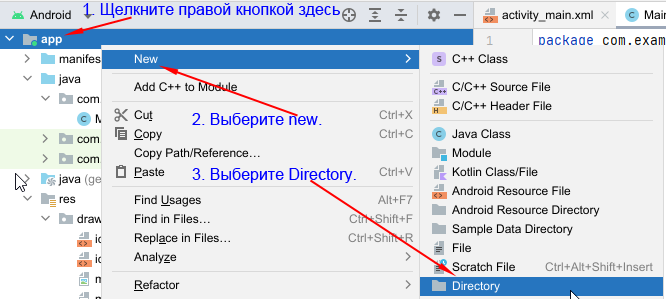
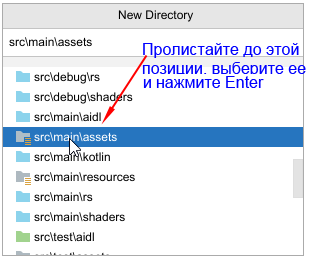
Задание (вложенные списки)

В предыдущем задании был рассмотрен пример реализации двух списков изображений, в котором одному рисунку из первого списка соответствовал один рисунок из второго списка. Фактически это реализация реляционного отношения один к одному. В данном примере рассматривается реализация вложенных списков, то есть, реализация реляционного отношения один ко многим. Один список (логотипы) в папке drawable, другой в assets. Так как структура обоих списков одинакова (рисунок и строка текста), то адаптер для обоих списков один и тот же, но разная инициализация списков.

1. Создайте проект.
2. В папку drawable скопируйте файлы из папки KindLogo.
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 Category и нажмите Enter.
2. В папке Category создайте вложенные папки Cats, Dogs, Fishes, Insects. (щелкните по папке Category правой кнопкой → New → Directory → набрать имя).
3. В новую папку Cats скопируйте файлы из папки Cats задания.
4. 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">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rv"

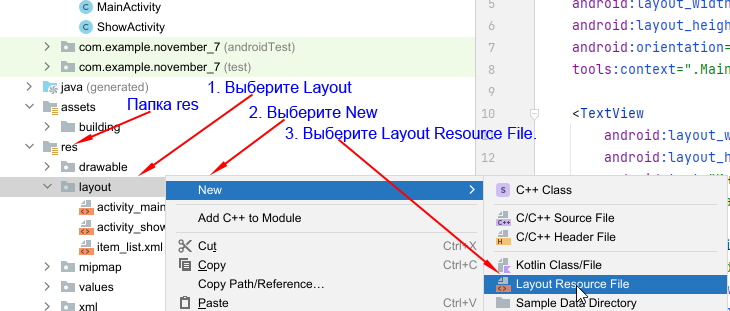
android:layout\_marginTop="10dp"

android:layout\_width="match\_parent"

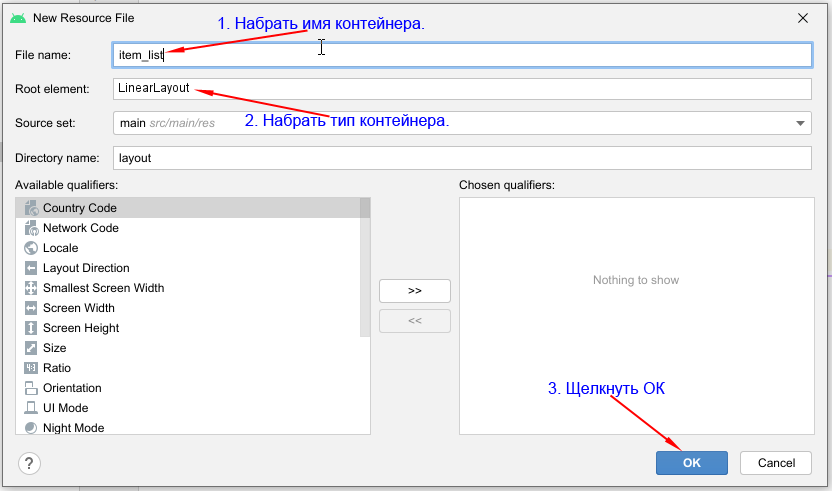
android:layout\_height="wrap\_content" />

</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"

