**Диалоги в Android. Пример работы с ProgressDialog**

В этом уроке мы создадим ProgressDialog, а также рассмотрим роль ProgressBar в ProgressDialog.

**Диалоги в Android. ProgressDialog**

ProgressDialog является расширением виджета AlertDialog ([подробнее о AlertDialog](https://javadevblog.com/primer-alertdialog-dialogovoe-okno-v-android.html)).  
ProgressDialog в Android представляет собой диалоговое окно с ходом (прогрессом) выполнения какой-то задачи. Сам ProgressDialog очень похож на ProgressBar за исключением того, что отображается в диалоговом окне. Чтобы создать ProgressDialog, нужно достаточно следующей строчки кода:

|  |  |
| --- | --- |
| 1 | ProgressDialog progressDialog = new ProgressDialog(this); |

**Разница между ProgressDialog и ProgressBar**

* ProgressBar представляет собой обычный View (как [Button](https://javadevblog.com/sozdaem-knopku-v-android-primer-raboty-s-vidzhetom-button-v-android.html" \t "_blank), [ImageView](https://javadevblog.com/primer-ispol-zovaniya-vidzheta-imageview-v-android.html" \t "_blank) и другие) и может быть использован в макете, чтобы показать прогресс выполнения какой-то задачи. Кроме того, он не мешает пользователю взаимодействовать с другими частями приложения.
* ProgressDialog представляет собой диалог со встроенным ProgressBar. Он также используется, чтобы показать прогресс выполнения какой-то задачи, но закрывает пользователю доступ к другим элементам экрана и требует дождаться выполнения задачи.

**Методы для работы с диалогом ProgressDialog**

* Метод setMessage() используется для отображения сообщения пользователю. В нем часто пишут что-то на подобии «Идет загрузка…».
* Метод setTitle() используется для установки заголовка в диалог.
* Метод setProgressStyle(ProgressDialog.STYLE\_HORIZONTAL) используется для указания горизонтального индикатора выполнения в диалоговом окне.
* Метод setProgressStyle(ProgressDialog.STYLE\_SPINNER) используется для демонстрации пользователю прогресса выполнения какой-либо задачи в виде вращающегося индикатора.
* Метод setMax() используется для установки максимального значения прогресса.
* Метод getMax() возвращает максимальное значение прогресса.
* Метод getProgress() используется для получения текущего значения прогресса в цифрах.
* Статический метод show(Context context, CharSequence title, CharSequence message) используется для отображения диалога ProgressDialog на экране.
* Метод incrementProgressBy(int diff) увеличивает индикатор на значение, переданное в качестве параметра diff.

**Пример работы с ProgressDialog**

Ниже мы рассмотри небольшой пример работы с диалогом ProgressDialog.

Наш XML макет:

|  |
| --- |
| <?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"      tools:context=".MainActivity" >        <Button          android:id="@+id/btn\_start\_horiz\_dialog"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="@string/horizontal\_dialog"          android:layout\_centerVertical="true"          android:layout\_centerHorizontal="true" />      <Button          android:id="@+id/btn\_start\_simple\_dialog"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="@string/simple\_dialog"          android:layout\_below="@+id/btn\_start\_horiz\_dialog"          android:layout\_centerHorizontal="true" />  </RelativeLayout> |

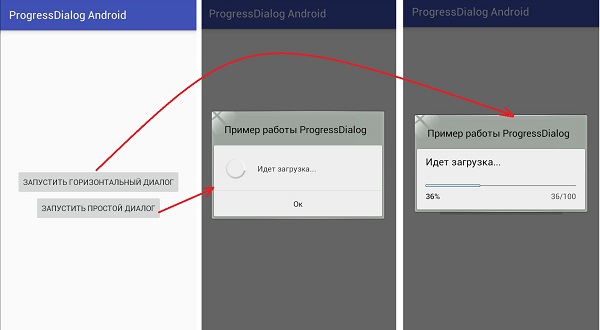
Листинг класса MainActivity:

Java

|  |
| --- |
| package ua.com.prologistic;    import android.app.Dialog;  import android.app.ProgressDialog;  import android.content.DialogInterface;  import android.os.Bundle;  import android.os.Handler;  import android.os.Message;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.Button;    public class MainActivity extends AppCompatActivity implements View.OnClickListener {        Button btnHorizontalDialog;      Button btnSimpleDialog;        ProgressDialog progressDialog;        Handler handler;    *@Override*      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            btnHorizontalDialog = (Button) findViewById(R.id.btn\_start\_horiz\_dialog);          btnSimpleDialog = (Button) findViewById(R.id.btn\_start\_simple\_dialog);            btnSimpleDialog.setOnClickListener(this);          btnHorizontalDialog.setOnClickListener(this);      }    *@Override*      public void onClick(View v) {          switch (v.getId()) {              case R.id.btn\_start\_simple\_dialog: {                  // создает простой диалог с бесконечной анимацией                  // (пока не нажали на кнопку Ок)                  progressDialog = new ProgressDialog(this);                  progressDialog.setMax(100);                  progressDialog.setMessage("Идет загрузка...");                  progressDialog.setTitle("Пример работы ProgressDialog");  progressDialog.setButton(Dialog.BUTTON\_POSITIVE, "Oк",  new DialogInterface.OnClickListener() {                      public void onClick(DialogInterface dialog, int which) {                      }                  });                  progressDialog.show();                    break;              }     case R.id.btn\_start\_horiz\_dialog: {                  progressDialog = new ProgressDialog(this);                  progressDialog.setMax(100);                  progressDialog.setMessage("Идет загрузка...");                  progressDialog.setTitle("Пример работы ProgressDialog");                  progressDialog.setProgressStyle(ProgressDialog.STYLE\_HORIZONTAL);                  progressDialog.show();                    handler = new Handler() {                      public void handleMessage(Message msg) {                          // и обновляем идикатор, пока шкала не заполнится                          if (progressDialog.getProgress() < progressDialog.getMax()) {                              // обновляем индикаторы на 3 пункта за 1 секунду (1000 милисекунд)                              progressDialog.incrementProgressBy(3);                              handler.sendEmptyMessageDelayed(0, 1000);                          } else {                              // когда шкала заполнилась, диалог пропадает                              progressDialog.dismiss();                          }                      }                  };                  // имитируем подключение к удаленному серверу                  // (ожидаем 1 секунду перед стартом обновления индикатора)                  handler.sendEmptyMessageDelayed(0, 1000);                  break;              }          }      }  } |

В нашем приложении использовался **Handler — механизм, позволяющий обрабатывать очередь из сообщений**. Мы можем отправлять, получать сообщения, задавать интервалы получения этих сообщений самому себе. Подробно с этим механизмом мы разберемся в следующих уроках, а пока достаточно того, что он позволяет обновлять индикатор не в своем потоке.

Результат работы приложения с использованием ProgressDialog:

[](https://javadevblog.com/wp-content/uploads/2016/03/simple.jpg)

Следите за обновлениями и подписывайтесь на рассылку новых