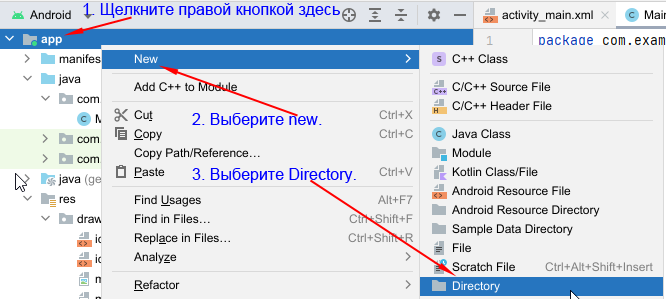
Задание

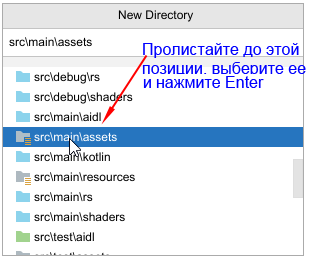
Логотипы институтов и их здания.

В папку drawable можно загрузить только один список. В данном задании реализованы два списка (один в папке drawable, другой в папке assets).

1. Создайте проект.
2. В папку drawable скопируйте логотипы
3. Создайте папку assets. Для этого щелкните правой кнопкой по папке app и пройдите по цепочке



* 1. В появившемся окне выберите src\main\assets.



1. Создайте вложенную папку. Для этого щелкните правой кнопкой по папке assets и в появившемся окне наберите имя папки building и нажмите Enter.



1. В новую папку скопируйте файлы из папки «Фото институтов».
2. Измените файл разметки activity\_main.
   1. Замените контейнер constraints на Layout (не забудьте вставить ориентацию по вертикали).
   2. Вставьте объявление Recycler

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Институты"

tools:ignore="HardcodedText" />

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rv"

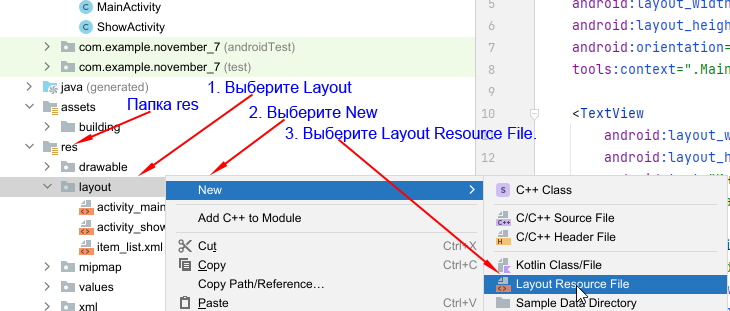
android:layout\_marginTop="10dp"

android:layout\_width="match\_parent"

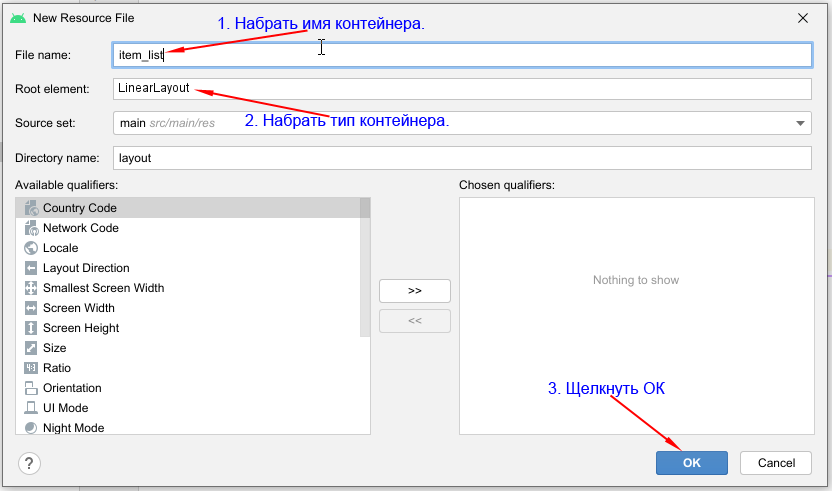
android:layout\_height="wrap\_content" />

</LinearLayout>

1. Добавьте файл разметки item\_list
   1. Откройте папку res, выберите папку layout и щелкните по ней правой кнопкой.



* 1. В появившемся окне наберите имя файла (item\_list) и наберите вид контейнера (LinearLayout).



