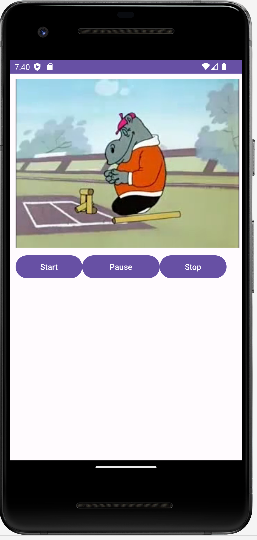
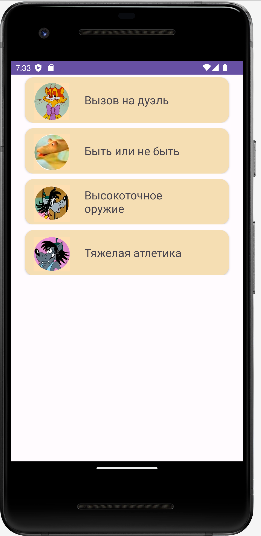
Задание

(Список из разнотипных элементов и звук)

В этом задании создается список, каждая строка которого содержит изображение героя клипа и его название. При касании пальцем строки происходит переключение на другой экран, на котором можно запустить просмотр клипа.

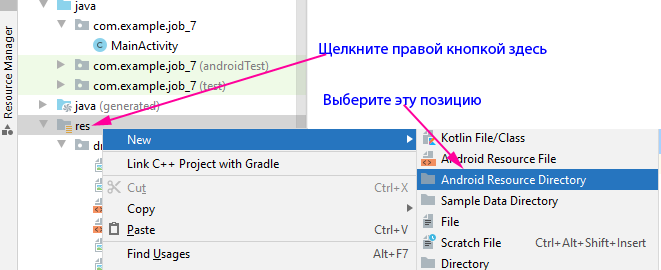


Создайте проект Android Studio (имя должно быть не MyApplication).

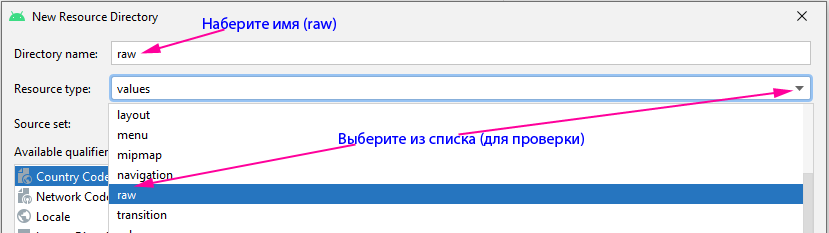
Примечание: В версии Android Studio 22.1.19 («Фламинго») надо выбирать шаблон проекта Empty Views Activity (шаблон Empty Activity запускается с языком kotlin).

Подготовительные операции

1. Откройте папку задания и скопируйте логотип logo.png в папку res/mipmap. Обычно на экран телефона будет выводиться стандартный логотип приложений Android. Вставленный логотип вы увидите, когда установите приложение на телефон. В других случаях он не виден.
2. В папку res/drawable скопируйте все фото из папки Images.
3. Создайте папку (raw) для хранения клипов и скопируйте в нее все клипы из папки movies:
4. Добавьте папку для хранения клипов:



1. Наберите имя raw



1. Возьмите у преподавателя и скопируйте во вновь созданную папку файлы клипов (с расширением mp4).
2. В папку res/values вставьте массив строк с названиями позиций меню. Для вставки массива строк:
   1. В папке res откройте папку values и щелкните правой кнопкой по строке strings.xml.
   2. В появившемся окне выберите позицию New.
   3. В следующем окне выберите позицию Values Resource File.
   4. В открывшемся окне наберите имя будущего массива (здесь, kind) и щелкните по кнопке ОК.
   5. Вставьте пустую строку между появившимися тегами resources.
   6. Вставьте строку с названием списка и его типа и закройте объявление закрывающей угловой скобкой:

<string-array name="kinds">

Она даст парную строку </string-array>, между ними вставьте требуемые строки.

