**Пример AlertDialog (Диалоговое окно) в Android**

Компонент AlertDialog (диалоговое окно) — один из важных элементов пользовательского интерфейса в любом приложении на Андроид.AlertDialog чаще всего используется в тех случаях, когда нам спросить пользователя о чем-то (разрешение/подтверждение выполнения какой-то операции). В этом уроке мы рассмотрим создание диалогового окна с помощью различных компоновок.

**Теория по AlertDialog в Android**

AlertDialog является базовым компонентом для построения Android приложений, поэтому мы должны хорошо понимать, как он работает и уметь создавать различные компоновки диалоговых окон.

Для того, чтобы создать AlertDialog в Android, нам необходимо создать объект AlertDialogBuilder, который является [вложенным классом](http://prologistic.com.ua/java-vlozhenny-e-klassy-lokal-ny-j-klass-staticheskij-vlozhenny-j-klass-anonimny-j-vnutrennij-klass.html) AlertDialog. Рассмотрим пример:

Java

|  |
| --- |
| AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this); |

**Компоновка Android AlertDialog**

Компоновка диалогового окна позволяет работать с несколькими составляющими:

**Название** — необязательный элемент диалогового окна

**Контент** — текст, который будет показан пользователю. Это может быть сообщение, список или же свой полностью кастомный макет.

**Кнопки**. Они бывают трех типов: для подтверждения (ОК), отмены и дополнительная (настраиваемая) кнопка. Любое диалоговое окно может содержать максимум три кнопки действий.

**Методы для работы с диалоговыми окнами**

* Метод setIcon(Drawable icon) устанавливает иконку (значок) в диалоговое окно
* Метод setCancelable(boolean cancelable) устанавливает возможность отмены действия в AlertDialog.
* Метод setMessage(CharSequence message) устанавливает сообщение, которое будет отображаться в диалоговом окне.
* setMultiChoiceItems(CharSequence[] items, boolean[] checkedItems, DialogInterface.OnMultiChoiceClickListener listener) задает список элементов, которые будут отображаться в диалоговом окне. Выбранный пользователем элемент списка будет отслеживаться слушателем
* После нажатия на кнопку «Отмена» диалогового окна, сработает метод setOnCancelListener (DialogInterface.OnCancelListener onCancelListener).
* Метод setTitle(CharSequence title) устанавливает заголовок в AlertDialog.
* Метод getListView() получает тип списка, используемый в диалоговом окне.

**Пример использования AlertDialog в Android**

Давайте рассмотрим пример работы с AlertDialog. Начнем с макета, в котором расположим 3 кнопки. Каждая кнопка будет вызывать диалоговое окно с различной компоновкой: с 1 кнопкой — ОК, с двумя кнопками — Отмена и ОК, а также окно с тремя вариантами: ОК, Отмена и так называемая нейтральная кнопка.

Начнем с макета activity\_main.xml нашего приложения:

Java

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:paddingBottom="@dimen/activity\_vertical\_margin"      android:paddingLeft="@dimen/activity\_horizontal\_margin"      android:paddingRight="@dimen/activity\_horizontal\_margin"      android:paddingTop="@dimen/activity\_vertical\_margin"      tools:context=".MainActivity">        <Button          android:id="@+id/btn\_alertBtn1"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:text="@string/alertdialog\_1" />        <Button          android:id="@+id/btn\_alertBtn2"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_below="@+id/btn\_alertBtn1"          android:text="@string/alertdialog\_2" />        <Button          android:id="@+id/btn\_alertBtn3"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_below="@+id/btn\_alertBtn2"          android:text="@string/alertdialog\_3" />  </RelativeLayout> |

Как видно из макета, у нас 3 вложенных [виджета Button](http://prologistic.com.ua/sozdaem-knopku-v-android-primer-raboty-s-vidzhetom-button-v-android.html) в корневой [RelativeLayout](http://prologistic.com.ua/android-layouts-obzor-relativelayout-i-linearlayout.html). Атрибуты android:text как и полагается, вынесены в strings.xml в папке **res/values/**:

Java

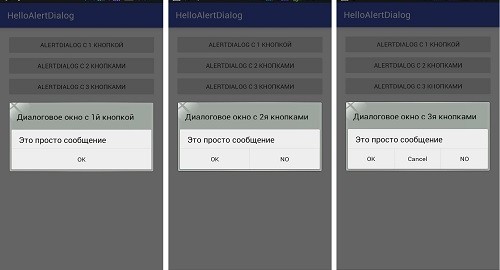
|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <resources>      <string name="app\_name">HelloAlertDialog</string>      <string name="alertdialog\_1">AlertDialog с 1 кнопкой</string>      <string name="alertdialog\_2">AlertDialog с 2 кнопками</string>      <string name="alertdialog\_3">AlertDialog с 3 кнопками</string>  </resources> |