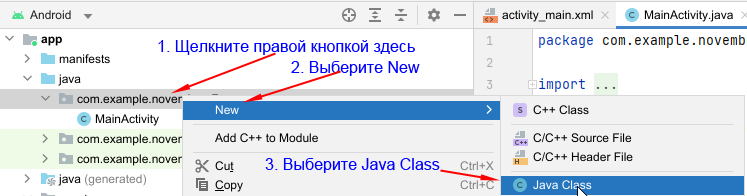
android:layout\_marginStart="20dp"

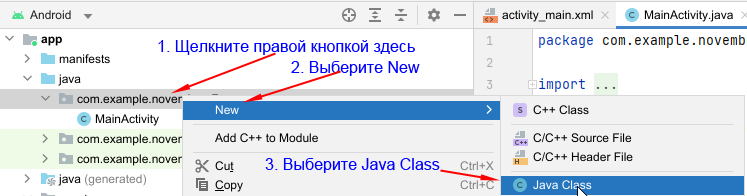
android:layout\_marginTop="20dp"

tools:ignore="HardcodedText" />

</LinearLayout>

1. Добавьте класс для объектов списка





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

public class Cell {

Drawable pic;

String str;

public Cell(Drawable pic, String str) {

this.pic = pic;

this.str = str;

}

public String getStr() {

return str;

}

public void setStr(String str) {

this.str = str;

}

public Drawable getPic() {

return pic;

}

public void setPic(Drawable pic) {

this.pic = pic;

}

}

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<Cell> mylst;

Cell - 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<Cell> mylst) {

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

this.mylst = mylst;

}

1. В заголовок адаптера вставить наследование

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 Cell, 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:

Cell cl = mylst.get(position);  
holder.imgView.setImageDrawable(cl.getPic());  
holder.txtView.setText(cl.getStr());

Здесь Cell – 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<Cell> 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 Cell(getDrawable(R.drawable.cat3), "Кошки"));

mylst.add(new Cell(getDrawable(R.drawable.dog1), "Собаки"));

mylst.add(new Cell(getDrawable(R.drawable.fish), "Рыбы"));

mylst.add(new Cell(getDrawable(R.drawable.insect), "Насекомые"));

}

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 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:

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.

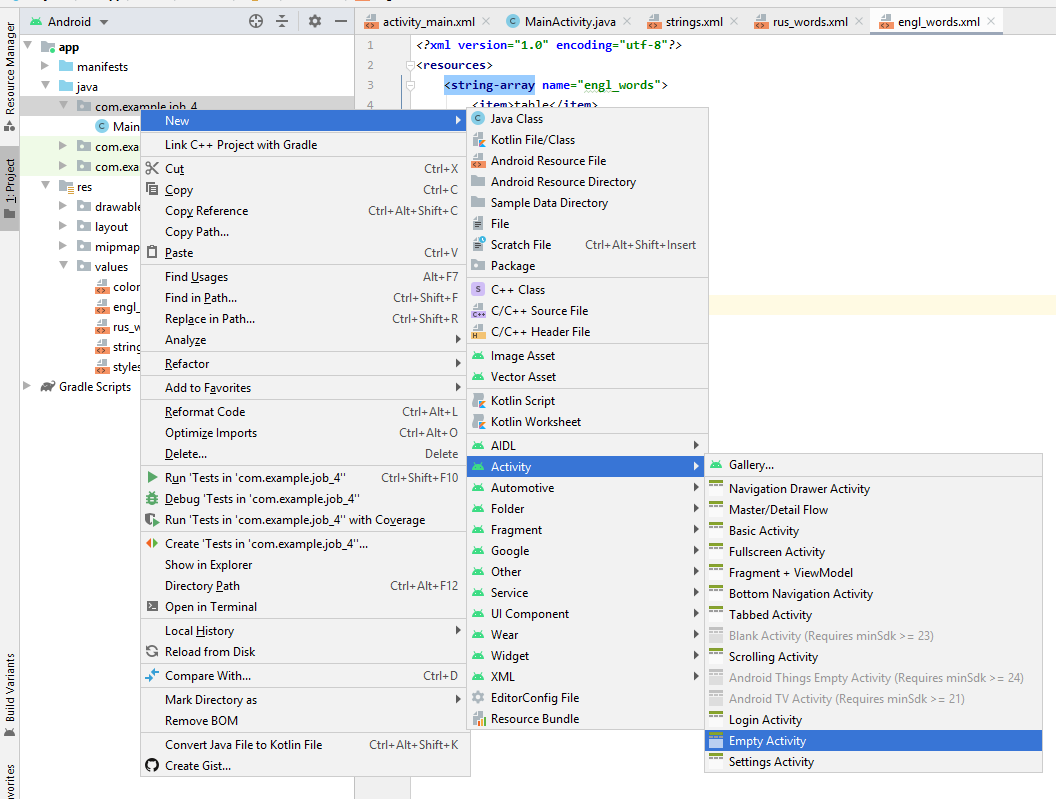
* 1. После строки:

adapter = new ArrayAdapter (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. Добавьте в приложение страницу. Для этого щелкните правой кнопкой в проводнике проекта по строке MainActivity, пройдите указанные меню и щелкните по позиции Empty Activity.



1. В появившемся окне наберите имя новой страницы (SecondActivity) и щелкните по кнопке Finish.
2. Добавление обработчика выбора позиции в списке. Наберите (далее) фрагмент

lstView.setOnItemClickListener(this). Слово this будет подчеркнуто красным. Щелкните по нему правой кнопкой и в появившемся меню Show Context Actions выберите позицию:

Make SecondActivityimplement ‘android.view.View.OnItemClickListener’

1. В классе MainActivity в обработчике выбора позиции оформите команду переключения на вторую страницу:

@Override

public void onItemClick(AdapterView<?> parent, View view, int position,

long id) {

Intent intent = new Intent(MainActivity.this, SecondActivity.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. Проверьте переключение (пока возврата нет).
2. Перейдите в файл activity\_second.xml и выполните следующие действия:
   1. Замените разметку ConstraintLayout на LinearLayout. (Не забудьте вставить ориентацию по вертикали).
   2. Вставьте элементы recyclerview (имя rv) и TextView.

<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=".SecondActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rv"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_weight="1" />

<Button

android:id="@+id/bt"

android:layout\_width="wrap\_content"

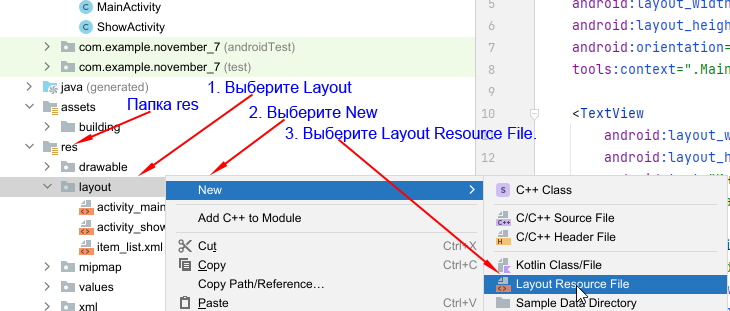
android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

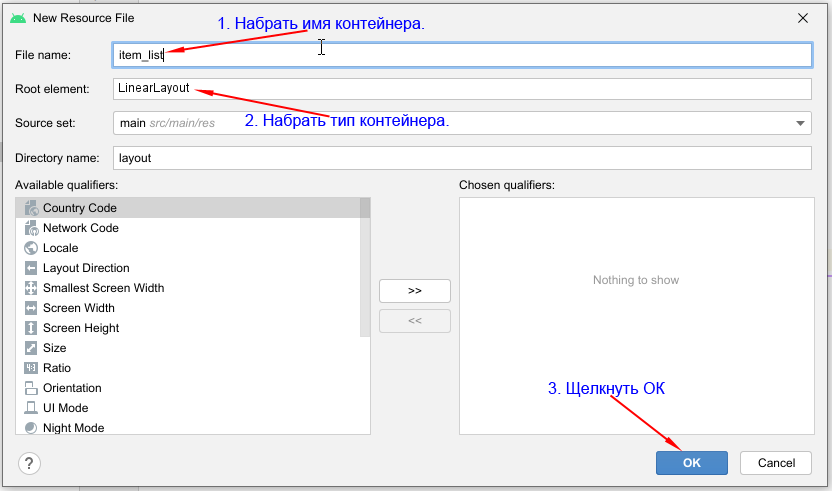
android:text="В начало" />

</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. Вставьте элементы и выполните их настройку

<LinearLayout

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

android:orientation="horizontal"

android:layout\_width="match\_parent"

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

<ImageView

android:id="@+id/imgView"

android:layout\_width="163dp"

android:layout\_height="128dp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="5dp"