* 1. Вставляйте поочередно пары тегов <item> </item> и между ними вставляйте строки:

<resources>

<string-array name="kind">

<item>Вызов на дуэль</item>

<item>Быть или не быть</item>

<item>Высокоточное оружие</item>

<item>Тяжелая атлетика</item>

</string-array>

</resources>

1. Аналогичным способом создайте массив curclip строк с именами файлов клипов:

<resources>

<string-array name="curclip">

<item>cat\_leopold.mp4</item>

<item>udav.mp4</item>

<item>wait.mp4</item>

<item>weightlifting.mp4</item>

</string-array>

</resources>

1. Откройте файл activity\_main, удалите имеющийся там элемент TextView и вставьте вместо него элемент RecyclerView.

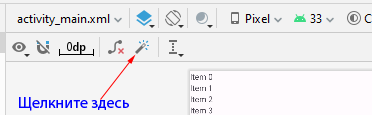
<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rv"

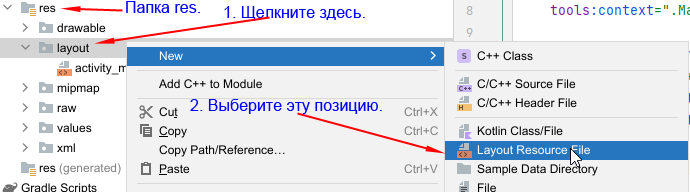
android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

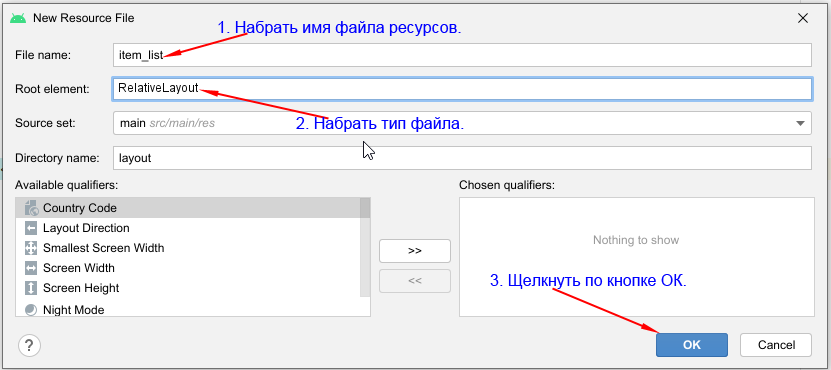
1. Перейдите на вкладку Design и щелкните по палочке, как показано.



1. Добавьте файл разметки с именем item\_list типа RelativeLayout:
   1. Щелкните правой кнопкой в папке ресурсов res по папке layout



* 1. Выполните указанные на скриншоте действия.



1. Вставьте в этот файл разметку:

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:card\_view="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:padding="5dp"

android:layout\_height="wrap\_content">

<androidx.cardview.widget.CardView

android:id="@+id/cv"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="20dp"

android:layout\_marginEnd="20dp"

app:cardCornerRadius="15dp"

app:cardBackgroundColor="#F5DEB3"

android:elevation="3dp"

app:contentPadding="5dp"

app:cardElevation="2dp"

app:strokeWidth="1dp"

app:strokeColor="@color/black">

<ImageView

android:id="@+id/imgView"

android:layout\_width="65dp"

android:layout\_height="65dp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="5dp"

app:srcCompat="@drawable/leopold"

tools:ignore="ContentDescription" />

<TextView

android:id="@+id/nameView"

android:layout\_width="196dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginStart="100dp"

android:textSize="20sp"

android:text="TextView"

tools:ignore="HardcodedText" />

</androidx.cardview.widget.CardView>

</RelativeLayout>

1. Создайте класс Picture:

public class Picture {

int img;

String str;

public Picture(int img, String str) {

this.img = img;

this.str = str;

}

public int getImg() {

return img;

}

public void setImg(int img) {

this.img = img;

}

public String getStr() {

return str;

}

public void setStr(String str) {

this.str = str;

}

}

1. Создайте адаптер для вывода списка:
   1. Добавьте класс адаптера таблицы (ClipAdapter)
   2. Вставьте члены класса адаптера (LayoutInflater и список (только ArrayList, список List не проходит)).

private final LayoutInflater inflater;

private final ArrayList< Picture > mylst;

Picture - имя класса для объектов списка.

