**Sepia имитация старых фотографий**

В графических редакторах часто присутствует эффект Sepia, обычно применяемый для имитации старых фотографий. О том, что такое **Сепия**, читайте в [Википедии](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BF%D0%B8%D1%8F). Обратите внимание, что даже в Википедии для иллюстрации термина используют фотографии кошек. Традиционным считается коричневый цвет для сепии, но это не обязательно.

Реализуем сепию на Android. За основу возьмём [предыдущий пример](http://developer.alexanderklimov.ru/android/catshop/flip.php).

Перейдем к коду.

butDemo.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

BitmapDrawable mydrawable = (BitmapDrawable) imageView.getDrawable();

Bitmap b = mydrawable.getBitmap();

b = createSepiaEffect(b, 1, 1.8, 2.0, 1.5);

imageView.setImageBitmap(b);

}

});

public static Bitmap createSepiaEffect(Bitmap src, int depth, double red,

double green, double blue) {

// размер изображения 1098

int width = src.getWidth();

int height = src.getHeight();

// создаем итоговое изображение

Bitmap bmOut = Bitmap.createBitmap(width, height, src.getConfig());

// константы для оттенков серого

final double GS\_RED = 0.3;

final double GS\_GREEN = 0.59;

final double GS\_BLUE = 0.11;

// информация о цвете точки

int A, R, G, B;

int pixel;

// проходим через все точки изображения

for(int x = 0; x < width; ++x) {

for(int y = 0; y < height; ++y) {

// получаем цвет точки

pixel = src.getPixel(x, y);

// получаем составляющие цвета точки

A = Color.alpha(pixel);

R = Color.red(pixel);

G = Color.green(pixel);

B = Color.blue(pixel);

// переводим в оттенки серого

B = G = R = (int)(GS\_RED \* R + GS\_GREEN \* G + GS\_BLUE \* B);

// устанавливаем уровень интенсивности для каждого цветового канала

R += (depth \* red);

if(R > 255) { R = 255; }

G += (depth \* green);

if(G > 255) { G = 255; }

B += (depth \* blue);

if(B > 255) { B = 255; }

// set new pixel color to output image

bmOut.setPixel(x, y, Color.argb(A, R, G, B));

}

}

// возвращаем финальное изображение

return bmOut;

}

Можете самостоятельно поиграться с различными значениями для параметров цвета. Для удобства я вывел используемые значения в заголовок программы.

Идея статьи: [Photography Sepia-toning Effect](http://xjaphx.wordpress.com/2011/06/21/image-processing-photography-sepia-toning-effect/" \l "1098)

   

**Р**