**Создание собственных диалогов**

Если есть необходимость в создании оригинального дизайна для диалога, можно создать собственную разметку для диалогового окна. После создания разметки необходимо передать корневой объект разметки в код создания диалога.

Чтобы получить корневой объект разметки, **используется класс Layoutinflater**. Этот класс нужен для преобразования XMLразметки в соответствующие объекты View в коде программы.

Сначала необходимо **инициализировать объект Layoutinflater** вызовом **метода getLayoutinfiater()**, затем получить корневое представление ***методом inflate (int, ViewGroup)***, где первый параметр— идентификатор ресурса схемы размещения, второй — идентификатор корневого представления разметки:

Layoutinflater inflater getLayoutlnflater();

View layout inflater.inflate(R.layout.custom\_layout,

(ViewGroup)findViewByid(R.id.toast\_layout));

Получив корневое представление, можно **методом findViewByid()**инициализировать все дочерние представления в разметке и задать для них информационное наполнение. Например, если в разметке определены ***виджеты TextView и ImageView***, код может выглядеть так:

TextView text (TextView)layout.findViewById(R.id.text);

text.setText("Are you sure you want to exit?");

ImageView image (ImageView)layout. findViewById(R.id. image);

image.setlmageResource(R.drawable.android3d);

Далее создается объект AlertDialog. Builder и методом setView () для него устанавливается полученная ранее разметка:

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

builder. setView(layout);

Остальная инициализация свойств диалога и работа с ним в коде программы аналогична работе **со стандартным AlertDialog.**

В качестве примера приложения с собственным диалоговым окном создайте новый проект и в диалоге *Create New Project* введите следующие значения:

* Project name — CustomDialogApp;
* Application name — CustomDialog Sample;
* Package name — сот. samples. customdialog;
* Create Activity — CustomDialogActivity.

Создайте также отдельный файл разметки для диалогового окна, как показано в листинге:

**Файл разметки custom\_layout.xml**

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

android:id="@+id/toast\_layout"

android:orientation="horizontal"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:padding="10dp"

android:background="#DAAA">

<ImageView android:id="@+id/image"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_marginRight="10dp"/>

<TextView android:id="@+id/text"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:textColor="#FFF"/>

</LinearLayout>

В **классе CustomDialogActivity** реализуйте создание и вызов диалогового окна с собственной разметкой, как было показано ранее в этом разделе. Полный ***код класса CustomDialogActivity*** показан в листинге:

**Файл CustomDialogActivlty.java**

package com.samples.customdialog;

import android.app.Activity;

import android.app.AlertDialog;

import android.app.Dialog;

import android.content.DialogInterface;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

public class CustomDialogActivity extends Activity {

private final static int IDD\_CUSTOM = 0;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

final Button callButton = (Button)findViewById(R.id.button);

// добавляем слушатель события для кнопки вызова диалога

callButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

showDialog(IDD\_CUSTOM);

}

});

}

@Override

protected Dialog onCreateDialog(int id) {

switch (id) {

case IDD\_CUSTOM:

LayoutInflater inflater = getLayoutInflater();

View layout = inflater.inflate(

R.layout.custom\_layout, (ViewGroup)findViewById(R.id.toast\_layout));

TextView text = (TextView)layout.findViewById(R.id.text);

text.setText("Are you sure you want to exit?");

ImageView image = (ImageView) layout.findViewById(R.id.image);

image.setImageResource(R.drawable.android3d);

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

builder.setView(layout);

builder.setMessage("This is a custom dialog!");

builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

CustomDialogActivity.this.finish();

}

});

builder.setNegativeButton("No", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

dialog.cancel();

}

});

builder.setCancelable(false);

return builder.create();

default:

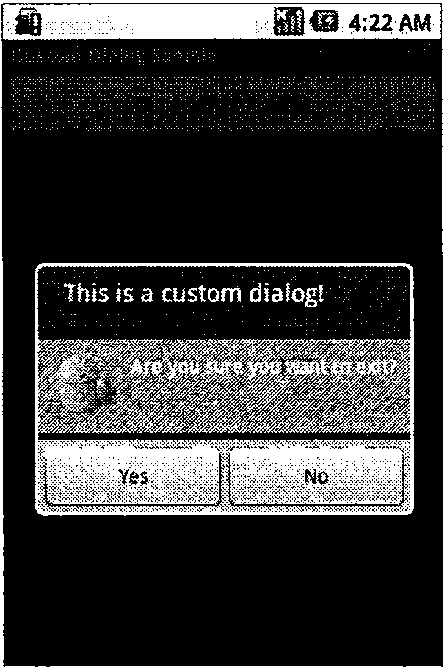
return null;

}

}

}

Запустите проект на выполнение. При нажатии кнопки вызова диалога должно появиться окно AlertDialog с кнопками Yes и No и с созданным нами нестандартным информационным наполнением. При закрытии диалога кнопкой Yes приложение закончит работу.



*Диалог с собственным дизайном*