Activities提供了一种方便管理的创建、保存、回复的对话框机制，例如 onCreateDialog(int), onPrepareDialog(int, Dialog), showDialog(int), dismissDialog(int)等方法，如果使用这些方法的话，Activity将通过getOwnerActivity()方法返回该Activity管理的对话框(dialog).

　　onCreateDialog(int):当你使用这个回调函数时，Android系统会有效的设置这个Activity为每个对话框的所有者，从而自动管理每个对话框的状态并挂靠到Activity上。这样，每个对话框继承这个Activity的特定属性。比如，当一个对话框打开时，菜单键显示为这个Activity定义的选项菜单，音量键修改Activity使用的音频流。

　　showDialog(int): 当你想要显示一个对话框时，调用showDialog(int id) 方法并传递一个唯一标识这个对话框的整数。当对话框第一次被请求时，Android从你的Activity中调用onCreateDialog(int id)，你应该在这里初始化这个对话框Dialog。这个回调方法被传以和showDialog(int id)相同的ID。当你创建这个对话框后，在Activity的最后返回这个对象。

　　onPrepareDialog(int, Dialog):在对话框被显示之前，Android还调用了可选的回调函数onPrepareDialog(int id, Dialog). 如果你想在每一次对话框被打开时改变它的任何属性，你可以定义这个方法。这个方法在每次打开对话框时被调用，而onCreateDialog(int) 仅在对话框第一次打开时被调用。如果你不定义onPrepareDialog()，那么这个对话框将保持和上次打开时一样。这个方法也被传递以对话框的ID，和在onCreateDialog()中创建的对话框对象。

　　dismissDialog(int):当你准备关闭对话框时，你可以通过对这个对话框调用dismiss()来消除它。如果需要，你还可以从这个Activity中调用dismissDialog(int id) 方法，这实际上将为你对这个对话框调用dismiss() 方法。 如果你想使用onCreateDialog(int id) 方法来管理你对话框的状态（就如同在前面的章节讨论的那样），然后每次你的对话框消除的时候，这个对话框对象的状态将由该Activity保留。如果你决定不再需要这个对象或者清除该状态是重要的，那么你应该调用removeDialog(int id)。这将删除任何内部对象引用而且如果这个对话框正在显示，它将被消除。

图1效果：该效果是当按返回按钮时弹出一个提示，来确保无误操作，采用常见的对话框样式。



　　　　　代码：

　　创建对话框方法dialog()

protected void dialog() {

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

builder.setMessage("确认退出吗？");

builder.setTitle("提示");

builder.setPositiveButton("确认", new OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

Main.this.finish();

}

});

builder.setNegativeButton("取消", new OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

}

});

builder.create().show();

}

在onKeyDown(int keyCode, KeyEvent event)方法中调用此方法

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (keyCode == KeyEvent.KEYCODE\_BACK && event.getRepeatCount() == 0) {

dialog();

}

return false;

}

图2效果：改变了对话框的图表，添加了三个按钮



Dialog dialog = new AlertDialog.Builder(this).setIcon(

android.R.drawable.btn\_star).setTitle("喜好调查").setMessage(

"你喜欢李连杰的电影吗？").setPositiveButton("很喜欢",

new OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub

Toast.makeText(Main.this, "我很喜欢他的电影。",

Toast.LENGTH\_LONG).show();

}

}).setNegativeButton("不喜欢", new OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub

Toast.makeText(Main.this, "我不喜欢他的电影。", Toast.LENGTH\_LONG)

.show();

}

}).setNeutralButton("一般", new OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub

Toast.makeText(Main.this, "谈不上喜欢不喜欢。", Toast.LENGTH\_LONG)

.show();

}

}).create();

dialog.show();

图3效果：信息内容是一个简单的View类型



new AlertDialog.Builder(this).setTitle("请输入").setIcon(

android.R.drawable.ic\_dialog\_info).setView(

new EditText(this)).setPositiveButton("确定", null)

.setNegativeButton("取消", null).show();

图4效果：信息内容是一组单选框



new AlertDialog.Builder(this).setTitle("复选框").setMultiChoiceItems(

new String[] { "Item1", "Item2" }, null, null)

.setPositiveButton("确定", null)

.setNegativeButton("取消", null).show();

图5效果：信息内容是一组多选框



new AlertDialog.Builder(this).setTitle("单选框").setIcon(

android.R.drawable.ic\_dialog\_info).setSingleChoiceItems(

new String[] { "Item1", "Item2" }, 0,

new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

}

}).setNegativeButton("取消", null).show();

图6效果：信息内容是一组简单列表项



new AlertDialog.Builder(this).setTitle("列表框").setItems(

new String[] { "Item1", "Item2" }, null).setNegativeButton(

"确定", null).show();

图7效果：信息内容是一个自定义的布局



1.布局文件

<?xml version="1.0" encoding="utf-8"?>

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

android:layout\_height="wrap\_content" android:layout\_width="wrap\_content"

android:background="#ffffffff" android:orientation="horizontal"

android:id="@+id/dialog">

<TextView android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:id="@+id/tvname" android:text="姓名：" />

<EditText android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content" android:id="@+id/etname" android:minWidth="100dip"/>

</LinearLayout>

2.调用代码

LayoutInflater inflater = getLayoutInflater();

View layout = inflater.inflate(R.layout.dialog,

(ViewGroup) findViewById(R.id.dialog));

new AlertDialog.Builder(this).setTitle("自定义布局").setView(layout)

.setPositiveButton("确定", null)

.setNegativeButton("取消", null).show();