

Федеральное агентство образования Российской Федерации
Государственное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет»
Кафедра ИСПИ

Лабораторная работа №2
по дисциплине «»
**«Изучение компонентов пользовательского интерфейса и
обработчиков событий»**

Разработал:

Владимир, 2015

Цель работы

Доработать созданное на предыдущей лабораторной работе приложение используя графические компоненты.

Теоретические сведения

Выполняя предыдущую лабораторную работу вы возможно уже столкнулись с панелью `palette`, которая содержит в себе различные графические элементы пользовательского интерфейса. Далее будут разобраны наиболее популярные из них.

Все элементы собраны в 10 групп: `Form widgets`, `Text fields`, `Layouts`, `Composite`, `Images & Media`, `Time & Date`, `Transitions`, `Advanced`, `Other`, `Custom & Library Views`.

`Form widgets` – содержит активные элементы с которыми пользователь наиболее часто взаимодействует.

`TextView` – служит для отображения текстовой информации.

`Button` – кнопка.

`ToggleButton` – кнопка с индикатором

`CheckBox` – служит для присвоения параметру булевого значения

`RadioButton` – группа таких элементов позволяет выбрать один из нескольких вариантов.

`TextFields` – в этой группе представлены поля для ввода значений. Они работают по одному принципу и различаются только перечнем допустимых для ввода значений.

`Layouts` – данная группа содержит компоненты использующиеся для размещения на них других элементов.

`Composite` – компоненты использующиеся в этой группе служат для представления данных в табличном или списочном виде.

`ListView` – служит для представления данных в списочном виде

`GridView` – служит для представления данных в табличном виде.

`(Horizontal)ScrollView` – служит для представления информации, которая не помещается на экран целиком (по вертикали или горизонтали)

`Image & Media` – элементы данной группы нужны для помощи в отображении на экране изображений и видео

`Time & Date` – в этой группе расположены элементы для отображения даты и времени.

Transitions – элементы данной группы помогают при переключении между изображениями или другими элементами.

Advanced – элементы с различными расширенными функциями.

Other – в этой группе располагаются остальные не классифицированные элементы интерфейса.

Custom & Library Views – в этой группе будут располагаться компоненты интерфейса созданные или модифицированные самим разработчиком, а также элементы из сторонних библиотек.

Порядок выполнения работы

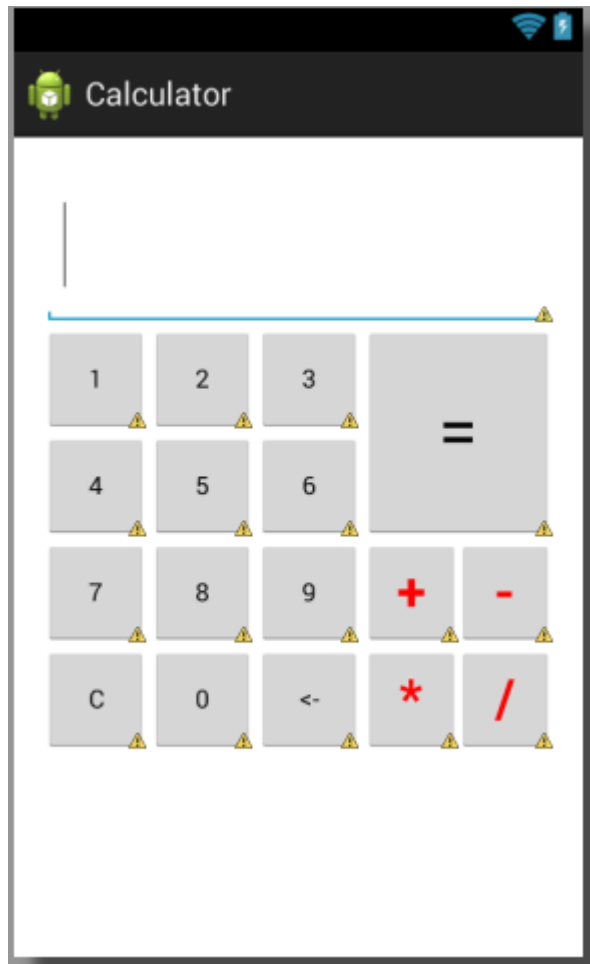
Замечание: для разработки использовалась IDE Eclipse Java EE IDE for Web Developers версии Kepler Service Release 2 с установленным плагином для разработки под Android

1. Создать новый Android Application Project;
2. Добавить на экран элементы пользовательского интерфейса;
3. Написать обработчики для нажатий на кнопки.
4. Написать логику работы приложения
5. Выполнить индивидуальные задания
6. Оформить отчет

Пример

Создадим новый проект. Назовем его Calculator.

