного цикла, применяя класс Service.

Листинг 5 – код в файле MainActivity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ImageView img = findViewById(R.id.animationView);
    img.setImageResource(R.drawable.a100);
    Animation animation = AnimationUtils.loadAnimation(this,
R.anim.common_animation);
    img.startAnimation(animation);
}
```

Листинг 6 – код в файле MediaService

```
public class MediaService extends Service {
    MediaPlayer ambientMediaPlayer;
    @Override
    public IBinder onBind(Intent intent) {
        throw new UnsupportedOperationException("Not yet
implemented");
    }
    @Override
    public void onCreate() {
        ambientMediaPlayer=MediaPlayer.create(this,
R.raw.matteblack);
        ambientMediaPlayer.setLooping(true);
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId){
        ambientMediaPlayer.start();
        return START_STICKY;
    }
    @Override
    public void onDestroy() {
        ambientMediaPlayer.stop();
    }
}
```



Рисунок 5 – тестирование Tween-animation

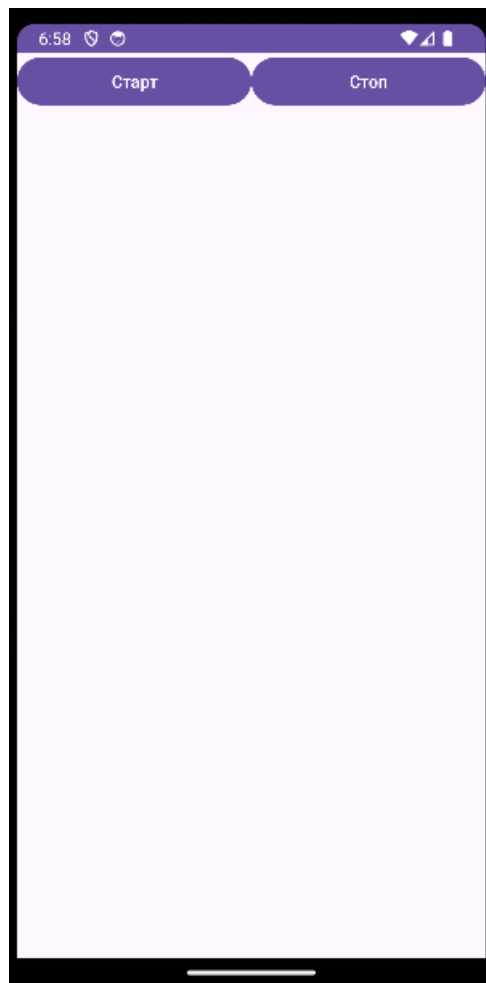


Рисунок 6 – тестирование сервиса

6. Реализовать диалоговые окна, применяя технологии DatePickerDialog и TimePickerDialog, которые позволяют выбрать дату и время.

Листинг 7 – код в файле MainActivity

```
TextView currentDateTime;  
Calendar dateAndTime = Calendar.getInstance();  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    currentDateTime = findViewById(R.id.currentDateTime);  
    setInitialDateTime();  
    public void setDate(View v) {  
        new DatePickerDialog(MainActivity.this, d,  
            dateAndTime.get(Calendar.YEAR),  
            dateAndTime.get(Calendar.MONTH),  
            dateAndTime.get(Calendar.DAY_OF_MONTH))  
            .show();  
    }  
    public void setTime(View v) {  
        new TimePickerDialog(MainActivity.this, t,  
            dateAndTime.get(Calendar.HOUR_OF_DAY),  
            dateAndTime.get(Calendar.MINUTE), true)  
            .show();  
    }  
}
```

