ProgressBar - Индикатор прогресса

[Основы](http://developer.alexanderklimov.ru/android/views/progressbar.php#basic)  
[Отдельный поток](http://developer.alexanderklimov.ru/android/views/progressbar.php#thread)  
[Вертикальный индикатор прогресса](http://developer.alexanderklimov.ru/android/views/progressbar.php#vertical)  
[Стилизация](http://developer.alexanderklimov.ru/android/views/progressbar.php#stylebar)  
[Индикатор с текстом](http://developer.alexanderklimov.ru/android/views/progressbar.php#relative)  
[Программное создание индикатора](http://developer.alexanderklimov.ru/android/views/progressbar.php#addview)

Основы

Компонент **ProgressBar** (Индикатор прогресса) применяется в тех случаях, когда пользователю нужно показать, что программа не зависла, а выполняет продолжительную работу.

Находится в категории **Widgets**.

Вы увидите, что студия предлагает несколько видов индикаторов: **ProgressBar (Large)**, **ProgressBar (Normal)**, **ProgressBar (Small)** и **ProgressBar (Horizontal)**. Стандартным является второй индикатор **Normal**, остальные являются модификациями и их отличия заложены в стилях, если посмотреть XML-описание компонентов:

style="?android:attr/progressBarStyleLarge"

style="?android:attr/progressBarStyleSmall"

style="?android:attr/progressBarStyleHorizontal"

Методы для работы с ProgressBar:

* **setProgress()** — устанавливает заданное значение индикатора прогресса;
* **getProgress()** — возвращает текущее значение индикатора прогресса;
* **incrementProgressBy()** — устанавливает величину дискретизации приращения значения индикатора;
* **setMax()** — устанавливает максимальное значение величины прогресса.

Круговые индикаторы **Normal**, **Small** и **Large** можно просто разместить на экране, и они будут бесконечно выводить анимацию вращения без единой строчки кода.

Как правило, "бесконечный" индикатор выводят в нужный момент, а когда задача выполнена, то его скрывают. Видимость можно установить через XML и программно.

<ProgressBar

android:id="@+id/progressBar"

**android:visibility="invisible"**

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

В коде используется метод **setVisibility()**

ProgressBar progressBar = (ProgressBar) findViewById(R.id.progressBar);

progressBar.setVisibility(ProgressBar.VISIBLE);

// запускаем длительную операцию

progressBar.setVisibility(ProgressBar.INVISIBLE);

Горизонтальный индикатор обычно используют, чтобы показать индикатор в действии. При длительной операции пользователь видит, что полоска индикатора постоянно увеличивается.

Отдельный поток

Выполнение длительных операций предпочтительнее производить в отдельном потоке. Android имеет специальный класс [Handler](http://developer.alexanderklimov.ru/android/theory/thread.php) для создания фоновых потоков и их взаимодействия с пользовательским интерфейсом. Либо используется [AsyncTask](http://developer.alexanderklimov.ru/android/theory/asynctask.php) и другие способы выполнения в другом потоке.

Вертикальный индикатор прогресса

По умолчанию используется горизонтальный индикатор. Можно создать собственный индикатор с вертикальной ориентацией. В папке **res/drawable** подготовим два файла:

verticalprogressbar.xml

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

<layer-list xmlns:android="http://schemas.android.com/apk/res/android" >

<item android:id="@android:id/background">

<shape>

<corners android:radius="5dip" />

<solid android:color="#FFFFFF" />

<stroke

android:width="1dp"

android:color="#FF000000" />

</shape>

</item>

<item android:id="@android:id/progress">

<clip

android:clipOrientation="vertical"

android:gravity="bottom" >

<shape>

<corners android:radius="5dip" />

<solid android:color="#FFAADD" />

<stroke

android:width="1dp"

android:color="#FF000000" />

</shape>

</clip>

</item>

</layer-list>

verticalprogressbar2.xml

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

<layer-list xmlns:android="http://schemas.android.com/apk/res/android" >

<item android:id="@android:id/background">

<shape>

<corners android:radius="5dip" />

<solid android:color="#FFFFFF" />

<stroke

android:width="1dp"

android:color="#FF000000" />

</shape>

</item>

<item android:id="@android:id/progress">

<clip

android:clipOrientation="vertical"

android:gravity="bottom" >

<shape>

<corners android:radius="5dip" />

<solid android:color="#00FF55" />

<stroke

android:width="1dp"

android:color="#FF000000" />

</shape>

</clip>

</item>

</layer-list>

Разница между двумя файлами только в цвете, которым закрашивается индикатор во время работы. Я создал второй файл только для демонстрации программного изменения индикатора - когда значение индикатора достигнет отметки 50, то цвет поменяется на зелёный.

