**Анимация разметки и групп представлений**

Класс **LayoutAnimation** нужен для применения анимации к группам представлений: передает одиночный объект **Animation** (или **AnimationSet**) для каждого дочернего представления в заранее определенной последовательности.

Используйте объект **LayoutAnimationController** для задания объекта **Animation** (или **AnimationSet**), который применяется к каждому представлению в группе. Ко всем представлениям, содержащим этот объект, будет применена одна и та же анимация, но вы можете задействовать **LayoutAnimationController** для указания порядка и времени запуска для каждого представления.

Для этих целей в Android предусмотрено два класса:

* **LayoutAnimationController** — позволяет выбирать начальный сдвиг для каждого представления (в миллисекундах) и порядок (нормальный, обратный или случайный), в каком анимация применится для каждого дочернего элемента;
* **GridLayoutAnimationController** — как производный класс назначает анимационную последовательность для дочерних представлений с использованием ссылки на строки и столбцы;

**Создание анимации для разметки**

Чтобы создать новую анимацию для разметки, нужно определить её для каждого дочернего представления. После создайте новый элемент **LayoutAnimation** (либо в коде программы, либо в виде внешнего ресурса), ссылающийся на ранее выполненную анимацию и описывающий порядок и время срабатывания, с которыми её необходимо применить.

Создадим простую анимацию в файле **popin.xml** в каталоге **res/anim**

**popin.xml**

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

android:interpolator="@android:anim/accelerate\_interpolator" >

<scale

android:duration="400"

android:fromXScale="0.0"

android:fromYScale="0.0"

android:pivotX="50%"

android:pivotY="50%"

android:toXScale="1.0"

android:toYScale="1.0" />

</set>

А также зададим описание **LayoutAnimation** в файле **popinlayout.xml** в той же папке. Элемент **LayoutAnimation** в случайном порядке применит всплывающую анимацию к каждому дочернему представлению заданной группы.

**popinlayout.xml**

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

android:animation="@anim/popin"

android:animationOrder="random"

android:delay="0.5" />

Теперь можно применить созданную анимацию к группе представлений двумя способами - с помощью XML-ресурса или программно. В XML это делается при помощи атрибута **android:layoutAnimation** у разметки:

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

...

android:layoutAnimation="@anim/popinlayout" >

Чтобы применить анимацию для разметки в коде программы, вызовите метод **setLayoutAnimation()** из объекта **ViewGroup**, передавая ему ссылку на соответствующий экземпляр **LayoutAnimation**.

В обоих случаях анимация выполнится один раз, при первом появлении группы представлений. Вы можете вызвать её повторно, запустив метод **scheduleLayoutAnimation()** из объекта **ViewGroup**. Таким образом, анимация выполнится при следующем появлении группы представлений.

Анимация для разметки также имеет поддержку интерфейса **AnimationListener**.

В следующем листинге показан повторный запуск анимации в контексте группы представлений с помощью интерфейса **AnimationListener**, который используется для выполнения дополнительных действий по завершении изначальной анимации.

aViewGroup.setLayoutAnimationListener(new AnimationListener() {

public void onAnimationEnd(Animation \_animation) {

// TODO: Действия, выполняемые по завершении выполнения анимации.

}

public void onAnimationRepeat(Animation \_animation) {}

public void onAnimationStart(Animation \_animation) {}

});

aViewGroup.scheduleLayoutAnimation();

**Примеры**

[Анимация при загрузке списка](http://developer.alexanderklimov.ru/android/listactivity.php#animation)