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

<TextView

android:id="@+id/txtView"

android:layout\_width="wrap\_content"

android:layout\_height="100dp"

android:layout\_marginStart="20dp"

android:layout\_gravity="center\_vertical"

android:layout\_marginTop="5dp"/>

</LinearLayout>

1. Добавьте класс ArrayAdapter
2. Вставьте члены класса адаптера (LayoutInflater и список ArrayList).

private final LayoutInflater inflater;

private final ArrayList<Cell> mylst;

Cell - 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<Cell> mylst) {

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

this.mylst = mylst;

}

1. В заголовок адаптера вставить наследование

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 Cell, но должен совпадать с порядком их вывода на экран).

ImageView imgView;

TextView txtView;

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*);

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

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:

Cell bl = mylst.get(position);

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

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

Здесь Cell – 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<Cell> 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 Cell(R.drawable.*cat3*, "Кошки"));

mylst.add(new Cell(R.drawable.*dog1*, "Собаки"));

mylst.add(new Cell(R.drawable.*fish*, "Рыбы"));

mylst.add(new Cell(R.drawable.*insect*, "Насекомые"));

}

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

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

recyclerView.setHasFixedSize(true);

LinearLayoutManager layoutManager = new LinearLayoutManager(this);

recyclerView.setLayoutManager(layoutManager);

setInitialData(); *// Вызов метода инициализации*

*// 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 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:

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

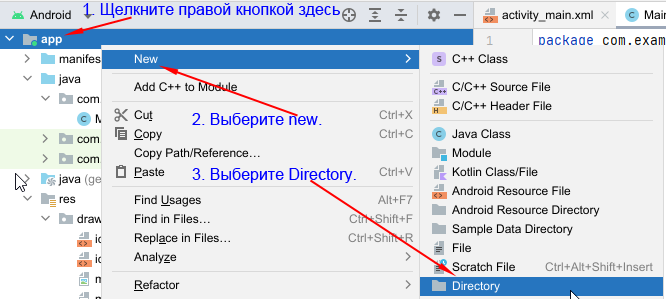
* 1. Заголовок класса MainActivity будет подчеркнут красным. Щелкните по нему правой кнопкой и в появившемся меню Show Context Actions выберите позицию Implement Methods.
  2. В добавленный метод вставьте строку и проверьте работу:

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

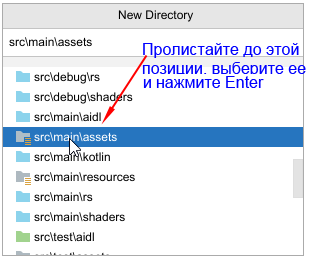
Toast.*LENGTH\_SHORT*).show();

Это требуется для проверки правильности подключения обработчика.

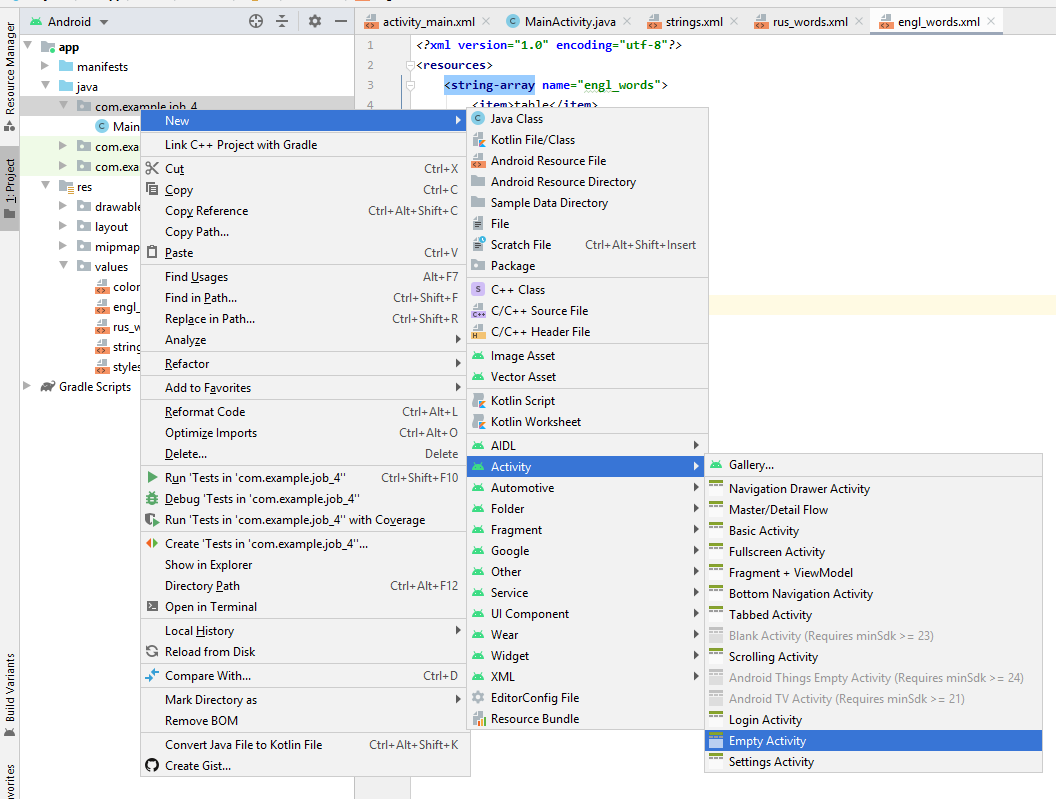
1. Создайте папку assets. Для этого щелкните правой кнопкой по папке app и пройдите по цепочке



