ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Лабораторная работа №1

«Калькулятор»

Выполнил: студент группы ИП-712

Алексеев С.В.

Работу принял: ассистент кафедры ПМиК

Павлова У.В.

Новосибирск 2020

Оглавление

[Теория 2](#_Toc40180622)

[Задание 7](#_Toc40180623)

[Листинг 7](#_Toc40180624)

[Скрины 15](#_Toc40180625)

Теория:

Единицы измерен Общая информация

Кнопка - один из самых распространенных элементов управления в программировании. Наследуется от Textview и является базовым

классом для класса СompoundButton. От класса CompoundButton в свою очередь наследуются такие элементы

как CheckBox, ToggleButton и RadioButton. В Android для кнопки используется классandroid.widget.Button. На кнопке располагается

текст и на кнопку нужно нажать, чтобы получить результат. Альтернативой ему может служить

элемент ImageButton (android.widget.ImageButton), у которого вместо текста используется изображение.

Элемент Button находится в папке Form Widgets:

Если вы растягиваете кнопку по всей ширине экрана (android:layout\_width="match\_parent"), то дополнительно рекомендую использовать

атрибут android:layout\_margin (или родственные ему layout\_marginRight и layout\_marginLeft) для создания отступов от краев экрана (веб-

мастера знакомы с этими терминами).

Так как является наследником Textview, то использует многие знакомые нам атрибуты: textColor, textSize и др.

Три способа обработки событий нажатий на кнопку

Если вы разместили на экране кнопку и будете нажимать на неё, то ничего не произойдёт. Необходимо написать код, который будет

выполняться при нажатии. Существует несколько способов обработки нажатий на кнопку.

Первый способ - атрибут onClick

Относительно новый способ, специально разработанный для Android - использовать атрибут onClick (на панели свойств отображается

как On Click):

android:onClick="onMyButtonClick"

Имя для события можно выбрать произвольное, но лучше не выпендриваться. Далее вам нужно прописать в классе активности

придуманное вами имя метода, который будет обрабатывать нажатие. Метод должен быть открытым (public) с одним параметром,

использующим объект View. Вам нужно выучить пять слов для создания метода, а сам метод поместить в класс (если вы ещё путаетесь в

структуре Java-кода, то вставьте метод перед последней фигурной скобкой):

public void onMyButtonClick(View view)

{

// выводим сообщение

Toast.makeText(this, "Зачем вы нажали?", Toast.LENGTH\_SHORT).show();

}

Когда пользователь нажимает на кнопку, то вызывается метод onMyButtonClick(), который в свою очередь генерирует всплывающее

сообщение.

Обратите внимание, что при подобном подходе вам не придётся даже объявлять кнопку через конструкцию

(Button)findViewById(R.id.button1), так как Android сама поймёт, что к чему. Данный способ применим не только к кнопке, но и к другим

элементам. Позволяет сократить количество строк кода.

Второй способ - метод setOnClickListener()

Более традиционный способ в Java - через метод setOnClickListener(), который прослушивает нажатия на кнопку. Так как для

начинающего программиста код может показаться сложным, то рекомендуется использовать подсказки Eclipse. Вот как это будет

выглядеть. Предположим, у вас на экране уже есть кнопка Button1. В коде вы объявляете её обычным способом:

Button button1 = (Button)findViewById(R.id.button1);

Следующий шаг - написание метода для нажатия. Напечатайте имя элемента и поставьте точку button1. - среда разработки покажет вам

список доступных выражений для продолжения кода. Вы можете вручную просмотреть и выбрать нужный вариант, а можно продолжать

набирать символы, чтобы ускорить процесс. Так как с нажатиями кнопок вам часто придётся работать, то запомните название его метода

(хотя бы первые несколько символов) - набрав шесть символов (setonc), вы увидите один оставшийся вариант, дальше можно сразу

нажать клавишу Enter, не набирая оставшиеся символы. У вас появится строка такого вида:

button1.setOnClickListener(l)

Символ l будет выделен, а также будет видна подсказка OnClickListener l. Пока символ выделен, начинайте набиратьnew OnClickListener.

