Project 1

Java 2D Graphics

Overview

In this project you will create 3 simple, binary 25x25 images or your choice and use Java 2D graphic methods to rotate, scale and translate each of the images.

Requirements:

1. Using Netbeans or Eclipse, develop a Java 2D graphics application that creates 3 images. The images should have the following specifications:
    a. Size: 25x25 pixels
    b. Type: binary (consists of ones or zeros)
    c. Simple form or shape (Hint: consider a letter or number, or even simple shapes such as crossing lines, rectangles, or circles
    d. You should generate the image inside of separate methods and store them as 2D arrays.
2. Use Java 2D graphics to display your original images.
3. For each image use the existing Java 2D graphics transformation methods to translate, rotate and scale each object. You should perform the following transformations on each image:
    a. Translate -5 in x direction, Translate +7 in the y direction.
    b. Rotate 45° counter clockwise.
    c. Rotate 90° clockwise
    d. Scale 2 times for the x component, scale 0.5 times for the y component
    e. Each of these transformations should be displayed in sequence with the images always starting from the previous transformation as opposed to the original image.
    f. Use Java 2D graphics to display each transformation for each image. (Hint: you can do this with loop and slight pause for each display)
4. All Java source code should be written using Google Java style guide.
5. Prepare, conduct and document a test plan verifying each method is functioning properly. (Hint: Using JUNIT tests are recommended)

Deliverables:

1. All Java source code used for this project. Code should adhere to the Google Java style guide.
2. Word or PDF file demonstrating with clearly labeled screen captures and associated well-written descriptions, the success execution of your 2D graphics transformation. The document should be well-written, well-organized, include page numbers, captions for all screen captures, and a title page including your name, class, section number and date. References should be included for all sources used and formatted in APA style.

Grading guidelines:

Attribute	Meets
Design	**20 points** Methods used to isolate functionality (10 points)

	Code is efficient without sacrificing readability and understanding. (5 points) Code can easily be used and maintained. (5 points)
Functionality	**50 points** Creates Simple, 25x25 binary 3 images (5 points) Stores images as 2D arrays. (3 points) Uses Java 2D graphics to display your original images. (2 points) Translates each image -5 in x direction. (5 points) Translates each image +7 in the y direction. (5 points) Rotates each image 45° counter clockwise. (5 points) Rotates each image 90° clockwise. (5 points) Scales each image 2 times for the x component (5 points) Scales each image 0.5 times for the y component. (5 points) Displays transformations in sequence. (5 points) Uses Java 2D graphics to display each transformation for each image. (5 points)
Testing	**10 points** Prepares, conducts and documents a test plan verifying each method is functioning properly. (10 points)
Documentation and deliverables	**20 points** Submits all Java source code used for this project. (5 points) Code adheres to the Google Java style guide. (5 points) Submits Word or PDF file demonstrating with clearly labeled screen captures and associated well-written descriptions, the success execution of your 2D graphics transformation. (5 points) The document is well-written, well-organized, includes page numbers, captions for all screen captures, and a title page including your name, class, section number and date. References are included for all sources used and formatted in APA style. (5 points)