Диалоговые окна в Android. Часть 1

Добрый день/вечер/утро, уважаемые хабравчане и Вы, %username%!  
Я занимаюсь разработкой приложений для Android, обожаю эту операционную систему и хочу поделиться своим опытом использования в своих проектах диалоговых окон. В первую очередь эта ~~статья~~ записка очень пригодится начинающим в области разработки для Android.  
Также для новичков в этой отрасли рекомендую сперва прочитать [вот этот](http://habrahabr.ru/post/109944/) пост уважаемого [Hoorsh](https://habrahabr.ru/users/hoorsh/).  
А теперь приступим к рассмотрению данного вопроса (у которого есть несколько подводных камней) под катом.

Dialog

Dialog — это класс, принадлежащий Android SDK и помогающий нам, простым ~~смертным~~ программистам, работать с диалоговыми окнами в Android. Класс Dialog имеет 4 подкласса:

* AlertDialog: это диалоговое окно для различных сообщений приложения, например «Вы хотите купить мое приложение?» или что то в этом роде. AlertDialog поддерживает три кнопки — утвердительную (OK), отрицательную (Cancel) и нейтральную (Later).
* ProgressDialog: это диалоговое окно для отображения выполнения различных процессов, загрузки, например.
* DatePickerDialog: диалоговое окно предлагает пользователю выбрать дату.
* TimePickerDialog: диалоговое окно предлагает пользователю выбрать время

На первом из этого списка, AlertDialog мы и остановимся подробнее.

AlertDialog

Чтобы создать диалоговое окно AlertDialog нам в помощь потребуется экземпляр класса Builder:  
AlertDialog.Builder builder = new AlertDialog.Builder(this);  
Теперь, когда у нас есть builder, мы можем «строить» свое собственное диалоговое окно. В параметры диалогового окна Вы можете написать множество различных параметров, но основными являются эти:

* setTitle(int resID) — задает заголовок диалогового окна, принимает в качестве аргументов ссылку на ресурс или строку.
* setMessage(int resID) — задает сообщение в диалоговом окне, также принимает ссылку на ресурс или строку.
* setPositiveButton(int textID, DialogInterface.OnClickListener listener) — устанавливает на Вашем диалоговом окне утвердительную кнопку с текстом ресурса textID и слушателем listener. Методы setNegativeButton и setNeutralButton идентичны, с разницей в назначении кнопок.

Пример создания AlertDialog:

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

builder.setTitle(R.string.dialog\_about\_title);

builder.setMessage(R.string.dialog\_about\_message);

builder.setCancelable(**true**);

builder.setPositiveButton(android.R.string.ok, **new** DialogInterface.OnClickListener() { *// Кнопка ОК*

**@Override**

**public** **void** **onClick**(DialogInterface dialog, **int** which) {

dialog.dismiss(); *// Отпускает диалоговое окно*

}

});

AlertDialog dialog = builder.create();

В этом примере я использовал setCancelable(true) — это разрешает пользователю закрывать диалоговое окно с помощью хардварной кнопки Back. Еще я создал слушателя для утвердительной кнопки и использовал метод create(). Метод create() возвращает готовое диалоговое окно с вашими параметрами как экземпляр класса AlertDialog.  
Вот мы и создали диалоговое окно! Следующая задача — показать его пользователю. Вот тут есть несколько вариантов:

* showDialog(AlertDialog dialog) — самый простой способ показать диалоговое окно, но начиная с версии 3.0 разработчики Android не рекомендуют пользоваться этим методом, и в результате вы получите предупреждение, которое можно обойти только @SurpressWarning, что есть не совсем хорошо.
* AlertDialog dialog.show() — альтернативный способ показать окно, для этого в экземпляре dialog должен лежать результат метода Builder.create().

Второй метод нам подходит больше, но писать в основной активности кучу различных Buider и AlertDialog портит читабельность кода. Поэтому целесообразней вынести создание и обработку диалоговых окон в новый класс, например:

**public** **class** **DialogScreen** {

**public** **static** **final** **int** IDD\_ABOUT = 1; *// Идентификаторы диалоговых окон*

**public** **static** **final** **int** IDD\_SETTINGS = 2;

**public** **static** **final** **int** IDD\_RATE = 3;

**public** **static** AlertDialog **getDialog**(Activity activity, **int** ID) {

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

**switch**(ID) {

**case** IDD\_ABOUT: *// Диалоговое окно About*

builder.setTitle(R.string.dialog\_about\_title);

builder.setMessage(R.string.dialog\_about\_message);

builder.setCancelable(**true**);

builder.setPositiveButton(android.R.string.ok, **new** DialogInterface.OnClickListener() { *// Кнопка ОК*

