**ViewFlipper**

[Простой пример](http://developer.alexanderklimov.ru/android/views/viewflipper.php#simple)  
[Слайд-шоу](http://developer.alexanderklimov.ru/android/views/viewflipper.php#autoslide)  
[Анимация переходов](http://developer.alexanderklimov.ru/android/views/viewflipper.php#anim)

**Простой пример**

Виджет **ViewFlipper** позволяет использовать режим просмотра слайдов. Суть работы заключается в следующем - вы добавляете в контейнер **ViewFlipper** дочерние элементы-слайды, которые затем могут по очереди показываться. Показываться может только один дочерний элемент, остальные будут скрыты. Также можно настроить автоматическую смену слайдов.

Находится в папке **Transitions**.

Рассмотрим самый простой пример.

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello" />

<ViewFlipper

android:id="@+id/viewflipper"

android:layout\_width="match\_parent"

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

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:background="#00ff00">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Первый экран" />

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="На следующую страницу" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:background="#ff0000" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Второй экран" />

<Button

android:id="@+id/button2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="На предыдущую страницу" />

</LinearLayout>

</ViewFlipper>

</LinearLayout>

Как видите, в **ViewFlipper** я поместил два дочерних элемента **LinearLayout**, которые закрасил разными цветами, чтобы лучше ориентироваться между ними. У каждого дочернего элемента есть кнопка, чтобы у нас была возможность переместиться на другой экран. Конечно, удобнее было бы перемещаться при помощи жеста прокрутки, но для первого примера ограничимся щелчками кнопок.

...

setContentView(R.layout.activity\_main);

final ViewFlipper viewFlipper = (ViewFlipper) findViewById(R.id.viewflipper);

Button button1 = (Button) findViewById(R.id.button1);

Button button2 = (Button) findViewById(R.id.button2);

button1.setOnClickListener(new Button.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

viewFlipper.showNext();

}

});

button2.setOnClickListener(new Button.OnClickListener() {

@Override

public void onClick(View v) {

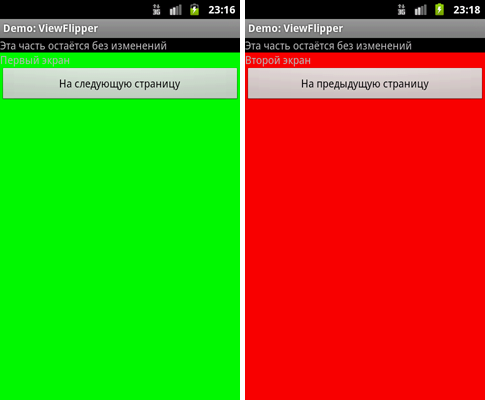
// TODO Auto-generated method stub

viewFlipper.showPrevious();

}

});

Для перехода на следующий слайд используется метод **showNext()**, а для возврата на предыдущий слайд - метод **showPrevious()**.



**Слайд-шоу**

Шоу должно продолжаться! У **ViewFlipper** есть встроенная возможность включения автоматического переключения между дочерними элементами.

Можно обойтись без написания кода и включить слайд-шоу, указав атрибуты **autoStart** и **flipInterval**.

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<ViewFlipper

android:id="@+id/viewFlipper1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:autoStart="true"

android:flipInterval="1500" >

<ImageView

android:id="@+id/imageView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat1" />

<ImageView

android:id="@+id/imageView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat2" />

<ImageView

android:id="@+id/imageView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat3" />

<ImageView

android:id="@+id/imageView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat4" />

<ImageView

android:id="@+id/imageView5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat5" />

</ViewFlipper>

</LinearLayout>

Можно управлять эффектом слайд-шоу программно. Модифицируем немного один из предыдущих примеров, просто добавив сверху ещё одну кнопку для запуска слайд-шоу.

...

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello" />

**<Button**

**android:id="@+id/buttonautoflip"**

**android:layout\_width="match\_parent"**

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

**android:text="Пуск" />**

<ViewFlipper

android:id="@+id/viewflipper"

android:layout\_width="match\_parent"

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

...

Теперь осталось написать код для слайд-шоу. Интервал задается при помощи метода **setFlipInterval()**, а за смену слайдов отвечает связка методов с говорящими именами **isFlipping()**, **startFlipping()**, **stopFlipping**.

