**Лабораторная работа № 5**

**Создание многоэкранного приложения**

**Цель лабораторной работы:**

Научиться создавать приложения, состоящие из нескольких активностей, и диалоговые окна, а также познакомиться с элементами тач-интерфейса

**Задачи лабораторной работы:**

* Научиться создавать многоэкранные приложения
* Научиться создавать диалоговые окна и всплывающие подсказки

**5.1 Введение**

Данная лабораторная работа посвящена различным способам разработки многоэкранных приложений. Предлагается разработать три приложения, каждое из которых посвящено отдельному аспекту этой тематики. *По* завершению работы над приложениями студенты смогут разработать собственные программы с использованием изученных компонентов и технологий.

**5.2 Создание многоэкранного приложения со списком**

**1.** Создайте проект **MultiScreen**. В *java*-файле замените Activity на ListActivity. *Класс*  ListActivity разработан таким образом, что на экране есть только прокручиваемый *список* и ему не нужна дополнительная разметка. Поэтому *файл* **activity\_main.xml** можно удалить. Также следует удалить следующую строчку из класса MultiScreen:

setContentView(R.layout.activity\_main);

т.к. *layout*-*файл* мы только что удалили.

**2.** Теперь нам нужен посредник, который свяжет *список* и названия элементов этого списка. Для этого в *Android* есть *интерфейс* Adapter. Нам потребуется наследник этого класса ArrayAdapter:

new ArrayAdapter(Context context, int textViewResourceId, String[] objects)

В качестве аргументов ArrayAdapter потребует текущий *контекст*, *идентификатор* ресурса с разметкой для каждой строки, *массив* строк.

Мы можем указать в качестве текущего контекста ListActivity (можно использовать ключевое *слово* this), готовый системный *идентификатор* ресурса (android.R.layout.simple\_list\_item\_1) и созданный *массив* строк (String[] islands = { "Канары", "Курилы", "Мальдивы", "Филиппины"};). А выглядеть это будет так:

private ArrayAdapter<String> adapter;

adapter = new ArrayAdapter<String>(this,

android.R.layout.simple\_list\_item\_1, islands);

Обратите внимание на строчку android.R.layout.simple\_list\_item\_1. В ней уже содержится необходимая разметка для элементов списка. Если вас не устраивает системная разметка, то можете создать собственную разметку в xml-файле и подключить её.

Осталось только подключить *адаптер*:

setListAdapter(adapter);

**3.** Теперь нам нужно подключить обработку нажатия. Для этого необходимо знать, на какой *пункт* списка осуществляется нажатие. Существует специальный *интерфейс*  OnItemClickListener с методом onItemClick(). Подключаем обработчик:

getListView().setOnItemClickListener(itemListener);

Набираем все в том же onCreate

OnItemClickListener itemListener = new OnItemClickListener(){

}

Если подчеркнётся первое *слово*, то импортируем нужный *класс*. Далее подведём *курсор* ко второму слову new OnItemClickListener. Нам будет предложено добавить метод (Add unimplemented method). Соглашаемся и получаем заготовку:

OnItemClickListener itemListener = new OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,

long arg3) {

// TODO Auto-generated method stub

}

};

Меняем имена переменных *arg* на более привычные и понятные

public void onItemClick(AdapterView<?> parent,

View v, int position, long id);

**4.** Осталось описать, что будет происходить при нажатии на элемент.

*По* нашей задумке каждый элемент будет открывать новое окно с соответствующим содержимым.

Для начала следует создать еще 4 класса: Canari, Curili, Maldivi, Philippini, и 4 xml-оболочки к ним. Можно просто создать и копировать *по* одному файлу в обеих директориях, меняя только название и содержимое.

Например, создадим файлы **Canari.java** и **canari.xml** типа Activity.

**Canari.java**

**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
  
**public class** Canari **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***canari***);  
 }  
}

**canari.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout**

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

**android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
</**LinearLayout**>

Обратите внимание, что как только мы создали ещё один *java*-*файл*, нужно записать его в *файл* манифеста, иначе *приложение* не будет видеть этот новый *класс*. *Файл* **AndtoidManifest.xml** находится сразу под папкой **res**. Добавьте код с именем класса в тегах <activity></activity>, сразу под главной активностью.