Здесь также не обязательно набирать имя полностью. Наберите слово oncl и нажмите комбинацию клавиш Ctrl+Space для вывода

вариантов продолжения. Eclipse предложит вам единственный вариант, поэтому сразу нажимайте Enter и вы получите готовую заготовку

для обработки нажатия кнопки:

button1.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

}

})

Вас могут испугать три красные метки об ошибках. Не пугайтесь, поставьте в самом конце заготовки символ точки с запятой ; после

последней круглой скобки, и первая ошибка исчезнет. Далее подведите курсор мыши к словуOnClickListener, которая подчёркнута

волнистой красной чертой. Появятся несколько вариантов для исправления ошибки. Выберите вариант Import 'OnClickListener'

(android.view.View) и исчезнут оставшиеся две метки. Теперь у вас есть рабочая заготовка и сразу после строки с комментарием TODO

Auto-generated method stub вы можете писать свой код. Рекомендую потренироваться и набить руку в создании заготовки. Это не так

сложно и с практикой навык закрепится автоматически.

Как вариант, можно вынести код для OnClickListener в отдельное место, это удобно, когда кнопок на экране несколько и такой подход

позволит упорядочить код. Удалите предыдущий пример и начните писать код заново. Принцип такой же, немного меняется порядок. В

предыдущем примере мы сразу прописали в методе setOnClickListener new OnClickListener... с методом onClick. Можно сначала отдельно

объявить отдельную переменную:

OnClickListener MyButtonClick = new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

}

};

Во время набора активно используйте подсказки через Ctrl+Space. Набрали несколько символов у первого слова и нажимайте эту

комбинацию, набрали после слова new несколько символов и снова нажимайте указанную комбинацию - заготовка будет создана за

несколько секунд, а вы избежите возможных опечаток.

У нас есть готовая переменная, и теперь, когда вы будете набирать код button1.setOnClickListener, то опять появится выделенный

символ l, но уже с подсказкой, что можно выбрать готовую переменную - выделите её и нажите Enter или вручную пишите

самостоятельно.

Для новичка описание может показаться сумбурным и не понятным, но лучше самостоятельно проделать эти операции и понять

механизм.

Третий способ - интерфейс OnClickListener

Третий способ является родственным второму способу и также является традиционным для Java. Кнопка присваивает себе обработчика с

помощью метода setOnClickListener (View.OnClickListener l), т.е. подойдет любой объект с интерфейсом View.OnClickListener. Мы можем

указать, что наш класс Activity будет использовать интерфейс View.OnClickListener.

Опять стираем код от предыдущего примера. Далее после слов extends Activity дописываем слова implements OnClickListener. Если вы

написали код через подсказку, то строка import android.view.View.OnClickListener;вставится в секцию импорта автоматически, в

противном случае пропишите самостоятельно.

Название вашего класса будет подчёркнуто волнистой красной чертой, подведите курсор мыши к слову и выберите вариант Add

unimplemented methods. В коде появится заготовка для нажатия кнопки.

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

}

Метод будет реализован не в отдельном объекте-обработчике, а в Activity, который и будет выступать обработчиком. В методе onCreate

присвоим обработчик кнопке. Это будет объект this, т.е. текущий объект нашей активности.

button1.setOnClickListener(this);

.

Плодитесь и размножайтесь - это про кошек, а не про

кнопки

Когда у вас одна кнопка в окне, то у вас будет один метод, две кнопки - два метода и так далее. Если у вас несколько кнопок, то не

обязательно для каждой прописывать свой метод, можно обойтись и одним, а уже в самом методе разделять код по идентификатору

кнопки. Если вы посмотрите на код в предыдущих примерах, то увидите, что в методе присутствует параметр View, который и позволяет

определить, для какой кнопки предназначен кусок кода:

public void onMyButtonClick(View view)

{

switch(view.getId()) {

case R.id.btn1: // идентификатор "@+id/btn1"

ShowAlertDialog();

break;

...