* 1. Вставить элементы и выполнить их настройку:

<LinearLayout

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<ImageView

android:id="@+id/imgView"

android:layout\_width="87dp"

android:layout\_height="64dp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="5dp"

app:srcCompat="@drawable/madilogo"

tools:ignore="ContentDescription" />

<TextView

android:id="@+id/nameView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="TextView"

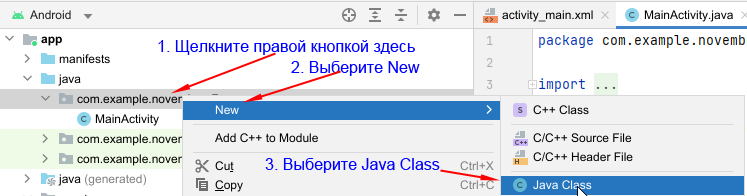
android:layout\_marginStart="20dp"

android:layout\_marginTop="20dp"

tools:ignore="HardcodedText" />

</LinearLayout>

1. Добавьте класс для объектов списка



1. В появившемся окне наберите имя класса (Build) с заглавной буквы.
2. В созданный класс добавьте члены класса и метода для обращения к ним:

public class Build {

int img;

String name;

public Build(int img, String name) {

this.img = img;

this.name = name;

}

public int getImg() {

return img;

}

public void setImg(int img) {

this.img = img;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

1. Добавьте класс адаптера таблицы (ArrayAdapter)
2. Вставьте члены класса адаптера (LayoutInflater и список (только ArrayList, список List не проходит)).

private final LayoutInflater inflater;

private final ArrayList<Build> mylst;

Build - 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 ArrayAdapter(Context context, ArrayList<Build> mylst) {

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

this.mylst = mylst;

}

1. В заголовок адаптера вставить наследование

extends RecyclerView.Adapter<ArrayAdapter.ViewHolder>{

ArrayAdapter – имя адаптера.

1. Строка заголовка подчеркнута красным. в появившемся меню Show Context Actions выберите позицию «Implement Methods».
2. В следующем окне щелкните ОК.
3. Выполните импорт классов
4. Строка ViewHolder подчеркнута красным и в появившемся меню Show Context Actions выберите позицию Create class ViewHolder. Этот класс выдает в метод класса адаптера (onBindViewHolder) представление (View) текущего элемента списка.
5. Щелкните снова правой кнопкой в строке заголовка класса адаптера (ArrayAdapter) по ViewHolder и выберите позицию

Make ViewHolder extends android.recyclerview.widget.RecyclerView.ViewHolder

(Будет добавлено наследование в класс ViewHolder ( 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.

В класс ViewHolder добавьте объявления элементов, которые будут обрабатываться в классе (порядок следования элементов может не совпадать с порядком их объявления в классе Build, но должен совпадать с порядком их вывода на экран).

ImageView imgView;

TextView nameView;

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*);

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:

Build bl = mylst.get(position);

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

holder.nameView.setText(bl.getName());

Здесь Build – имя класса для объектов списка

mylst – имя списка

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

1. Объявите список и адаптер в классе MainActivity

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

ArrayAdapter adapter;

RecyclerView recyclerView;

* 1. оформите метод для инициализации списка (перед последней фигурной скобкой, закрывающей класс MainActivity).

public void setInitialData(){

mylst.add(new Build(R.drawable.*madilogo*, "Автодорожный"));

mylst.add(new Build(R.drawable.*mailogo*, "Авиационный"));

mylst.add(new Build(R.drawable.*meilogo*, "Энергетический"));

mylst.add(new Build(R.drawable.*mgulogo*, "Московский университет"));

mylst.add(new Build(R.drawable.*mvtulogo*,"Бауманский"));

}

* 1. В методе OnCreate после строки setContentView(R.layout.activity\_main);

setInitialData(); *// Вызов метода инициализации*

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

recyclerView.setHasFixedSize(true);

LinearLayoutManager layoutManager = new LinearLayoutManager(this);

recyclerView.setLayoutManager(layoutManager);

*// Создание адаптера*

adapter = new ArrayAdapter(this, mylst);

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

*// обработчика щелчка по выбранной позиции в списке*

*// Установка для списка адаптера*

recyclerView.setAdapter(adapter);

1. Подключение обработчика выбора позиции в списке.
   1. В класс адаптера (ArrayAdapter) скопируйте переменную для обработки щелчка по выбранной позиции:

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 вставьте строку:

itemView.setOnClickListener(this);

* 1. (Заголовок класса ViewHolder) this будет подчеркнуто красным. Щелкните по нему правой кнопкой и в появившемся меню Show Context Actions выберите позицию Implement Methods.
  2. В добавленный метод вставьте строку:

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

getAdapterPosition());

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

public class MainActivity extends AppCompatActivity

implements ArrayAdapter.ItemClickListener

* 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.