```

}
private void setInitialDateTime() {
    currentDateTime.setText(DateUtils.formatDateTime(this,
        dateAndTime.getTimeInMillis(),
        DateUtils.FORMAT_SHOW_DATE |
        DateUtils.FORMAT_SHOW_YEAR
        | DateUtils.FORMAT_SHOW_TIME));
}
TimePickerDialog.OnTimeSetListener t = (view, hourOfDay, minute) -> {
    dateAndTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
    dateAndTime.set(Calendar.MINUTE, minute);
    setInitialDateTime();
};
DatePickerDialog.OnDateSetListener d = (view, year, monthOfYear,
dayOfMonth) -> {
    dateAndTime.set(Calendar.YEAR, year);
    dateAndTime.set(Calendar.MONTH, monthOfYear);
    dateAndTime.set(Calendar.DAY_OF_MONTH, dayOfMonth);
    setInitialDateTime();
};

```

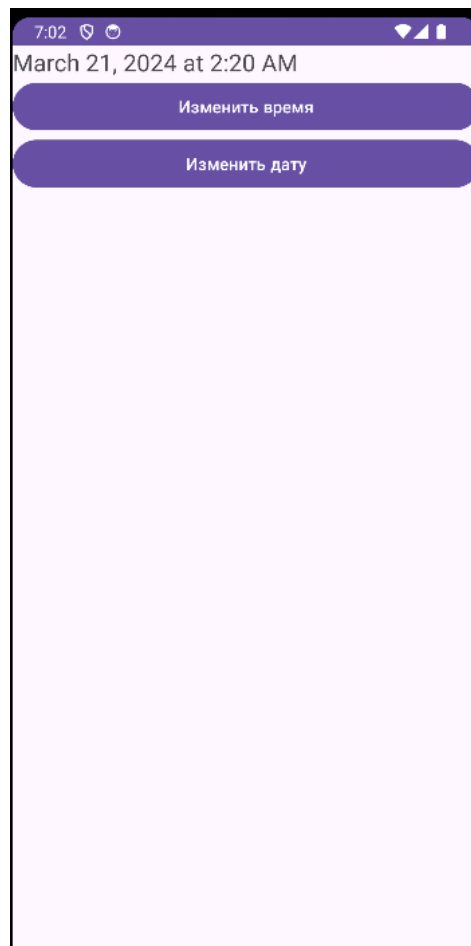


Рисунок 7 – тестирование кода

7. Применяя класс DialogFragment создать свои диалоговые окна.

Листинг 8 – код в файле CustomDialogFragment

```
public class CustomDialogFragment extends DialogFragment {
    @NonNull
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder=new
AlertDialog.Builder(getActivity());
        return builder
            .setTitle("Диалоговое окно")
            .setIcon(android.R.drawable.ic_dialog_alert)
            .setMessage("Для закрытия окна нажмите ОК")
            .setPositiveButton("ОК", null)
            .setNegativeButton("Отмена", null)
            .create();
    }
}
```

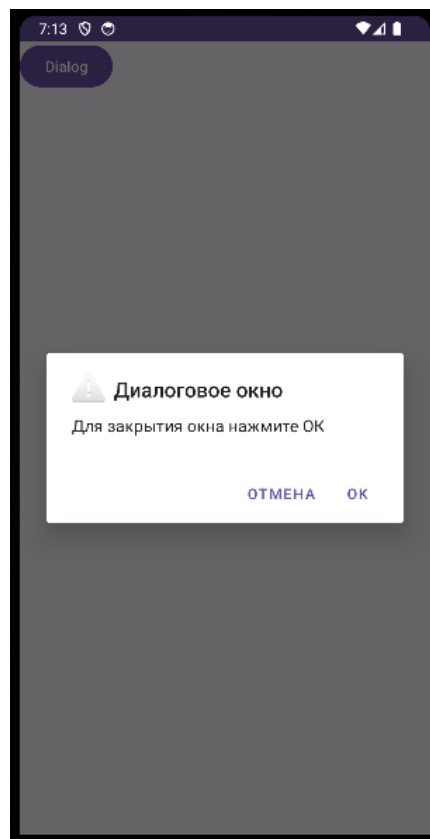


Рисунок 8 – тестирование кода

8. Реализовать передачу данных в диалоговое окно, как и в любой фрагмент, с помощью объекта Bundle.

Листинг 9 – код в файле MainActivity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ListView phonesList = findViewById(R.id.phonesList);
    ArrayList<String> phones = new ArrayList<>();
    phones.add("Google Pixel");
    phones.add("Huawei P9");
    phones.add("LG G5");
    phones.add("Samsung Galaxy S8");
    ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
        android.R.layout.simple_list_item_1, phones);
    phonesList.setAdapter(adapter);
    phonesList.setOnItemClickListener((parent, view, position, id) ->
    {
        String selectedPhone = adapter.getItem(position);
        CustomDialogFragment dialog = new CustomDialogFragment();
        Bundle args = new Bundle();
        args.putString("phone", selectedPhone);
        dialog.setArguments(args);
        dialog.show(getSupportFragmentManager(), "custom");
    });
}
```

Листинг 10 – код в файле CustomDialogFragment

```
public class CustomDialogFragment extends DialogFragment {
    @NonNull
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        String phone = getArguments().getString("phone");
        AlertDialog.Builder builder=new
AlertDialog.Builder(getActivity());
        return builder
            .setTitle("Диалоговое окно")
            .setIcon(android.R.drawable.ic_dialog_alert)
            .setMessage("Вы хотите удалить " + phone + "?")
            .setPositiveButton("ОК", null)
            .setNegativeButton("Отмена", null)
            .create();
    }
}
```