}

Предположим, у вас есть три кнопки:

import android.view.View.OnClickListener;

public class ButtonDemoActivity extends Activity implements OnClickListener...

final Button button1 = (Button)findViewById(R.id.button1);

final Button button2 = (Button)findViewById(R.id.button2);

final Button button3 = (Button)findViewById(R.id.button3);

// устанавливаем один обработчик для всех кнопок

button1.setOnClickListener(this);

button2.setOnClickListener(this);

button3.setOnClickListener(this);

// анализируем, какая кнопка была нажата. Всего один метод для всех кнопок

@Override

public void onClick(View v){

switch (v.getId()) {

case R.id.button1: editText.setText("Нажата кнопка Button1); break;

case R.id.button2: editText.setText("Нажата кнопка Button2); break;

case R.id.button3: editText.setText("Нажата кнопка Button3); break;

}

}

Как видите, мы сократили количество кода. Теперь у нас один обработчик onClick, в котором прописаны действия для трёх кнопок.

Сделать кнопку недоступной

Иногда нужно сделать кнопку недоступной и активировать её при определённых условиях. Через XML нельзя сделать кнопку недоступной

(нет подходящего атрибута). Это можно сделать программно через метод setEnabled():

myButton.setEnabled(false);

Как альтернативу можете рассмотреть атрибут android:clickable, который позволит кнопке не реагировать на касания, но при этом вид

кнопки останется обычным.

Задание: Реализуйте простейший Калькулятор. Имеется набор кнопок (цифр), (арифм. операций) и циферблат. Калькулятор позволяет вычислять (+,-,\*,/) Предусмотреть обработку ситуации деления на ноль.

Листинг:

MainActivity.java

package com.example.lab01\_1;  
  
import android.os.Bundle;  
  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.google.android.material.snackbar.Snackbar;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.appcompat.widget.Toolbar;  
  
import android.view.View;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.Button;  
import android.widget.TextView;  
  
import java.text.ParseException;  
  
public class MainActivity extends AppCompatActivity {  
 private double result = 0;  
 private double a = 0;  
 private double b = 0;  
 private int operationNum = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 FloatingActionButton fab = findViewById(R.id.*fab*);  
 fab.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Snackbar.*make*(view, "Replace with your own action", Snackbar.*LENGTH\_LONG*)  
 .setAction("Action", null).show();  
 }  
 });  
  
 add7();  
 add8();  
 add9();  
 clear();  
 addDiv();  
 equals();  
 add4();  
 add5();  
 add6();  
 add1();  
 add2();  
 add3();  
 add0();  
 mul();  
 plus();  
 minus();  
  
  
  
  
 }  
  
  
 private void add7() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button7*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "7");  
 }  
 });  
 }  
  
 private void add8() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button8*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "8");  
 }  
 });  
 }  
  
 private void add9() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button9*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "9");  
 }  
 });  
 }  
  
 private void clear() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*buttonClear*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText("");  
 }  
 });  
 }  
  
  
  
  
 private void add4() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button4*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "4");  
 }  
 });  
 }  
  
 private void add5() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button5*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "5");  
 }  
 });  
 }  
  
 private void add6() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button6*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "6");  
 }  
 });  
 }  
 private void add1() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button1*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "1");  
 }  
 });  
 }  
  
 private void add2() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button2*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "2");  
 }  
 });  
 }  
  
 private void add3() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button3*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "3");  
 }  
 });  
 }  
 private void add0() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*button0*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 tw.setText(tw.getText() + "0");  
 }  
 });  
 }  
 private void mul() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*buttonMul*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 a = Double.*parseDouble*(tw.getText().toString());  
 }  
 catch( NumberFormatException e){  
 tw.setText("not a number");  
 }  
 tw.setText("");  
 operationNum = 1;  
 }  
 });  
 }  
 private void plus() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*buttonPlus*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 a = Double.*parseDouble*(tw.getText().toString());  
 }  
 catch(NumberFormatException e){  
 tw.setText("not a number");  
 }  
 tw.setText("");  
 operationNum = 2;  
 }  
 });  
 }  
 private void minus() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*buttonMinus*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 a = Double.*parseDouble*(tw.getText().toString());  
 }  
 catch(NumberFormatException e){  
 tw.setText("not a number");  
 }  
 tw.setText("");  
 operationNum = 3;  
 }  
 });  
 }  
 private void addDiv() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*buttonDiv*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 a = Double.*parseDouble*(tw.getText().toString());  
 }  
 catch(NumberFormatException e){  
 tw.setText("not a number");  
 }  
  