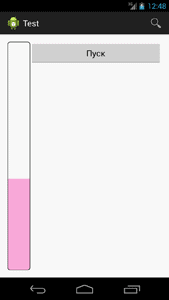
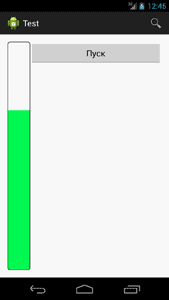
Код можно вставить после строчки из предыдущего примера:

// код из предыдущего примера

Thread.sleep(1000);

// вставим код для изменения цвета

if(myProgress >= 50) myProgressBar.setProgressDrawable(getResources().getDrawable(R.drawable.verticalprogressbar2));

Стилизация

Если вам не нравится внешний вид стандартных индикаторов, то можете придумать свой. Есть два варианта для реализации этой цели. В первом случае можно подготовить готовый png-файл (spin\_custom.png) и поместить его в ресурс, например, в папку **res/drawable**.

Custom Spin

Затем в этой же папке создаём xml-файл *spinner\_png.xml* с таким содержанием:

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

<animated-rotate xmlns:android="http://schemas.android.com/apk/res/android"

android:drawable="@drawable/spin\_custom"

android:pivotX="50%"

android:pivotY="50%"

/>

Теперь в разметке активности можно определять **ProgressBar** так:

<ProgressBar

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:indeterminateDrawable="@drawable/spinner\_png"

android:indeterminateOnly="true" />

Если запустить проект, то увидим такой индикатор:

http://developer.alexanderklimov.ru/android/views/spinner_png.gif

В реальности индикатор выглядит немного корявым, подёргиваясь на экране. Кроме того, изображение можно улучшить, уменьшив количество цветов и сократив размер файла.

Второй вариант выглядит предпочтительней. На этот раз мы обойдёмся без графических файлов, а будем использовать только XML-разметку. В этой же папке поместим файл*spinner\_ring.xml* следующего содержания:

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

<animated-rotate xmlns:android="http://schemas.android.com/apk/res/android"

android:pivotX="50%"

android:pivotY="50%" >

<shape

android:innerRadiusRatio="4"

android:shape="ring"

android:thicknessRatio="5.333"

android:useLevel="false" >

<size

android:height="18dip"

android:width="18dip" />

<gradient

android:centerColor="#886688cc"

android:centerY="0.50"

android:endColor="#ff6688cc"

android:startColor="#006688cc"

android:type="sweep"

android:useLevel="false" />

</shape>

</animated-rotate>

Теперь в разметке активности пропишем строки:

<ProgressBar

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="20dip"

android:indeterminateDrawable="@drawable/spinner\_ring"

android:indeterminateOnly="true" />

Получим следующий результат:

http://developer.alexanderklimov.ru/android/views/spinner_ring.gif

[Источник](http://megadarja.blogspot.com/2011/03/progressbar.html)

Ещё несколько вариантов индикаторов можно найти [здесь](https://github.com/iklementiev/custom-progressbar).



[Описание от автора](http://k1im.ru/blog/kastomizaciya-progressbar-v-android).

Индикатор с текстом

К сожалению, индикатор не выводит текст во время работы, что не очень удобно. Можно создать свой [ProgressBar из TextView](http://developer.alexanderklimov.ru/android/views/progresstextview.php) или создать свой компонент на основе **ProgressBar**. Другой простой способ - наложить текстовый блок прямо на индикатор, используя **RelativeLayout**. Пример для пояснения.

