## Задание №2

Постановка задачи:

В следующих заданиях создайте тестовое приложение для демонстрации вашего решения, при этом:

* Для изображения указанной в задании фигуры создать класс, реализующий интерфейс Shape;
* Создать указанный фильтр изображения; при тестировании выведите фигуру без фильтра и с фильтром (аналогично фильтрам из примеров);
* Моделируйте освещение и тень от объекта при помощи альфа-канала и/или механизма обработки изображения;
* При рисовании используйте сглаживание, внеэкранный буфер и преобразования координат.

Вариант 13.

Фигура (дорожный знак): надпись STOP в прямоугольнике, цвет прямоугольника и надписи – красный, цвет фона – серый с градиентной заливкой снизу-вверх. Фильтр: Rotate CW degrees (поворот по часовой стрелке на 45).



Рисунок 2 – Знак до и после примененных фильтров

Особенности реализации:

Для реализации использован интерфейс Shape, в методе paint() которого осуществляется требуемая заливка:

gf.setPaint(new GradientPaint((int)rectangle.getX(),(int)rectangle.getY() + height,new Color(50, 50, 50),  
 (int)rectangle.getX(),(int)rectangle.getY(),new Color(150, 150, 150)));  
gf.fillRect((int)rectangle.getX(),(int)rectangle.getY(),200,150);

В методе main() задаются основные действия над фигурами, например, прорисовка тени изображения с помощью аффинновых преобразований до применения фильтра:

g2d.setPaint(new Color(0, 0, 0, 75));  
g2d.translate(75, 75);  
AffineTransform shadow = AffineTransform.getShearInstance(-1.0, 0.0);  
shadow.scale(1.0, 0.5);  
g2d.fill(shadow.createTransformedShape(new Rectangle2D.Double(X+20, Y+20, width, height)));  
g2d.setTransform(save);  
roadsign.paint(g2d);

И после:

img.setPaint(new Color(0, 0, 0, 75));  
img.translate(75, 75);  
shadow = AffineTransform.getShearInstance(-1.0, 0.0);  
shadow.scale(1.0, 0.5);  
img.fill(shadow.createTransformedShape(new Rectangle2D.Double(X, Y, width, height)));

Выполнение поворота изображения:

AffineTransform rotate = AffineTransform.getRotateInstance(Math.PI / 4,100,250);  
AffineTransformOp filter = new AffineTransformOp(rotate, AffineTransformOp.TYPE\_BILINEAR);

Использование BufferedImage:

BufferedImage bimage = new BufferedImage(width + X + 70, height + Y, BufferedImage.TYPE\_INT\_ARGB);  
Graphics2D img = bimage.createGraphics();

Для обработки изображения BufferedImage() передаются метода filter().