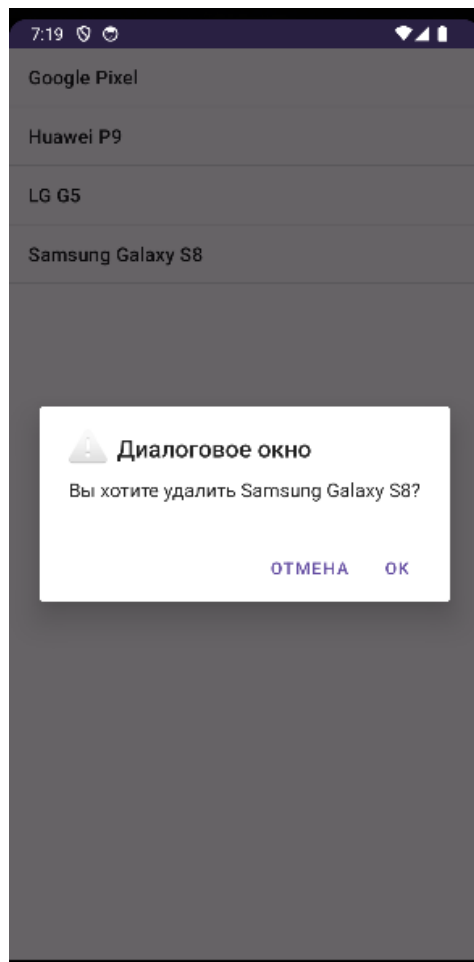


Рисунок 9 – тестирование кода

9.Реализовать взаимодействие диалогового окна с Activity

Листинг 11 – код в файле MainActivity

```
private ArrayAdapter<String> adapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    ListView phonesList = findViewById(R.id.phonesList);
    ArrayList<String> phones = new ArrayList<>();
    phones.add("Google Pixel");
    phones.add("Huawei P9");
    phones.add("LG G5");
    phones.add("Samsung Galaxy S8");
    adapter = new ArrayAdapter<>(this,
    android.R.layout.simple_list_item_1,
        phones);
    phonesList.setAdapter(adapter);
    phonesList.setOnItemClickListener((parent, view, position, id) ->
    {
        String selectedPhone = adapter.getItem(position);
        CustomDialogFragment dialog = new CustomDialogFragment();
        Bundle args = new Bundle();
```

```

        args.putString("phone", selectedPhone);
        dialog.setArguments(args);
        dialog.show(getSupportFragmentManager(), "custom");
    });
}
@Override
public void remove(String name) {
    adapter.remove(name);
}

```

Листинг 12 – код в файле CustomDialogFragment

```

public class CustomDialogFragment extends DialogFragment {
    private Removable removable;
    @Override
    public void onAttach(Context context){
        super.onAttach(context);
        removable = (Removable) context;
    }
    @NonNull
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        final String phone = getArguments().getString("phone");
        AlertDialog.Builder builder=new
AlertDialog.Builder(getActivity());
        return builder
            .setTitle("Диалоговое окно")
            .setIcon(android.R.drawable.ic_dialog_alert)
            .setMessage("Вы хотите удалить " + phone + "?")
            .setPositiveButton("ОК", (dialog, which) ->
removable.remove(phone))
            .setNegativeButton("Отмена", null)
            .create();
    }
}

```

Листинг 13 – код в файле Removable

```

public interface Removable {
    void remove(String name);
}

```

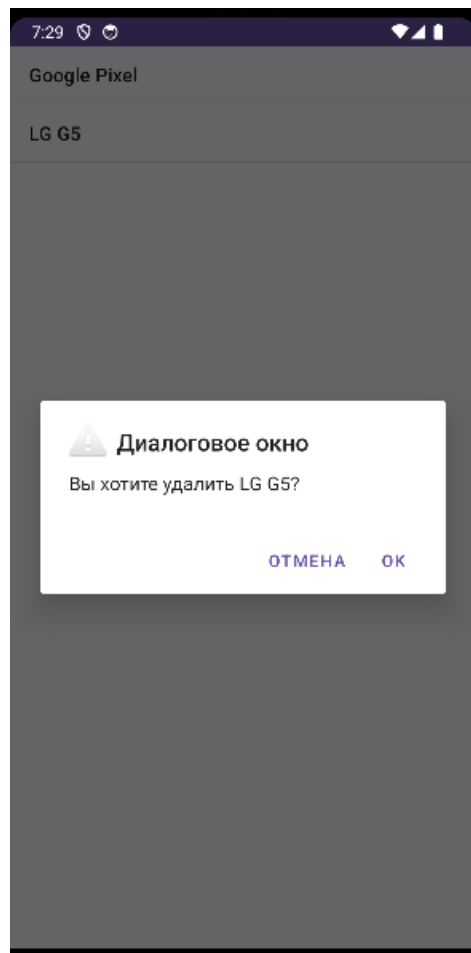


Рисунок 10 – тестирование кода

Вывод

В ходе работы мне удалось научиться работать с мультимедиа, анимацией и диалоговыми окнами.