* 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 Category и нажмите Enter.
2. В папке (таким же способом, как это делается в Windows) создайте вложенные папки: Cats, Dogs, Fishes и Insects.
3. В новую папку скопируйте файлы из папки Cats, приложенной к заданию.
4. 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 (SecondActivity) 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, SecondActivity.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\_second
   1. Замените контейнер constraints на Layout.
   2. Добавьте ориентацию по вертикали.
   3. Вставьте элементы RecyclerView 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="match\_parent"

android:orientation="vertical"

tools:context=".SecondActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rv"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_weight="1" />

<Button

android:id="@+id/bt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="В начало" />

</LinearLayout>

1. Объявите в классе SecondActivity глобально элементы

Button bt;

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

ArrayAdapter adapter;

String[] files = new String[20];

Drawable[] drw = new Drawable[20];

1. Свяжите элемент Button с разметкой.
2. Слово this подчеркнуто красным. Щелкните по нему правой кнопкой и в появившемся меню Show Context Actions выберите позицию:

Make SecondActivityimplement ‘android.view.View.OnClickListener’

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

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

this.startActivity(intent);

1. Обеспечьте прием выбранной позиции

*// Прием выбранной позиции с предыдущей страницы*

Bundle arguments = getIntent().getExtras();

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

1. Создайте метод считывания списка изображений из папки assets (после добавленного обработчика щелчка):

private void copyAssets() {

AssetManager assetManager = getAssets();

files = null;

try {

files = assetManager.list("Category/Cats/");

} catch (IOException e) {

Log.*e*("tag", "Failed to get asset file list.", e);

}

for (int i = 0; i < files.length; i++) {

try {

*// Объявление входного потока*

InputStream ims = getAssets().open("Category/Cats/" + files[i]);

*// Загрузить изображение в массив типа Drawable*

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

drw[i] = d;

ims.close();

} catch (IOException ignored) {

}

}

}

1. Создайте метод инициализации списка (после метода copyAssets):

public void setInitialData() {

mylst.add(new Cell(drw[0], "Саванна (ашера)"));

mylst.add(new Cell(drw[1], "Эльф"));

mylst.add(new Cell(drw[2], "Персидская"));

mylst.add(new Cell(drw[3], "Русская голубая"));

mylst.add(new Cell(drw[4], "Донской сфинкс"));

mylst.add(new Cell(drw[5], "Сибирская"));

}

1. Вставьте остальные операторы класс SecondActivity

(верхняя строка добавляется через меню Show Context Actions )

@SuppressLint({"MissingInflatedId", "LocalSuppress"})

RecyclerView recyclerView= findViewById(R.id.*rv*);*//*

recyclerView.setHasFixedSize(true);

LinearLayoutManager layoutManager = new LinearLayoutManager(this);

recyclerView.setLayoutManager(layoutManager);

copyAssets();

setInitialData();

*// создаем адаптер*

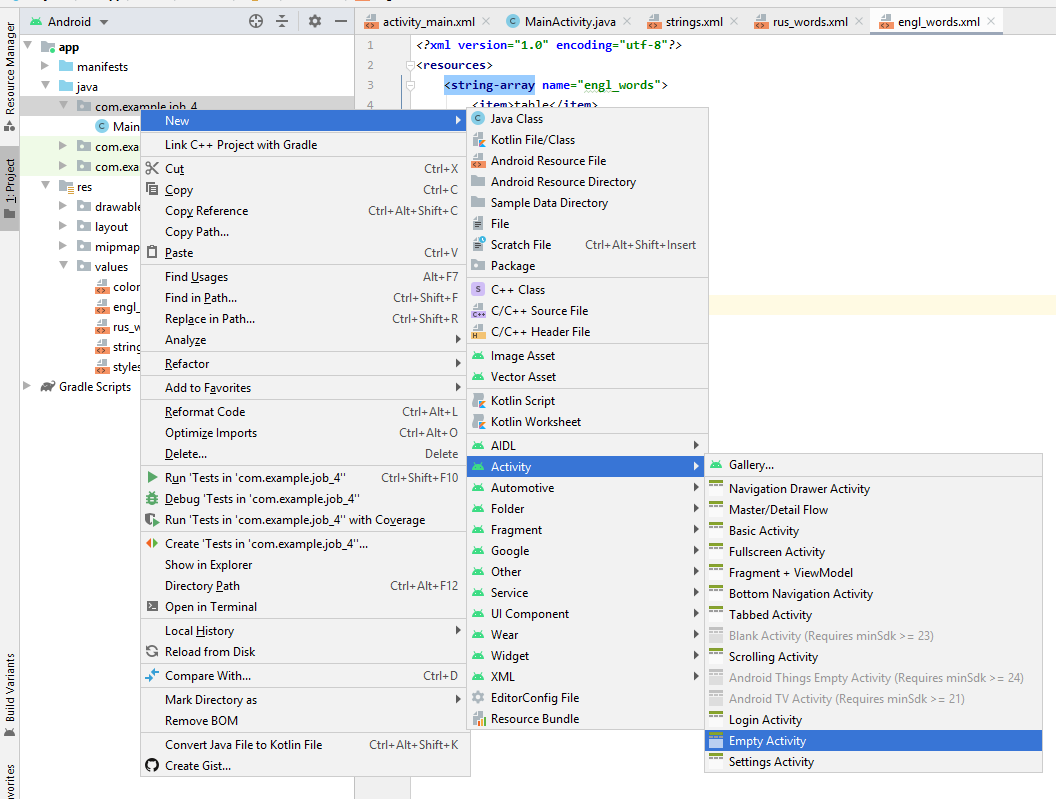
adapter = new ArrayAdapter(this, mylst);

adapter.setClickListener(this);

*// // устанавливаем для списка адаптер*

recyclerView.setAdapter(adapter);

1. Проверьте работу приложения и предъявите ее преподавателю.
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 вызов 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 в классе ShowActivity глобально массив и инициализируйте его:

String[] fname = {"ashera.png", "elf.png", "pers.png", "russblue.png", "sfinx.png", "sibir.png"};

1. Выполните действия в классе ShowActivity:

ImageView img = findViewById(R.id.*imageView*);

@SuppressLint({"MissingInflatedId", "LocalSuppress"}) Button btr = findViewById(R.id.*btr*);

btr.setOnClickListener(this);

*// Прием выбранной позиции с предыдущей страницы*

Bundle arguments = getIntent().getExtras();

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

try

{

*// Получение входного потока*

InputStream ims = getAssets().open("Category/Cats/" + fname[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.

Самостоятельно оформите обработку списка для собак

(можно выбрать другую категорию или создать новую)

1. В папку Dogs вставьте скриншоты собак по породам.
2. Для получения одиночного скриншота откройте список пород собак в Интернете. Выберите понравившуюся породу и запомните ее название.
3. Выполните скриншот, скопируйте его в буфер обмена (Ctrl+C).
4. Откройте новый файл в приложении Paint или Photoshop и вставьте туда скриншот (Ctrl+V).
5. Сохраните файл в папке Dogs обязательно с расширением png, имя любое (латинскими буквами).
6. Повторить для следующей породы (создать не менее 5 файлов).
7. Далее повторить все, как и для кошек, но для другой позиции меню выбора категории. Новых окон создавать не требуется.