* 1. Объявленные члены подчеркнуты красным. Щелкните по одному из них правой кнопкой и в появившемся меню Show Context Actions выберите позицию «Add constructor parameters».
  2. В появившемся окне выберите все члены класса и нажмите ОК.
  3. Вручную измените конструктор, как показано

public ClipAdapter(Context context, ArrayList<Picture> mylst) {

this.inflater = LayoutInflater.*from*(context);

this.mylst = mylst;

}

* 1. В заголовок адаптера вставить наследование

public class ClipAdapter

extends RecyclerView.Adapter<ClipAdapter.ViewHolder>{

ClipAdapter – имя адаптера.

* 1. Строка заголовка подчеркнута красным. В появившемся меню Show Context Actions выберите позицию «Implement Methods» и в следующем окне щелкните ОК.
  2. Выполните импорт классов

Версия Android Studio «Электрический угорь»

* 1. Элемент ViewHolder подчеркнут красным и в появившемся меню Show Context Actions выберите позицию Create class ViewHolder. Этот класс будет выдавать в метод класса адаптера (onBindViewHolder) представление (View) текущего элемента списка.
  2. Щелкните снова правой кнопкой в строке заголовка класса адаптера (ClipAdapter) по ViewHolder и выберите позицию

Make ViewHolder ‘extends

android.recyclerview.widget.RecyclerView.ViewHolder’

(Будет добавлено наследование в класс

ViewHolder ( RecyclerView.ViewHolder)

Версия Android Studio «Фламинго»

* 1. Строка ClipAdapter.ViewHolder подчеркнута красным и в появившемся меню Show Context Actions выберите позицию.
  2. Строка ClipAdapter.ViewHolder подчеркнута красным и в появившемся меню Show Context Actions выберите позицию:

Make ViewHolder ‘extends

android.recyclerview.widget.RecyclerView.ViewHolder’

* 1. В класс будет ViewHolder добавлено наследование

public class ViewHolder extends RecyclerView.ViewHolder {

1. Заголовок класса ViewHolder будет подчеркнут красной линией. Щелкните по нему правой кнопкой и выберите позицию Create constructor matching super. Будет создан конструктор в этом классе для базового класса.
2. В класс ViewHolder добавьте объявления элементов, которые будут обрабатываться в классе (порядок следования элементов может не совпадать с порядком их объявления в классе Bird, но должен совпадать с порядком их вывода на экран).

ImageView imgView;

TextView nameView;

CardView cardView;

1. В класс ViewHolder добавьте связи между элементами в памяти и их описанием в разметке (после строки super(itemView). Не забудьте импортировать классы)

imgView = itemView.findViewById(R.id.*imgView*);

nameView = itemView.findViewById(R.id.*nameView*);

cardView = itemView.findViewById(R.id.*cv*);

1. В метод getItemCount вставьте вместо 0 выражение mylst.size();

Здесь mylst – имя списка, объявленного как член класса адаптера.

1. В метод onCreateViewHolder вставьте вместо оператора return операторы (item\_list- это описание разметки для одной строки списка, имя задается пользователем при создании элемента Layout):

View view = inflater.inflate(R.layout.item\_list, parent, false);

return new ViewHolder(view);

1. В метод onBindViewHolder вставьте операторы:

Picture pic = mylst.get(position);

holder.imgView.setImageResource(pic.getImg());

holder.nameView.setText(pic.getStr());

Здесь Picture – имя класса для объектов списка

mylst – имя списка

imgView и nameView– имена элементов в файле item\_list (добавленный файл Layout для элементов списка.)

1. Перейдите в класс MainActivity
   1. Объявите глобально список, адаптер, список названий клипов и элемент RecyclerView в классе MainActivity

private final ArrayList<Picture> mylst = new ArrayList<>();

ClipAdapter adapter;

RecyclerView recyclerView;

String [] clp = new String[4];*//Название клипа*

* 1. Оформите метод для инициализации списка (перед последней фигурной скобкой, закрывающей класс MainActivity) но после фигурной скобки, закрывающей метод OnCreate:

public void setInitialData(){

mylst.add(new Picture(R.drawable.*leopold*, clp[0]));

mylst.add(new Picture(R.drawable.*udav*, clp[1]));

mylst.add(new Picture(R.drawable.*wolf2*, clp[2]));

mylst.add(new Picture(R.drawable.*wolf4*, clp[3]));

}

* 1. В методе OnCreate после строки

setContentView(R.layout.activity\_main);

*// Получить список клипов из ресурсов*

clp = getResources().getStringArray(R.array.*kind*);

setInitialData(); *// Вызов метода инициализации*

* 1. Добавить обработку RecyclerView

recyclerView= findViewById(R.id.*rv*);*// Связь разметки с кодом*

* 1. Лексема this будет подчеркнута красным, выберите позицию Show Context Actions и выберите позицию Suppress MissingInflatedId with annotation.
  2. Вставьте ограничение размера списка:

recyclerView.setHasFixedSize(true);

LinearLayoutManager layoutManager =

new LinearLayoutManager(this);

recyclerView.setLayoutManager(layoutManager);

*// Создание адаптера*

adapter = new ClipAdapter(this, mylst);

*// adapter.setClickListener(this); эту строку добавить после создания*

*// обработчика щелчка по выбранной позиции в списке*

*// Установка для списка адаптера*

recyclerView.setAdapter(adapter);

1. Проверьте работу приложения. На экран должен выводиться список позиций меню.
2. Подключение обработчика выбора позиции в списке.
3. В класс адаптера (ClipAdapter) скопируйте переменную для обработки щелчка по выбранной позиции (после объявления списка):

private ItemClickListener mClickListener;

1. Объявите интерфейс для этой переменной

*// Родительская activity будет вызывать этот метод в ответ на // событие щелчка.*

public interface ItemClickListener {

void onItemClick(View view, int position);

}

1. Добавьте метод, позволяющий захватить событие щелчка.

void setClickListener(ItemClickListener itemClickListener) {

this.mClickListener = itemClickListener;

}

1. В конструктор класса ViewHolder (после привязки nameView к разметке) вставьте строку:

itemView.setOnClickListener(this);

1. Лексема this будет подчеркнута красным. Щелкните по ней правой кнопкой и в появившемся меню Show Context Actions выберите позицию: Make ViewHolder implement “android.view.View.onClickListener”.
2. В добавленный метод вставьте строку:

if (mClickListener != null) mClickListener.onItemClick(view,

getAdapterPosition());

Примечание. В появившемся обработчике описание параметра представления элемента может иметь вид View v. В добавленной строке ему соответствует view. Обеспечьте соответствие.

1. Перейдите в класс MainActivity и заголовок класса вставьте объявление интерфейса:

public class MainActivity extends AppCompatActivity

implements ClipAdapter.ItemClickListener

(интерфейс добавлен после extends AppCompatActivity)

1. Заголовок класса MainActivity будет подчеркнут красным. Щелкните по нему правой кнопкой и в появившемся меню Show Context Actions выберите позицию Implement Methods.
2. В добавленный метод вставьте строку:

Toast.*makeText*(this, "Вы щелкнули по строке " + position,

Toast.*LENGTH\_SHORT*).show();

Это требуется для проверки правильности подключения обработчика.

1. После строки:

adapter = new ClipAdapter(this, mylst);

вставьте строку:

adapter.setClickListener(this);

1. Запустите проект и проверьте правильность работы. На экране должен появиться список. При щелчке по карточке должен выводиться текст и номер позиции.
2. В класс MainActivity после проверочного вывода (его лучше закомментировать) добавьте переключение на вторую страницу:

Intent intent = new Intent(MainActivity.this, ShowActivity.class);

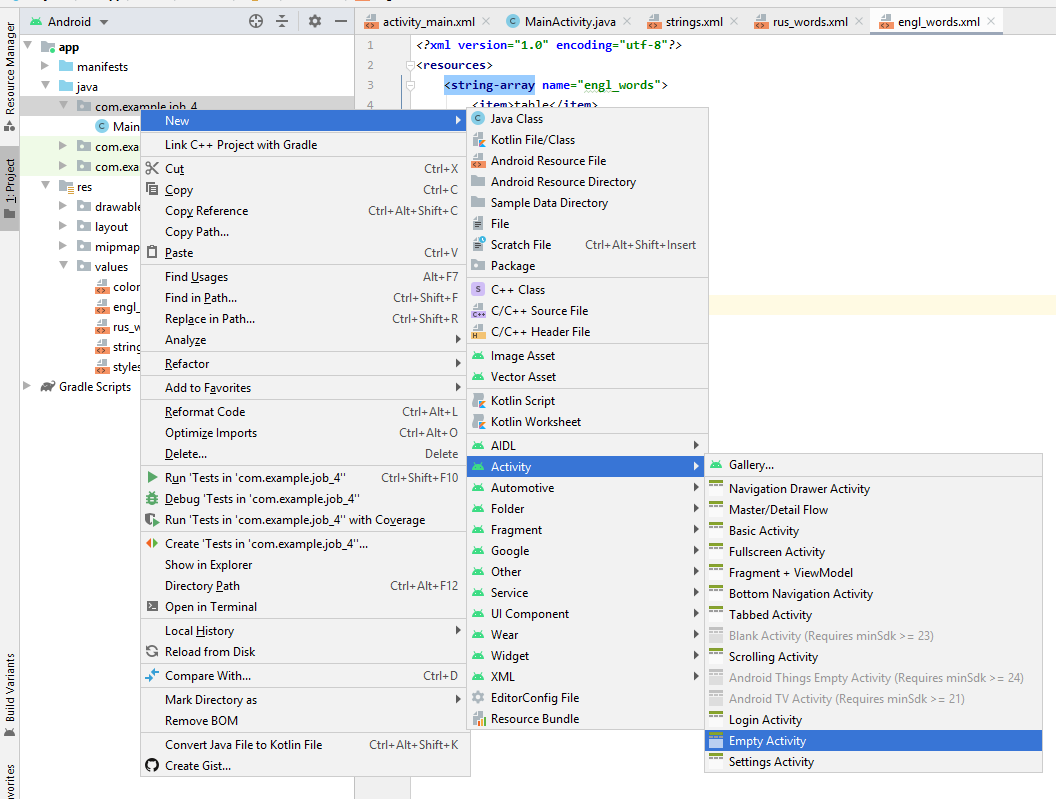
int pos = position;

*// pos – выбранная позиция, ее надо передать на вторую страницу*

intent.putExtra("pos", pos); *// “pos” имя для передачи*

this.startActivity(intent);

1. Для показа клипов добавим в приложение страницу. Для этого щелкните правой кнопкой по строке над строкой MainActivity, пройдите указанные меню и щелкните по позиции Empty Activity.



1. В появившемся окне наберите имя новой страницы (ShowActivity) и щелкните по кнопке Finish.

Примечание. В версии компилятора Фламинго выбирается Empty View Activity

1. Перейдите в файл разметки новой страницы activity\_show.xml и выполните следующие действия:
2. Замените разметку ConstraintLayout на LinearLayout
   1. Разметка должна получить вид:

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<VideoView

android:id="@+id/vid"

android:layout\_width="397dp"

android:layout\_height="558dp"

android:layout\_marginHorizontal="10dp"

android:layout\_marginVertical="10dp" />

<LinearLayout

android:layout\_width="374dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_marginBottom="20dp"

android:layout\_marginStart="10dp"

android:orientation="horizontal">

<Button

android:id="@+id/bts"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Button"

tools:ignore="ButtonStyle,HardcodedText" />

<Button

android:id="@+id/btp"

android:layout\_width="107dp"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:text="Button"

tools:ignore="ButtonStyle,HardcodedText" />

<Button

android:id="@+id/bte"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Button"

tools:ignore="ButtonStyle,HardcodedText" />

</LinearLayout>

</LinearLayout>

1. Перейдите в класс ShowActivity и добавьте глобальные объявления

VideoView vidV;

MediaController mediaControls;

Button bts, btp, bte;

1. Закрепите элементы

vidV = (VideoView) findViewById(R.id.*vid*);

bts = findViewById(R.id.*bts*);

btp = findViewById(R.id.*btp*);

bte = findViewById(R.id.*bte*);

bts.setOnClickListener(this);

btp.setOnClickListener(this);

bte.setOnClickListener(this);

Если закрепления элементов будут подчеркнуты красным, щелкните по одной из них правой кнопкой выберите позицию Show Context Actions и выберите позицию Suppress MissingInflatedId with annotation.

1. Добавьте (через this) обработчик щелчков по кнопкам

(Make ‘ShowActivity’ implement ‘android.view.View.OnClickListener’)

@Override

public void onClick(View view) {

}

1. Вставьте в этот обработчик

if (mediaControls == null) {

*// create an object of media controller class*

mediaControls = new MediaController(ShowActivity.this);

mediaControls.setAnchorView(vidV);

}

*// set the media controller for video view*

vidV.setMediaController(mediaControls);

1. Оформите обработку за каждой кнопкой (вставка после текста, вставленного в пункте)
2. В версии компилятора Android Studio Электрический угорь

switch (view.getId()) {

case R.id.*bts*:

vidV.start();

break;

case R.id.*btp*:

vidV.pause();

case R.id.*bte*:

if (vidV!=null) {

vidV.stopPlayback();

vidV.resume();

}

break;

}

1. В версии компилятора Android Studio Фламинго

int n = 0;

if(view.getId() == R.id.*bts*) n = 1;

if(view.getId() == R.id.*btp*) n = 2;

if(view.getId() == R.id.*bte*) n = 3;

switch (n) {

case 1:

vidV.start();

break;

case 2:

vidV.pause();

case 3:

if (vidV!=null) {

vidV.stopPlayback();

vidV.resume();

}

break;

}

1. Оформите прием выбранной позиции (вставка после строки bte.setOnClickListener(this);).

*// Прием выбранной позиции с предыдущей страницы*

Bundle arguments = getIntent().getExtras();

int ind = arguments.getInt("pos");

clipchoose(ind);*// Вывод на экран выбранного клипа*

1. Оформите метод выбора клипа

public void clipchoose(int ind){

switch (ind) {

case 0:

vidV.setVideoURI(Uri.*parse*("android.resource://"

+ getPackageName() + "/" + R.raw.*cat\_leopold*));

break;

case 1:

vidV.setVideoURI(Uri.*parse*("android.resource://"

+ getPackageName() + "/" + R.raw.*udav*));

break;

case 2:

vidV.setVideoURI(Uri.*parse*("android.resource://"

+ getPackageName() + "/" + R.raw.*wait*));

break;

case 3:

vidV.setVideoURI(Uri.*parse*("android.resource://"

+ getPackageName() + "/" + R.raw.*weightlifting*));

Intent intent = new Intent(ShowActivity.this, MainActivity.class);

this.startActivity(intent);

break;

}

}

1. Проверьте работу приложения и предъявите его преподавателю.