Теперь инициализируем наши виджеты в коде. Также нам нужно проставить слушатели на кнопки и по нажатию вызывать нужный AlertDialog. Сделаем мы это через реализацию интерфейса View.OnClickListener, а вызов нужного диалогового окна будет происходить в операторе switch. Листинг класса MainActivity приведен ниже:

Java

|  |
| --- |
| package ua.com.prologistic;    import android.app.AlertDialog;  import android.content.DialogInterface;  import android.os.Bundle;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.Button;  import android.widget.Toast;    public class MainActivity extends AppCompatActivity implements View.OnClickListener {        private Button btnOne;      private Button btnTwo;      private Button btnThree;    *@Override*      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            // инициализируем кнопки по идентификаторах в макете          btnOne = (Button) findViewById(R.id.btn\_alertBtn1);          btnTwo = (Button) findViewById(R.id.btn\_alertBtn2);          btnThree = (Button) findViewById(R.id.btn\_alertBtn3);            // устанавливаем слушатели          btnOne.setOnClickListener(this);          btnTwo.setOnClickListener(this);          btnThree.setOnClickListener(this);      }          // показываем AlertDialog в зависимости от нажатой клавиши  *@Override*      public void onClick(View v) {          int id = v.getId();          // находит по id нажатый виджет Button и вызывает нужный метод          switch (id) {              case R.id.btn\_alertBtn1:                  createOneButtonAlertDialog("Диалоговое окно с 1й кнопкой", "Это просто сообщение");                    break;              case R.id.btn\_alertBtn2:                  createTwoButtonsAlertDialog("Диалоговое окно с 2я кнопками", "Это просто сообщение");                  break;                case R.id.btn\_alertBtn3:                  createThreeButtonsAlertDialog("Диалоговое окно с 3я кнопками", "Это просто сообщение");                  break;          }      }        // создает диалоговое окно с 1й кнопкой      private void createOneButtonAlertDialog(String title, String content) {          // объект Builder для создания диалогового окна          AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);          // добавляем различные компоненты в диалоговое окно          builder.setTitle(title);          builder.setMessage(content);          // устанавливаем кнопку, которая отвечает за позитивный ответ          builder.setPositiveButton("OK",                  // устанавливаем слушатель                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog,                                          int which) {                          // по нажатию создаем всплывающее окно с типом нажатой конпки                          showMessage("Нажали ОК");                      }                  });          // объект Builder создал диалоговое окно и оно готово появиться на экране          // вызываем этот метод, чтобы показать AlertDialog на экране пользователя          builder.show();      }        // создает диалоговое окно с тремя кнопкаи      private void createThreeButtonsAlertDialog(String title, String content) {          AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);          builder.setTitle(title);          builder.setMessage(content);          builder.setNegativeButton("NO",                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog,                                          int which) {                          showMessage("Нажали Нет");                      }                  });          builder.setPositiveButton("OK",                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog,                                          int which) {                          showMessage("Нажали ОК");                      }                  });          // устанавливаем кнопку, которая отвечает за выбранный нами ответ          // в данном случаем мы просто хотим всплывающее окно с отменой          builder.setNeutralButton("Cancel",                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog,                                          int which) {                          showMessage("Нажали Отмена");                      }                  });            builder.show();      }        // создает диалоговое окно с двумя кнопками      private void createTwoButtonsAlertDialog(String title, String content) {          AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);          builder.setTitle(title);          builder.setMessage(content);          builder.setNegativeButton("NO",                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog,                                          int which) {                          showMessage("Нажали Нет");                      }                  });          builder.setPositiveButton("OK",                  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog,                                          int which) {                          showMessage("Нажали ОК");                      }                  });          builder.show();      }        // вспомогательный метод для вывода всплывающих сообщений      private void showMessage(String textInMessage) {          Toast.makeText(getApplicationContext(), textInMessage, Toast.LENGTH\_LONG).show();      }  } |

В результате мы получим сборный результат нажатий на каждую кнопку. После нажатия на каждую кнопку появляется всплывающее окно:

[](http://prologistic.com.ua/wp-content/uploads/2016/03/AlertDialogResult.jpg)