Также есть методы, связанные с автозапуском - **isAutoStart()** и **setAutoStart()**.

final Button buttonAutoFlip = (Button) findViewById(R.id.buttonautoflip);

viewFlipper.setFlipInterval(2000);

buttonAutoFlip.setOnClickListener(new Button.OnClickListener() {

@Override

public void onClick(View v) {

if (viewFlipper.isFlipping()) {

viewFlipper.stopFlipping();

buttonAutoFlip.setText("Пуск");

} else {

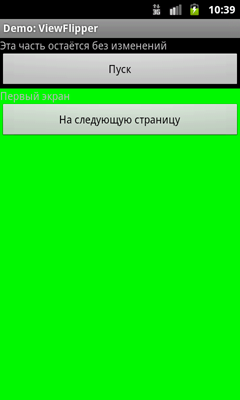
viewFlipper.startFlipping();

buttonAutoFlip.setText("Стоп");

}

}

});



Напишем более универсальный пример, позволяющий выбрать режим слайд-шоу: автоматический или через кнопку вручную

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<LinearLayout

android:id="@+id/linearLayout1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal" >

<Button

android:id="@+id/flipbyinterval"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="0.25"

android:onClick="flipByInterval"

android:text="Старт" />

<Button

android:id="@+id/flipbyclick"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="0.49"

android:onClick="flipByClick"

android:text="Flip By Click" />

</LinearLayout>

<ViewFlipper

android:id="@+id/viewFlipper1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:autoStart="true"

android:flipInterval="1500" >

<ImageView

android:id="@+id/imageView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat1" />

<ImageView

android:id="@+id/imageView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat2" />

<ImageView

android:id="@+id/imageView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat3" />

<ImageView

android:id="@+id/imageView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat4" />

<ImageView

android:id="@+id/imageView5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:src="@drawable/cat5" />

</ViewFlipper>

</LinearLayout>

Код.

package ru.alexanderklimov.test;

import ...

public class MainActivity extends Activity {

ViewFlipper flipper;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

flipper = (ViewFlipper) findViewById(R.id.viewFlipper1);

}

// Первая кнопка

public void flipByClick(View v) {

if (flipper.isFlipping())// Checking flipper is flipping or not.

{

flipper.stopFlipping(); // stops the flipping .

}

flipper.showNext();// shows the next view element of ViewFlipper

}

// Вторая кнопка

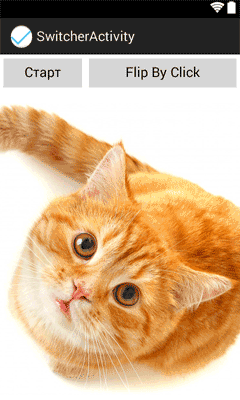
public void flipByInterval(View v) {

flipper.setFlipInterval(1500);// setting the interval 1500 milliseconds

flipper.startFlipping(); // views flipping starts.

}

}



**Анимация переходов**

Так как **ViewFlipper** является подклассом класса **android.widget.ViewAnimator**, то было бы неплохо использовать эту возможность в примере. Сейчас у нас один слайд просто заменяется другим без всякой анимации. Сделаем слайд-шоу более реалистичным, когда один слайд выталкивает другой. Для этого нужно создать новую папку **res/anim** и поместить в нём два файла анимации **flipin.xml** и **flipout.xml**:

**flipin.xml**

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

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

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

<translate

android:duration="500"

android:fromXDelta="-100%"

android:toXDelta="0%" />

</set>

**flipout.xml**

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

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

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

<translate

android:duration="500"

android:fromXDelta="0%"

android:toXDelta="100%" />

</set>

Созданные анимации можно присоединить через атрибуты **ViewFlipper**:

<ViewFlipper

...

android:inAnimation="@anim/flipin"

android:outAnimation="@anim/flipout" >

Мы зададим анимацию программно, добавив несколько строчек кода к предыдущему коду

Animation animationFlipIn = AnimationUtils.loadAnimation(this,

R.anim.flipin);

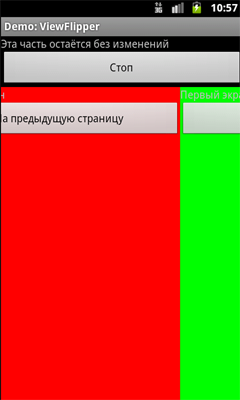
Animation animationFlipOut = AnimationUtils.loadAnimation(this,

R.anim.flipout);

flipper.setInAnimation(animationFlipIn);

flipper.setOutAnimation(animationFlipOut);

Запустите пример и нажмите на кнопку запуска слайд-шоу. Теперь смена слайдов будет происходит красиво. На скриншоте можно наблюдать переходный процесс смены слайдов.



**Дополнительное чтение**

[ViewFlipper. Продвинутые приёмы](http://developer.alexanderklimov.ru/android/views/viewflipper4.php) (Закрытая зона/3-й месяц)

[ViewFlipper. Управление жестами](http://developer.alexanderklimov.ru/android/views/viewflipper2.php) (Закрытая зона/3-й месяц)

[ViewFlipper. 3D Flip](http://developer.alexanderklimov.ru/android/views/viewflipper3.php) (Закрытая зона/3-й месяц)