пишем приложение - калькулятор

Попробуем написать простейший калькулятор, который берет два числа и проводит с ними операции сложения, вычитания, умножения или деления. Результат отображает в виде полного выражения.

Создадим проект:

**Project name**: P0191\_SimpleCalculator  
**Build Target**: Android 2.3.3  
**Application name**: SimpleCalculator  
**Package name**: ru.startandroid.develop.simplecalculator  
**Create Activity**: MainActivity

Откроем **main.xml** и нарисуем экран:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:orientation="vertical"      android:layout\_width="fill\_parent"      android:layout\_height="fill\_parent">      <LinearLayout          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:id="@+id/linearLayout1"          android:layout\_marginLeft="10pt"          android:layout\_marginRight="10pt"          android:layout\_marginTop="3pt">          <EditText              android:layout\_weight="1"              android:layout\_height="wrap\_content"              android:layout\_marginRight="5pt"              android:id="@+id/etNum1"              android:layout\_width="match\_parent"              android:inputType="numberDecimal">          </EditText>          <EditText              android:layout\_height="wrap\_content"              android:layout\_weight="1"              android:layout\_marginLeft="5pt"              android:id="@+id/etNum2"              android:layout\_width="match\_parent"              android:inputType="numberDecimal">          </EditText>      </LinearLayout>      <LinearLayout          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:id="@+id/linearLayout2"          android:layout\_marginTop="3pt"          android:layout\_marginLeft="5pt"          android:layout\_marginRight="5pt">          <Button              android:layout\_height="wrap\_content"              android:layout\_width="match\_parent"              android:layout\_weight="1"              android:text="+"              android:textSize="8pt"              android:id="@+id/btnAdd">          </Button>          <Button              android:layout\_height="wrap\_content"              android:layout\_width="match\_parent"              android:layout\_weight="1"              android:text="-"              android:textSize="8pt"              android:id="@+id/btnSub">          </Button>          <Button              android:layout\_height="wrap\_content"              android:layout\_width="match\_parent"              android:layout\_weight="1"              android:text="\*"              android:textSize="8pt"              android:id="@+id/btnMult">          </Button>          <Button              android:layout\_height="wrap\_content"              android:layout\_width="match\_parent"              android:layout\_weight="1"              android:text="/"              android:textSize="8pt"              android:id="@+id/btnDiv">          </Button>      </LinearLayout>      <TextView          android:layout\_height="wrap\_content"          android:layout\_width="match\_parent"          android:layout\_marginLeft="5pt"          android:layout\_marginRight="5pt"          android:textSize="12pt"          android:layout\_marginTop="3pt"          android:id="@+id/tvResult"          android:gravity="center\_horizontal">      </TextView>  </LinearLayout> |

Тут есть **два поля ввода**, **4 кнопки** и **текстовое поле для вывода**. Обратите внимание на атрибут **inputType**для**EditText**. Он задает**тип содержимого**. Я указал **numberDecimal** – т.е. в поле получится ввести **только цифры и запятую**, буквы он не пропустит. Это удобно, не надо самому кодить различные проверки.

Для **TextView**указан атрибут **gravity**. Он указывает, как будет**расположен текст** в TextView. Не путайте с layout\_gravity, который отвечает за размещение TextView в ViewGroup.

Теперь нам надо читать содержимое полей, определять какую кнопку нажали и выводить нужный результат. Открываем **MainActivity.java** и пишем код

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83 | public class MainActivity extends Activity implements OnClickListener {      EditText etNum1;    EditText etNum2;      Button btnAdd;    Button btnSub;    Button btnMult;    Button btnDiv;      TextView tvResult;      String oper = "";      /\*\* Called when the activity is first created. \*/    @Override    public void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.main);        // находим элементы      etNum1 = (EditText) findViewById(R.id.etNum1);      etNum2 = (EditText) findViewById(R.id.etNum2);        btnAdd = (Button) findViewById(R.id.btnAdd);      btnSub = (Button) findViewById(R.id.btnSub);      btnMult = (Button) findViewById(R.id.btnMult);      btnDiv = (Button) findViewById(R.id.btnDiv);        tvResult = (TextView) findViewById(R.id.tvResult);        // прописываем обработчик      btnAdd.setOnClickListener(this);      btnSub.setOnClickListener(this);      btnMult.setOnClickListener(this);      btnDiv.setOnClickListener(this);      }      @Override    public void onClick(View v) {      // TODO Auto-generated method stub      float num1 = 0;      float num2 = 0;      float result = 0;        // Проверяем поля на пустоту      if (TextUtils.isEmpty(etNum1.getText().toString())          || TextUtils.isEmpty(etNum2.getText().toString())) {        return;      }        // читаем EditText и заполняем переменные числами      num1 = Float.parseFloat(etNum1.getText().toString());      num2 = Float.parseFloat(etNum2.getText().toString());        // определяем нажатую кнопку и выполняем соответствующую операцию      // в oper пишем операцию, потом будем использовать в выводе      switch (v.getId()) {      case R.id.btnAdd:        oper = "+";        result = num1 + num2;        break;      case R.id.btnSub:        oper = "-";        result = num1 - num2;        break;      case R.id.btnMult:        oper = "\*";        result = num1 \* num2;        break;      case R.id.btnDiv:        oper = "/";        result = num1 / num2;        break;      default:        break;      }        // формируем строку вывода      tvResult.setText(num1 + " " + oper + " " + num2 + " = " + result);    }  } |

Думаю, все понятно по каментам. Читаем значения, определяем кнопку, выполняем операцию и выводим в текстовое поле. Обработчиком нажатий на кнопки выступает **Activity**.

Все сохраним и запустим.



Давайте для большего функционала сделаем **меню** с пунктами**очистки полей** и **выхода из приложения**. Пункты будут называться **Reset**и **Quit**.

Добавим**две константы** – это будут**ID** пунктов меню.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class MainActivity extends Activity implements OnClickListener {    final int MENU\_RESET\_ID = 1;  final int MENU\_QUIT\_ID = 2;    EditText etNum1; |

(добавляете только строки 3 и 4)

И напишем код **создания**и **обработки**меню:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | // создание меню  @Override  public boolean onCreateOptionsMenu(Menu menu) {      menu.add(0, MENU\_RESET\_ID, 0, "Reset");      menu.add(0, MENU\_QUIT\_ID, 0, "Quit");      return super.onCreateOptionsMenu(menu);  }    // обработка нажатий на пункты меню  @Override  public boolean onOptionsItemSelected(MenuItem item) {      switch (item.getItemId()) {          case MENU\_RESET\_ID:              // очищаем поля              etNum1.setText("");              etNum2.setText("");              tvResult.setText("");              break;          case MENU\_QUIT\_ID:              // выход из приложения              finish();              break;      }      return super.onOptionsItemSelected(item);  } |

Сохраним все, запустим. Появилось два пункта меню:  
**Reset**– очищает все поля  
**Quit**– закрывает приложение

В качестве самостоятельной работы вы можете реализовать **проверку деления на ноль**. И выводить какое-нить сообщение с помощью Toast или прямо в поле результата.