<activity android:name="com.mypackage.multiscreen.Canari"

android:label="@string/can" >

</activity>

Строку can нужно создавать в файле **strings.xml**, также как и другие строки.

Перейдем в *файл* **canari.xml** и создадим на экране TextView. Экраны будут отличаться друг от друга картинками. Возьмите 4 любые картинки с вашего компьютера (можете также найти их в интернете) и перетащите из проводника *Windows* в папку **res\drawable**. Теперь вы можете поместить элемент ImageView на экран и, выбрав нужную картинку из ресурсов проекта, подключить ее.

Остальные экраны создаются аналогично.

**5.** Теперь перейдем в главный *класс* и опишем обработку события onItemClick(). Создадим *переключатель*, который зависит от номера элемента.

switch (position) {

case 0:

break;

case 1:

break;

case 2:

break;

case 3:

break;

}

Для запуска нового экрана необходимо создать экземпляр класса Intent и указать *класс*, на который будем переходить (у нас их 4, поэтому для каждого случая выбираем свою). После этого вызывается метод startActivity(), который и запускает новый экран.

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

startActivity(intent);

**6.** Теперь осталось добавить всплывающее окно **Toast**, которое будет показывать, какой элемент мы выбрали. Этот виджет можно импортировать так же, как и предыдущие. Нам потребуется метод makeText(), у которого есть три параметра: *контекст* приложения, текстовое сообщение и продолжительность времени показа уведомления.

Toast.makeText(getApplicationContext(), "Вы выбрали " + parent.getItemAtPosition(position).toString(),Toast.LENGTH\_SHORT).show();

Полные листинги файлов проекта, в которых были сделаны изменения, см. ниже.

package com.mypackage.multiscreen;

import android.os.Bundle;

import android.view.View;

import android.app.ListActivity;

import android.content.Intent;

import android.widget.AdapterView;

import android.widget.AdapterView.OnItemClickListener;

import android.widget.ArrayAdapter;

import android.widget.Toast;

**public class** MainActivity **extends** ListActivity {  
 String[] **islands** = { **"Канары"**, **"Курилы"**, **"Мальдивы"**, **"Филиппины"**};  
 **private** ArrayAdapter<String> **adapter**;  
 **private** Intent **intent**, **intentCa**, **intentCu**, **intentMa**, **intentPh**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 **adapter** = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1***, **islands**);  
  
 **intentCa** = **new** Intent(MainActivity.**this**, Canari.**class**);  
 **intentCu** = **new** Intent(MainActivity.**this**, Curili.**class**);  
 **intentMa** = **new** Intent(MainActivity.**this**, Maldivi.**class**);  
 **intentPh** = **new** Intent(MainActivity.**this**, Philippini.**class**);  
  
 setListAdapter(**adapter**);

AdapterView.OnItemClickListener itemListener = **new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View v, **int** position, **long** id) {  
 **switch** (position) {  
 **case** 0:  
 **intent** = **intentCa**;  
 **break**;  
 **case** 1:  
 **intent** = **intentCu**;  
 **break**;  
 **case** 2:  
 **intent** = **intentMa**;  
 **break**;  
 **case** 3:  
 **intent** = **intentPh**;  
 **break**;  
 }  
  
 startActivity(**intent**);  
 Toast.*makeText*(getApplicationContext(), **"Вы выбрали "** +  
 parent.getItemAtPosition(position).toString(),  
 Toast.***LENGTH\_SHORT***).show();  
  
 }  
 };  
  
 getListView().setOnItemClickListener(itemListener);  
 }  
}

**Листинг 5.1. Файл MultiScreenMainActivity.java**

package com.mypackage.multiscreen;

import android.app.Activity;

import android.os.Bundle;

public class Canari extends Activity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.canari);

}

}

**Листинг 5.2. Файл Canari.java**

<?xml version="1.0" encoding="utf-8"?>

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

android:id="@+id/RelativeLayout1"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:padding="10dip" >

<ImageView

android:id="@+id/imageView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="186dp"

android:src="@drawable/canari" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="80dp"

android:text="@string/enjoy"

android:textAppearance="?android:attr/textAppearanceLarge" />

</RelativeLayout>

**Листинг 5.3. Файл canari.xml**

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="app\_name">Куда бы поехать в отпуск?</string>

<string name="action\_settings">Settings</string>

<string name="enjoy">Enjoy yourself!</string>

<string name="can">Канары</string>

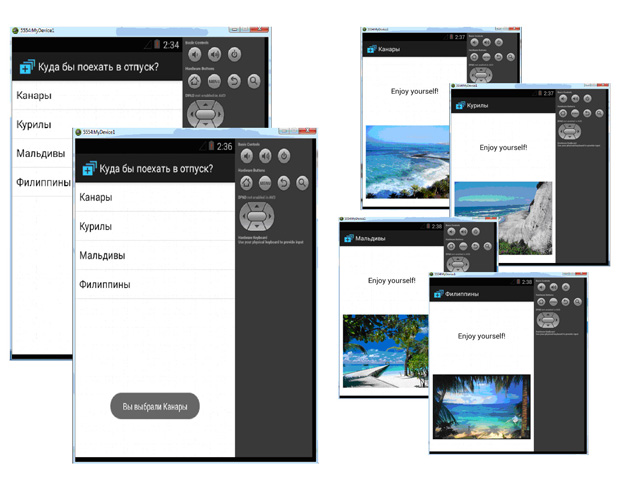
<string name="phil">Филиппины</string>

<string name="cur">Курилы</string>

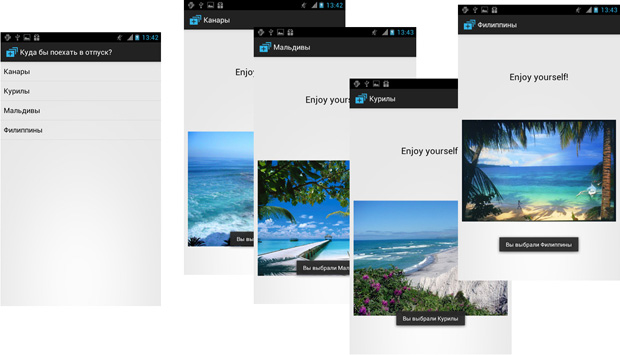
<string name="mal">Мальдивы</string>

</resources>

**Листинг 5.4. Файл strings.xml**

[](http://www.intuit.ru/EDI/26_09_16_2/1474842043-31192/tutorial/1258/objects/8/files/08_01.jpg)

**Рис. 5.1.**Приложение "Куда бы поехать в отпуск?", запущенное на эмуляторе

[](http://www.intuit.ru/EDI/26_09_16_2/1474842043-31192/tutorial/1258/objects/8/files/08_02.jpg)

**Рис. 5.2.**Приложение "Куда бы поехать в отпуск?", запущенное на устройстве

### 5.3 Создание диалогового окна

**1.** Создайте новый проект **Dialog**

**2.** Создайте кнопку на вашей активности и назовите ее "**Выбрать фон**" (для этого нужно в файле **strings** создать строку соответствующего содержания). Присвойте активности и кнопке **id**.

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

android:id="@+id/relativelayout"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" >

<Button

android:id="@+id/background\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="174dp"

android:text="@string/bg"/>

</RelativeLayout>

Листинг 5.5. Код файла activity\_main.xml

**3.** Наше *приложение* меняет фон на выбранный. Значит, нам нужно создать цвета и их названия в файле **strings.xml**. Как вы помните, этот*файл* находится в папке **values**, которая в свою *очередь* находится в папке **res**. Также создадим строку messages, которая нам понадобятся для диалогового окна.

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="app\_name">Dialog</string>

<string name="action\_settings">Settings</string>

<string name="bg">Выбрать фон</string>

<string name="message">Хотите поменять фон?</string>

<color name="redColor">#FF0000</color>

<color name="yellowColor">#FFFF00</color>

<color name="greenColor">#00FF00</color>

<string name="red">Красный</string>

<string name="yellow">Жёлтый</string>

<string name="green">Зелёный</string>

</resources>

Листинг 5.6. Файл strings.xml

**4.** Перейдем в *файл* **MainActivity.java**. Создайте следующие переменные:

private Button bgButton;

public RelativeLayout relativeLayout;

Context context;

Если *компилятор* подчеркивает тип и сообщает об ошибке, например, подчеркивает Context, наведите *курсор* на подчеркнутое *слово*: должно появиться контекстное *меню*, предлагающее варианты, как можно исправить ошибку. Выберете Import 'Context', чтобы импортировать библиотеку.

**5.** Теперь нужно описать, что будет происходить при нажатии на нашу кнопку.

Для начала свяжем объекты из **activity\_main.xml** и переменные в **MainActivity.java** через **id** (в onCreate):

bgButton = (Button)findViewById(R.id.background\_button);

relativeLayout = (RelativeLayout)findViewById(R.id.relativelayout);

Context - это *объект*, который предоставляет *доступ* к базовым функциям приложения.

Добавляем в код

context = MainActivity.this;

**6.** Теперь нужно добавить событие onClick и навесить на кнопку OnClickListener. Добавляем в заголовок класса MainActivity implements OnClickListener. Чтобы связать кнопку и Listener в onCreate пишем

bgButton.setOnClickListener(this);

Создадим теперь событие onClick

@Override

public void onClick(View v){

}

В этом объекте создаем собственно диалог и называем его:

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle(R.string.message);

AlertDialog alert = builder.create();

alert.show();

Мы будем создавать *диалоговое окно*, предоставляющее пользователю выбор из списка. Для этого потребуется ещё одна *переменная*, которая сформирует *список* из имеющихся строк.

final CharSequence[] items = {

getText(R.string.red), getText(R.string.yellow), getText(R.string.green)

};

**7.** Сформируем собственно наш *список* и зададим еще один Listener, который будет менять *цвет фона* на выбранный:

builder.setItems(items, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int item) {

switch (item) {

case 0: { relativeLayout.setBackgroundResource(R.color.redColor);

break;}

case 1: {relativeLayout.setBackgroundResource(R.color.yellowColor);

break;}

case 2: {relativeLayout.setBackgroundResource(R.color.greenColor);

break;}

}

**8.** Осталось добавить в каждый case всплывающие окна Toast, и *приложение* полностью готово!

Toast.makeText(context, R.string.green, Toast.LENGTH\_LONG).show();

package com.mypackage.dialog;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.RelativeLayout;

import android.widget.Toast;

public class MainActivity extends Activity implements OnClickListener {

private Button bgButton;

public RelativeLayout relativeLayout;

Context context;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

bgButton = (Button) findViewById(R.id.background\_button);

bgButton.setOnClickListener(this);

context = MainActivity.this;

relativeLayout = (RelativeLayout)findViewById(R.id.relativelayout);

}

@Override

public void onClick(View v) {

final CharSequence[] items = {getText(R.string.red) , getText(R.string.yellow),getText(R.string.green)

};

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle(R.string.message);

builder.setItems(items, new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int item) {

switch (item) {

case 0: { relativeLayout.setBackgroundResource(R.color.redColor);

Toast.makeText(context, R.string.red, Toast.LENGTH\_LONG).show();

break;}

case 1: {relativeLayout.setBackgroundResource(R.color.yellowColor);

Toast.makeText(context, R.string.yellow, Toast.LENGTH\_LONG).show();

break;}

case 2: {relativeLayout.setBackgroundResource(R.color.greenColor);

Toast.makeText(context, R.string.green, Toast.LENGTH\_LONG).show();

break;}

}

}

});

AlertDialog alert = builder.create();

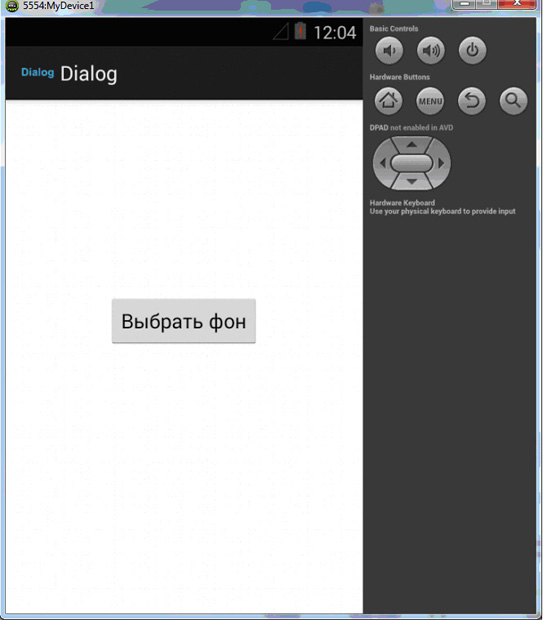
alert.show();

}

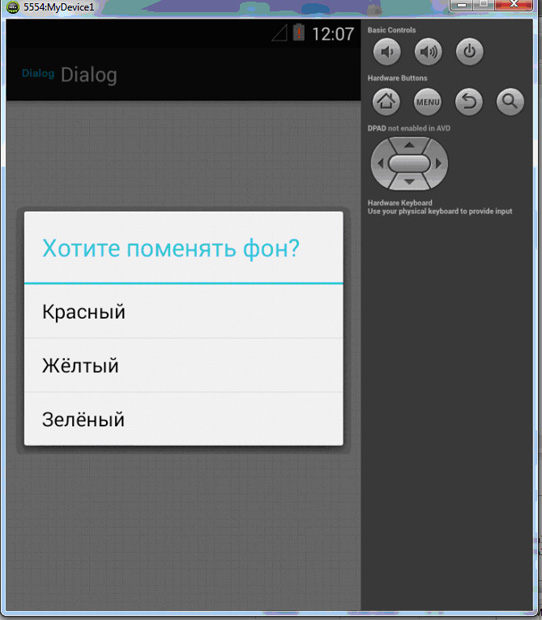
}

Листинг 5.7. Код файла MainActivity.java

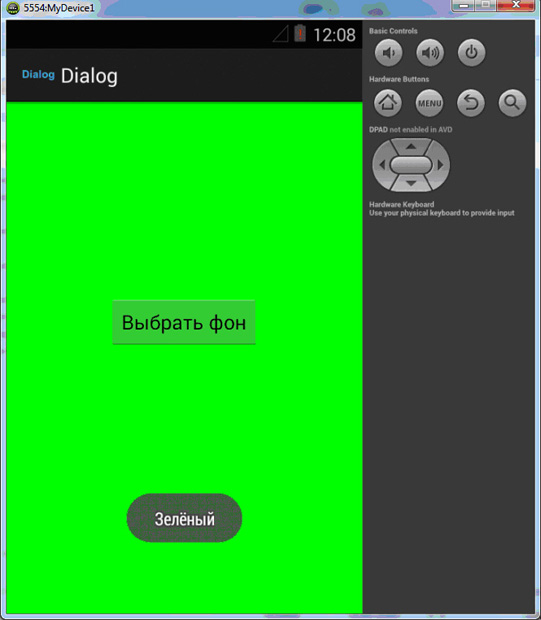
Скриншоты работающего приложения

[](http://www.intuit.ru/EDI/26_09_16_2/1474842043-31192/tutorial/1258/objects/8/files/08_03.jpg)

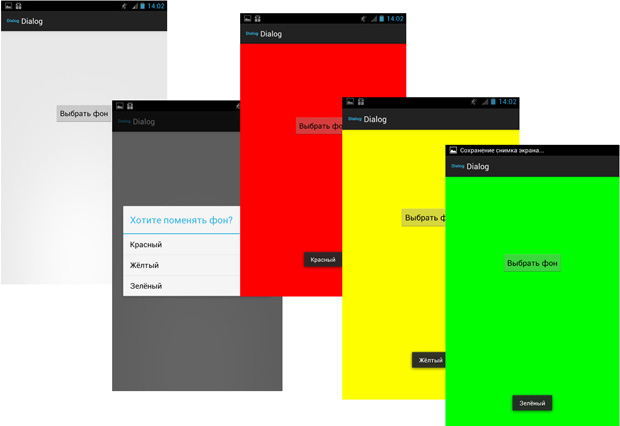
**Рис. 5.3.**Приложение "Dialog", запущенное на эмуляторе

[](http://www.intuit.ru/EDI/26_09_16_2/1474842043-31192/tutorial/1258/objects/8/files/08_04.jpg)

**Рис. 5.4.**Диалоговое окно

[](http://www.intuit.ru/EDI/26_09_16_2/1474842043-31192/tutorial/1258/objects/8/files/08_05.jpg)

**Рис. 5.5.**Выбран зеленый фон

[](http://www.intuit.ru/EDI/26_09_16_2/1474842043-31192/tutorial/1258/objects/8/files/08_06.jpg)

**Рис. 5.6.**Приложение "Dialog", запущенное на устройстве

### Задание для самостоятельного выполнения

Подумайте над собственным приложением, сочетающим различные возможности проектирования многооконных приложений, рассмотренные выше. Создайте прототип этого приложения и настройте его пользовательский *интерфейс*.