**@Override**

**public** **void** **onClick**(DialogInterface dialog, **int** which) {

dialog.dismiss(); *// Отпускает диалоговое окно*

}

});

**return** builder.create();

**case** IDD\_RATE: *// Диалоговое окно Rate the app*

builder.setTitle(R.string.dialog\_rate\_title);

builder.setMessage(R.string.dialog\_rate\_message);

builder.setCancelable(**true**);

builder.setPositiveButton(R.string.dialog\_rate\_ok, **new** DialogInterface.OnClickListener() { *// Переход на оценку приложения*

**@Override**

**public** **void** **onClick**(DialogInterface dialog, **int** which) {

*// Переход*

dialog.dismiss();

}

});

builder.setNeutralButton(R.string.dialog\_rate\_cancel, **new** DialogInterface.OnClickListener() { *// Оценить приложение потом*

**@Override**

**public** **void** **onClick**(DialogInterface dialog, **int** which) {

dialog.dismiss(); *// Отпускает диалоговое окно*

}

});

builder.setNegativeButton(R.string.dialog\_rate\_buy, **new** DialogInterface.OnClickListener() { *// Переход на покупку AdFree версии*

**@Override**

**public** **void** **onClick**(DialogInterface dialog, **int** which) {

*// Переход*

dialog.dismiss();

}

});

**return** builder.create();

**case** IDD\_SETTINGS: *// Диалог настроек*

View view = activity.getLayoutInflater().inflate(R.layout.settings, **null**); *// Получаем layout по его ID*

builder.setView(view);

builder.setTitle(R.string.dialog\_settings\_title);

builder.setPositiveButton(android.R.string.ok, **new** DialogInterface.OnClickListener() { *// Кнопка ОК*

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

MainActivity.doSaveSettings(); *// Переход в сохранение настроек MainActivity*

dialog.dismiss();

}

});

builder.setNegativeButton(android.R.string.cancel, **new** DialogInterface.OnClickListener() { *// Кнопка Отмена*

**public** **void** **onClick**(DialogInterface dialog, **int** which) {

dialog.dismiss();

}

});

builder.setCancelable(**true**);

**return** builder.create();

**default**:

**return** **null**;

}

}

}

Единственная заметка по поводу этого кода будет про использования метода setView(View view) в диалоге IDD\_SETTINGS. У AlertDialog как и у всех остальных диалоговых окон есть одна приятная особенность — им можно задавать собственные Layout, что позволяет полностью видоизменять диалоговые окна и их содержимое. Здесь есть один подводный камень: обработку элементов этого Layout вы должны будете производить именно в той Activity, где вызываете это диалоговое окно. Например я показываю диалоговое окно IDD\_SETTINGS в MainActivity с Layout по имени settings:

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<SeekBar

android:id="@+id/seekVol"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_below="@+id/seekSense"

android:layout\_marginTop="42dp" />

<TextView

android:id="@+id/textVol"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_below="@+id/seekSense"

android:layout\_marginTop="20dp"

android:text="@string/dialog\_settings\_vol"

android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView

android:id="@+id/textSense"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true"

android:text="@string/dialog\_settings\_sense"

android:textAppearance="?android:attr/textAppearanceMedium" />

<SeekBar

android:id="@+id/seekSense"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_below="@+id/textSense" />

</RelativeLayout>

Соответственно я загружаю это окно в MainActivity:

AlertDialog dialog = DialogScreen.getDialog(**this**, DialogScreen.IDD\_SETTINGS);

dialog.show();

initSettings(dialog);

}

Метод initSettings класса MainActivity в данном случае будет выглядеть так:

*// Определяем SeekBar и привязываем к нему дельты настроек*

SeekBar sb\_sense = (SeekBar)dialog.findViewById(R.id.seekSense);

SeekBar sb\_vol = (SeekBar)dialog.findViewById(R.id.seekVol);

*// Задаем этим SeekBar текущие значения настроек*

sb\_sense.setProgress(sense);

sb\_vol.setProgress(volume);

Ну а дальше обрабатываете свои объекты как вам угодно.

Небольшой итог

1) Для большинства целей диалоговых окон подходит AlertDialog  
2) AlertDialog может принимать вид layout с любыми объектами  
3) Гораздо лучше использовать для диалоговых окон отдельный класс и пользоваться AlertDialog.show()  
4) Обработка объектов кастомного layout производится в активности, вызвавшей диалог  
  
В следующей части этой статьи пойдет речь о DialogFragment, который был включен в Android SDK начиная с 3.0 Honeycomb. Надеюсь эта статья поможет новичкам и не очень в своих квестах по завоеванию Google Play.