 tw.setText("");  
 operationNum = 0;  
 }  
 });  
 }  
 private void equals() {  
 final TextView tw = (TextView) findViewById(R.id.*textView2*);  
 Button ctb = (Button) findViewById(R.id.*buttonEquals*);  
 ctb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try {  
 b = Double.*parseDouble*(tw.getText().toString());  
 }  
 catch(NumberFormatException e){  
 tw.setText("not a number");  
 }  
  
 try{  
// if (b == 0.0 && operationNum == 0) {  
// tw.setText("can't divide by zero");  
// } else  
 if(operationNum == 0){  
 result = a / b;  
 tw.setText(String.*valueOf*(result));  
 }  
 else if(operationNum == 1){  
 result = a \* b;  
 tw.setText(String.*valueOf*(result));  
 }  
 else if(operationNum == 2){  
 result = a + b;  
 tw.setText(String.*valueOf*(result));  
 }  
 else if(operationNum == 3){  
 result = a - b;  
 tw.setText(String.*valueOf*(result));  
 }  
  
 }catch(ArithmeticException e) {  
 tw.setText("wrong math");  
 }  
  
  
 }  
 });  
 }  
  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // Inflate the menu; this adds items to the action bar if it is present.  
 getMenuInflater().inflate(R.menu.*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();  
  
 //noinspection SimplifiableIfStatement  
 if (id == R.id.*action\_settings*) {  
 return true;  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
}

content\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">  
  
 <LinearLayout  
 android:layout\_width="454dp"  
 android:layout\_height="758dp"  
 android:orientation="vertical"  
 tools:layout\_editor\_absoluteX="1dp"  
 tools:layout\_editor\_absoluteY="1dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="95dp"  
 android:orientation="vertical"></LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="114dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text=""  
  
 />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="86dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/button7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="7"  
 android:onClick="pressed7"  
 android:clickable="true"  
  
 />  
  
  
 <Button  
 android:id="@+id/button8"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="86dp"  
 android:layout\_weight="1"  
 android:text="8" />  
  
 <Button  
 android:id="@+id/button9"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="86dp"  
 android:layout\_weight="1"  
 android:text="9" />  
  
 <Button  
 android:id="@+id/buttonDiv"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="86dp"  
 android:layout\_weight="1"  
 android:text="/" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="103dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/button4"  
 android:layout\_width="77dp"  
 android:layout\_height="93dp"  
 android:layout\_weight="1"  
 android:text="4" />  
  
 <Button  
 android:id="@+id/button5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="88dp"  
 android:layout\_weight="1"  
 android:text="5" />  
  
 <Button  
 android:id="@+id/button6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="88dp"  
 android:layout\_weight="1"  
 android:text="6" />  
  
 <Button  
 android:id="@+id/buttonMul"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="89dp"  
 android:layout\_weight="1"  
 android:text="\*" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="82dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/button1"  
 android:layout\_width="77dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="1" />  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="82dp"  
 android:layout\_weight="1"  
 android:text="2" />  
  
 <Button  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="82dp"  
 android:layout\_weight="1"  
 android:text="3" />  
  
 <Button  
 android:id="@+id/buttonMinus"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="82dp"  
 android:layout\_weight="1"  
 android:text="-" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="90dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/buttonClear"  
 android:layout\_width="78dp"  
 android:layout\_height="90dp"  
 android:layout\_weight="1"  
 android:text="Clear" />  
  
 <Button  
 android:id="@+id/button0"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="90dp"  
 android:layout\_weight="1"  
 android:text="0" />  
  
 <Button  
 android:id="@+id/buttonEquals"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="90dp"  
 android:layout\_weight="1"  
 android:text="=" />  
  
 <Button  
 android:id="@+id/buttonPlus"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="90dp"  
 android:layout\_weight="1"  
 android:text="+" />  
 </LinearLayout>  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

Скрины:

