**Использование диалогов**

**Простой диалог**

Сначала добавим в класс StartActivity.java следующие константы:

*// код результата проверки*

**private** static final int ALERT\_NONE=0;*// параметры введены верно*

**private** static final int ALERT\_COLUMNS=1;*// некорректное число столбцов*

**private** static final int ALERT\_ROWS=2;*// некорректное число строк*

**private** static final int ALERT\_CELLS=3;*// некорректное начальное число клеток*

*// границы допустимых значений числа столбцов*

**private** static final int COLUMNS\_MIN=5;

**private** static final int COLUMNS\_MAX=25;

*// границы допустимых значений числа строк*

**private** static final int ROWS\_MIN=5;

**private** static final int ROWS\_MAX=35;

Теперь добавляем метод, осуществляющий проверку введенных параметров:

**private** int checkInputParameters(int cols, int rows, int cells)

{

if(cols< COLUMNS\_MIN|| cols> COLUMNS\_MAX)

{

return ALERT\_COLUMNS;

}

if(rows< ROWS\_MIN|| rows> ROWS\_MAX)

{

return ALERT\_ROWS;

}

if(cells> rows\* cols)

{

return ALERT\_CELLS;

}

return ALERT\_NONE;

}

Метод onClick теперь должен работать немного по-другому: сначала проверять введенные параметры, и, если проверка прошла (ALERT\_NONE), открывать окно RunActivity, иначе же показывать предупреждение. Итак, onClick будет выглядеть так:

**public** void onClick(View v)

{

EditText rowsEditor=(EditText)findViewById(R.id.rows\_count);

EditText colsEditor=(EditText)findViewById(R.id.columns\_count);

EditText cellsEditor=(EditText)findViewById(R.id.cells\_count);

int cols= Integer.parseInt(colsEditor.getText().toString());

int rows= Integer.parseInt(rowsEditor.getText().toString());

int cells= Integer.parseInt(cellsEditor.getText().toString());

int alertCode= checkInputParameters(cols, rows, cells);

if(alertCode!= ALERT\_NONE)

{

showDialog(alertCode);

return;

}

Intent intent=**new** Intent();

intent.setClass(this, RunActivity.**class**);

intent.putExtra(RunActivity.EXT\_COLS, cols);

intent.putExtra(RunActivity.EXT\_ROWS, rows);

intent.putExtra(RunActivity.EXT\_CELLS, cells);

startActivity(intent);

finish();

}

Метод showDialog(id) класса Activity пытается открыть диалог. При этом вызывается метод onCreateDialog, возвращающий объект класса Dialog. Так что нужно перекрыть метод onCreateDialog и сконструировать тот диалог, который нам нужен. Ну у нас тут всё просто, диалоги будут отличаться только выводимым сообщением, так что можно сделать следующее:

@Override

**protected** Dialog onCreateDialog(int id)

{

DialogInterface.OnClickListener doNothing=**new** DialogInterface.OnClickListener()

{

**public** void onClick(DialogInterface dialog, int whichButton)

{

}

};

int alertMessage;

switch(id)

{

case ALERT\_COLUMNS:

alertMessage= R.string.alert\_columns;

break;

case ALERT\_ROWS:

alertMessage= R.string.alert\_rows;

break;

case ALERT\_CELLS:

alertMessage= R.string.alert\_cells;

break;

default:

return **null**;

}

Return **new** AlertDialog.Builder(this)

.setMessage(alertMessage)

.setNeutralButton(R.string.ok, doNothing)

.create();

}

В strings.xml при этом нужно добавить следующие значения:

<string name="alert\_columns">Incorrect columns number: should be from 5 to 25</string>

<string name="alert\_rows">Incorrect rows number: should be from 5 to 35</string>

<string name="alert\_cells">Number of cells is greater then the size of grid</string>

<string name="ok">OK</string>

Итак, в методе onCreateDialog мы решили, какое сообщение (alertMessage) будем выводить, создали диалог с помощью AlertDialog.Builder, вывели туда сообщение, поставили единственную кнопку OK и привязали к ней обработчик doNothing, который ничего не делает (нам, в принципе, делать ничего и не надо). Аналогичным образом можно добавить иконку формы, добавить ещё кнопок и т.д.

Подробнее про то, как строятся диалоги, можно посмотреть в адроидовских сэмплах - там на самом деле все достаточно понятно и просто.

Теперь при вводе некорректных данных мы видим вот это:



**Диалог для закрытия приложения**

Добавим на форму RunActivity кнопку Close. Для этого напишем такую разметку в run.xml:

<?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">

<TextView

android:id="@+id/message"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:paddingBottom="10dip"

/>

<Button

android:id="@+id/close"

android:text="@string/close"

android:textStyle="bold"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

/>

В strings.xml добавим следующие значения:

<string name="yes">Yes</string>

<string name="no">No</string>

<string name="close">Close</string>

<string name="submit\_close">Are you sure you want to close this wonderful application?</string>

Теперь вносим изменения в класс RunActivity:

**publicclass** RunActivity**extends** Activity implements OnClickListener

{

**public** static final String EXT\_COLS="cols";

**public** static final String EXT\_ROWS="rows";

**public** static final String EXT\_CELLS="cells";

Button mCloseButton;

*/\*\* Called when the activity is first created. \*/*

@Override

**public** void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.run);

mCloseButton=(Button) findViewById(R.id.close);

mCloseButton.setOnClickListener(this);

Bundle extras= getIntent().getExtras();

int cols= extras.getInt(EXT\_COLS);

int rows= extras.getInt(EXT\_ROWS);

int cells= extras.getInt(EXT\_CELLS);

TextView message=(TextView)findViewById(R.id.message);

message.setText("Rows: "+ rows+"**\n**Columns: "+ cols+"**\n**Cells: "+ cells);

}

**public** void onClick(View view)

{

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

builder

.setMessage(R.string.submit\_close)

*// кнопка "Yes", при нажатии на которую приложение закроется*

.setPositiveButton(R.string.yes,**new** DialogInterface.OnClickListener()

{

**public** void onClick(DialogInterface dialog, int whichButton)

{

finish();

}

})

*// кнопка "No", при нажатии на которую ничего не произойдет*

.setNegativeButton(R.string.no,**new** DialogInterface.OnClickListener()

{

**public** void onClick(DialogInterface dialog, int whichButton)

{

}

})

.show();

}

}

Объяснять тут особо нечего, и так все понятно. Теперь, при нажатии на кнопку Close, мы видим следующее:



При нажатии на Yes приложение закроется, на No — продолжит работу.

[Исходники](http://www.mobilab.ru/files/Life_part3.zip)