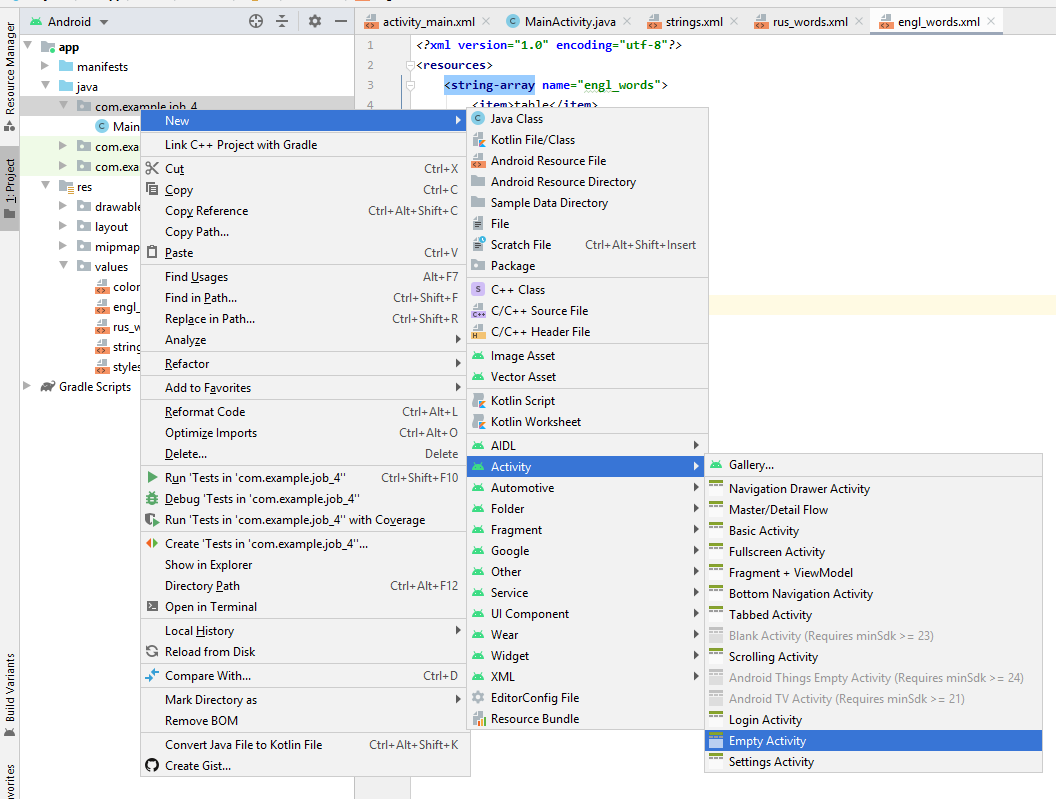
* 1. После строки:

adapter = new BirdAdapter(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 номер позиции.
2. Добавим в приложение страницу. Для этого щелкните правой кнопкой по строке MainActivity, пройдите указанные меню и щелкните по позиции Empty Activity.



1. В появившемся окне наберите имя новой страницы (ShowActivity) и щелкните по кнопке Finish.
2. В обработчике выбора позиции замените строку Toast на операторы переключения на другую 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. Перейдите в файл activity\_show
   1. Замените контейнер constraints на Layout.
   2. Добавьте ориентацию по вертикали.
   3. Вставьте элементы ImageView, TextView и кнопку Button. Разметка должна получить вид:

<LinearLayout

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

tools:context=".ShowActivity">

<ImageView

android:id="@+id/imgView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

tools:ignore="ContentDescription" />

<TextView

android:id="@+id/txtView"

android:layout\_width="match\_parent"

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

<Button

android:id="@+id/button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="ОК" />

</LinearLayout>

1. Объявите на ShowActivity глобально массив и инициализируйте его:

String [] str = {"madi.png", "mai.png", "mei.png", "mgu.png", "mvtu.png", };

1. Объявите глобально массив и инициализируйте его:

String [] desc = {"МАДИ", "МАИ", "МЭИ", "МГУ", "МВТУ"};

1. Объявите глобально элементы

ImageView img;

TextView txt;

Button bt;

1. Свяжите элементы в коде с разметкой:

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

txt = findViewById(R.id.*txtView*);

bt = findViewById(R.id.*button*);

1. Обеспечьте прием выбранной позиции

*// Прием выбранной позиции с предыдущей страницы*

Bundle arguments = getIntent().getExtras();

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

1. Добавьте заполнение подписи под изображением:

txt.setText(desc[ind]);

1. Выполните считывание изображения из папки assets:

try

{

*// Получение входного потока*

InputStream ims = getAssets().open("building/" + str[ind]);

*// load image as Drawable*

Drawable d = Drawable.*createFromStream*(ims, null);

*// Связать изображение с элементом на экране*

img.setImageDrawable(d);

img.setScaleType(ImageView.ScaleType.*FIT\_XY*);

ims .close();

}

catch(IOException ignored)

{

}

1. Добавьте обработчик щелчка по кнопке

bt.setOnClickListener(this);

1. Слово this подчеркнуто красным. Щелкните по нему правой кнопкой и в появившемся меню Show Context Actions выберите позицию:

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

1. В обработчик щелчка по кнопке вставьте операторы возврата на предыдущую страницу.

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

this.startActivity(intent);

1. Проверьте работу приложения и предъявите ее преподавателю.