Перейдем к файлу activity_main.xml и перейдем в режим графического представления. После чего удалим TextView и вместо него создадим следующую структуру.



```

<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.calculator.MainActivity" >

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:ems="10"
        android:minHeight="90dp"
        android:textSize="40sp" >

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button1"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText1"
        android:layout_below="@+id/editText1"

```

```
android:minHeight="60dp"
android:minWidth="60dp"
android:onClick="buttonClick"
android:text="1" />
```

```
<Button
    android:id="@+id/button2"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button1"
    android:layout_toRightOf="@+id/button1"
    android:minHeight="60dp"
    android:minWidth="60dp"
    android:onClick="buttonClick"
    android:text="2" />
```

```
<Button
    android:id="@+id/button3"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button2"
    android:layout_toRightOf="@+id/button2"
    android:minHeight="60dip"
    android:minWidth="60dp"
    android:onClick="buttonClick"
    android:text="3" />
```

```
<Button
    android:id="@+id/button4"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button1"
    android:layout_below="@+id/button1"
    android:minHeight="60dp"
    android:minWidth="60dp"
    android:onClick="buttonClick"
    android:text="4" />
```

```
<Button
    android:id="@+id/button5"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button4"
    android:layout_toRightOf="@+id/button4"
    android:minHeight="60dp"
    android:minWidth="60dp"
    android:onClick="buttonClick"
    android:text="5" />
```

```
<Button
    android:id="@+id/button6"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/button3"
    android:layout_alignTop="@+id/button5"
    android:minHeight="60dp"
    android:minWidth="60dp"
    android:onClick="buttonClick"
    android:text="6" />
```

```
<Button
```

```
android:id="@+id/button7"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/button4"
android:minHeight="60dp"
android:minWidth="60dp"
android:onClick="buttonClick"
android:text="7" />
```

<Button

```
android:id="@+id/button8"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/button5"
android:layout_alignTop="@+id/button7"
android:minHeight="60dp"
android:minWidth="60dp"
android:onClick="buttonClick"
android:text="8" />
```

<Button

```
android:id="@+id/button9"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/button6"
android:layout_alignTop="@+id/button8"
android:minHeight="60dp"
android:minWidth="60dp"
android:onClick="buttonClick"
android:text="9" />
```

<Button

```
android:id="@+id/buttonResult"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBottom="@+id/button6"
android:layout_alignRight="@+id/editText1"
android:layout_alignTop="@+id/button3"
android:layout_toRightOf="@+id/button3"
android:text=""
android:onClick="buttonClick"
android:textSize="40sp" />
```

<Button

```
android:id="@+id/buttonClear"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/button7"
android:layout_toLeftOf="@+id/button8"
android:minHeight="60dp"
android:minWidth="60dp"
android:onClick="buttonClick"
android:text="C" />
```

<Button

```
android:id="@+id/button0"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/buttonClear"
```

```
android:layout_toLeftOf="@+id/button9"
android:minHeight="60dp"
android:minWidth="60dp"
android:onClick="buttonClick"
android:text="0" />
```

<Button

```
android:id="@+id/buttonBackSpace"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button0"
android:layout_alignBottom="@+id/button0"
android:layout_toRightOf="@+id/button8"
android:minHeight="60dp"
android:minWidth="60dp"
android:onClick="buttonClick"
android:text="&lt;- " />
```

<Button

```
android:id="@+id/ButtonAdd"
style="@style/AppTheme"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/buttonBackSpace"
android:layout_alignLeft="@+id/buttonResult"
android:minHeight="60dp"
android:minWidth="55dp"
android:text="+"
android:textColor="#FF0000"
android:textColorHint="#FFFFFF"
android:textSize="30sp"
android:onClick="buttonClick"
android:textStyle="bold" />
```

