**Рисуем многоугольники**

Будем учиться рисовать правильные многоугольники.

Создадим новый класс, наследующий от **View** и начнём экспериментировать.

package ru.alexanderklimov.simplepaint;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.Path;

import android.view.View;

public class Draw2D extends View {

float w, h, cx, cy, radius;

Paint paint;

public Draw2D(Context context) {

super(context);

init();

}

private void init() {

paint = new Paint();

}

@Override

protected void onDraw(Canvas canvas) {

int width = 480, height = 600;

paint.setColor(Color.RED);

canvas.drawPaint(paint);

paint.setColor(Color.BLUE);

canvas.drawPath(createPolygon(6, 150, width / 2, height / 2), paint);

}

public static Path createPolygon(int corners, int radius, int x, int y) {

if (corners < 3) {

corners = 3; // многоугольников с двумя углами не бывает

}

float to\_x, to\_y;

Path path = new Path();

int angle = 360 / corners;

path.moveTo(x + radius, y);

for (int j = 1; j < corners; j++) {

to\_x = (float) (x + radius \* Math.cos(Math.toRadians(j \* angle)));

to\_y = (float) (y + radius \* Math.sin(Math.toRadians(j \* angle)));

path.lineTo(to\_x, to\_y);

}

return path;

}

}

Подключаем класс в методе **setContentView()**:

Draw2D d = new Draw2D(this);

setContentView(d);

Многоугольник создаётся в методе **createPolygon()**, которому нужно передать нужное число углов. Минимальное число углов у многоугольника равно трём, поэтому при меньшем значении принудительно устанавливаем допустимое значение. Далее сплошная математика - вычисляем нужные координаты через косинусы/синусы и определяем новую точку вершины фигуры. На скриншотах приведены примеры многоугольников от трех до шести углов. Далее фигуры всё больше стремится к окружности и уже не так интересно смотрятся.

   