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="fill\_parent"

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

<RelativeLayout

android:id="@+id/rl\_progress\_bar\_set"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/btn\_start"

android:visibility="visible" >

<ProgressBar

android:id="@+id/pb\_horizontal"

style="@android:style/Widget.ProgressBar.Horizontal"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:max="100" />

<TextView

android:id="@+id/tv\_progress\_horizontal"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@id/pb\_horizontal"

android:background="@android:color/transparent"

android:gravity="center" />

<RelativeLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_below="@id/pb\_horizontal"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp" >

<ProgressBar

style="@android:style/Widget.ProgressBar.Large"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:background="@android:color/background\_dark" />

<TextView

android:id="@+id/tv\_progress\_circle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:background="@android:color/transparent"

android:gravity="center"

android:textColor="@android:color/white" />

</RelativeLayout>

</RelativeLayout>

<Button

android:id="@+id/btn\_start"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="15dp"

android:onClick="onClick"

android:text="Поехали"

android:textSize="25dp" />

</RelativeLayout>

Код, чтобы увидеть работу индикатора. При каждом щелчке кнопки значение индикаторов увеличиваются на 10 процентов.

package ru.alexanderklimov.test;

import ...

public class MainActivity extends Activity {

private int progress = 0;

private ProgressBar pbHorizontal;

private TextView tvProgressHorizontal;

private TextView tvProgressCircle;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

pbHorizontal = (ProgressBar) findViewById(R.id.pb\_horizontal);

tvProgressHorizontal = (TextView) findViewById(R.id.tv\_progress\_horizontal);

tvProgressCircle = (TextView)findViewById(R.id.tv\_progress\_circle);

}

public void onClick(View v) {

progress = progress + 10;

postProgress(progress);

}

private void postProgress(int progress) {

String strProgress = String.valueOf(progress) + " %";

pbHorizontal.setProgress(progress);

if (progress == 0) {

pbHorizontal.setSecondaryProgress(0);

} else {

pbHorizontal.setSecondaryProgress(progress + 5);

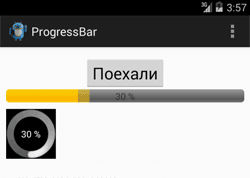
}

tvProgressHorizontal.setText(strProgress);

tvProgressCircle.setText(strProgress);

}

}



Программное создание индикатора

Данный способ уже устарел, так как через XML намного удобнее. Для общего развития.

// Зададим цвета для фона и самого индикатора

Drawable background = new ColorDrawable(0xFF373737);

Drawable progress = new ColorDrawable(0xFF00B51C);

ClipDrawable clipProgress = new ClipDrawable(progress, Gravity.LEFT,

ClipDrawable.HORIZONTAL);

// Слои для индикатора

LayerDrawable layerlist = new LayerDrawable(new Drawable[] {

background, clipProgress });

layerlist.setId(0, android.R.id.background);

layerlist.setId(1, android.R.id.progress);

// Создаем новый индикатор с горизонтальным стилем и заданным фоном

ProgressBar progressBar = new ProgressBar(this, null,

android.R.attr.progressBarStyleHorizontal);

progressBar.setLayoutParams(new LinearLayout.LayoutParams(

LinearLayout.LayoutParams.MATCH\_PARENT,

LinearLayout.LayoutParams.WRAP\_CONTENT));

progressBar.setProgressDrawable(layerlist);

progressBar.setProgress(76);

LinearLayout linearlayout = (LinearLayout) findViewById(R.id.linearlayout);

linearlayout.addView(progressBar);

Индикаторы прогресса также можно использовать в заголовках приложения у старых устройств, а также в **ActionBar** и **ToolBar** у новых устройств.

Библиотеки

[castorflex/SmoothProgressBar](https://github.com/castorflex/SmoothProgressBar)

[Todd-Davies/ProgressWheel](https://github.com/Todd-Davies/ProgressWheel)

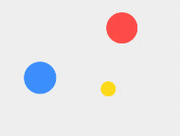
[passsy/android-HoloCircularProgressBar](https://github.com/passsy/android-HoloCircularProgressBar)

[daimajia/NumberProgressBar](https://github.com/daimajia/NumberProgressBar)

[lsjwzh/MaterialLoadingProgressBar](https://github.com/lsjwzh/MaterialLoadingProgressBar)

[Подборка индикаторов](https://github.com/wasabeef/awesome-android-ui/blob/master/pages/Progress.md)

[glomadrian/loading-balls: A highly configurable library to do loading progress with animated balls](https://github.com/glomadrian/loading-balls) - анимация для длительных операций.



Реклама