<Button

```
android:id="@+id/ButtonSubtract"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/buttonBackSpace"
android:layout_alignParentRight="true"
android:minHeight="60dp"
android:minWidth="55dp"
android:text="-"
android:textColor="#FF0000"
android:textSize="30sp"
android:onClick="buttonClick"
android:textStyle="bold" />
```

<Button

```
android:id="@+id/ButtonDivide"
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/ButtonMultiply"
android:layout_alignBottom="@+id/ButtonMultiply"
android:layout_alignParentRight="true"
android:minHeight="60dp"
android:minWidth="55dp"
android:text="/"
android:textColor="#FF0000"
android:textSize="30sp"
android:onClick="buttonClick"
android:textStyle="bold" />
```

```

<Button
    android:id="@+id/ButtonMultiply"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/ButtonAdd"
    android:layout_alignTop="@+id/buttonBackSpace"
    android:minHeight="60dp"
    android:minWidth="55dp"
    android:text="*"
    android:textColor="#FF0000"
    android:textSize="30sp"
    android:onClick="buttonClick"
    android:textStyle="bold" />

</RelativeLayout>

```

Теперь перейдем в файл MainActivity.java. В нем создадим обработчики для нажатий на кнопки цифр и напомним логику для операции сложения.

```

package com.example.calculator;

import java.util.HashMap;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.TextureView;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SectionIndexer;
import android.widget.TextView;

public class MainActivity extends Activity {
    EditText editText;
    Integer firstValue;
    Integer secondValue;
    String operation;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = (EditText) findViewById(R.id.editText1);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.

```



```

        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    public void buttonClick(View view) {
        Button button = (Button) view;
        switch (button.getText().toString()) {
            case "C":
                firstValue = null;
                secondValue = null;
                operation = null;
                editText.setText("");
                break;
            case "<-":
                break;
            case "+":
                firstValue =
Integer.valueOf(editText.getText().toString());
                operation="+";

                editText.setText(editText.getText().toString()+button.getText());
                break;
            case "-":

                editText.setText(editText.getText().toString()+button.getText());

                break;
            case "*":

                editText.setText(editText.getText().toString()+button.getText());
                break;
            case "/":

                editText.setText(editText.getText().toString()+button.getText());
                break;
            case "=":

                switch (operation) {
                    case "+":

                        editText.setText(String.valueOf((firstValue+secondValue)));
                        break;

                    default:
                        break;
                }

                firstValue = null;
                secondValue = null;
                operation = null;

                break;
            default:
                if(operation != null){
                    if(secondValue!=null)
                        secondValue =
Integer.valueOf((String.valueOf(secondValue)+button.getText().toString()));
                    else
                        secondValue =
Integer.valueOf(button.getText().toString());
                }

```

```

        editText.setText(editText.getText().toString()+button.getText());
        break;
    }
}

```

Как видите, данный код реализует только функционал операции сложения, а также не учитывает ситуации повторного нажатия на кнопку операции и т.д.

Задания

1. Выполнить пример из лабораторной работы.
2. Реализовать функционал остальных трех операций.
3. Выполнить одно из следующих заданий:
 - 3.1. Изменить логику таким образом, чтобы было возможно производить операции над двумя и больше числами.
 - 3.2. Реализуйте логику, которая не дает нажать кнопку операции, если последний введенный символ не число.
 - 3.3. Реализуйте функционал кнопки `backspace`
 - 3.4. Добавьте возможность сохранять промежуточный результат и очищать его. (Кнопки М и МС).
 - 3.5. Добавьте операции взятия корня числа и возведения в квадрат.

Требования к оформлению отчета

Отчет должен содержать:

1. титульный лист;
2. цель работы;
3. задание;
4. описание выполнения примера
5. описание выполнения индивидуального задания со скриншотами.
6. выводы.

Контрольные вопросы

1. Каково одно из главных преимуществ платформы Android?
2. Какие инструменты входят в состав Android SDK?
3. Что такое эмулятор Android?

Дополнительная литература

1. <http://www.startandroid.ru/ru/uroki/vse-uroki-spiskom/>
2. <http://developer.alexanderklimov.ru/android/>
- 3.