`StampIt.java`

1 Purpose

This assignment introduces you to the acm.graphics package, which we will be using to create graphics and animation. You'll see an example and then modify it, using some math to create shapes on the screen where the user clicks.

2 Program Description

Modify the StampIt.java file to create a program that draws graphics of your choice on the screen as the user clicks.

Your program must:

- produce graphical output that is different than the example provided.
- center your graphic at the x,y point where the user clicked.
- show something different depending on which quadrant the mouse click is in.
- prevent your shape from going off the edge of the screen.

Watch the videos for a demo and some helpful tips.

Read all of the comments in the code you are given.

3 Using named constants

You must create and use at least one named constant for a literal value. Dimensions and colors are good candidates for named constants. Name them using all upper case letters, with words separated by underscore. For example: HEAD_FILL_COLOR

4 Using the java.lang.Math class (optional)

You may wish to use the java.lang.Math class. For example, Math.sin() and Math.cos() Here is a video explaining how to use sin and cos to draw objects tangent to a circle: https://www.youtube.com/watch?v=aHaFwnqH5CU

The Java standard library works in radians. As a review, a circle is 360 degrees, which is the same as 2π radians, where π is the constant value Math.PI. So if you want the cosine of 45 degrees, you would have to convert 45 to radians: Math.cos(Math.PI / 4.0).

More generally, the formula for converting degrees to radians is:

$$radians = degrees * \pi/180$$

5 Experimenting with color, font, etc.

Feel free to change any of the example code given. Netbeans will give you hints (for example, when you type Color. you will get a list of possibilities. To create your own RGB colors, you can create a new Color with numbers for red, green, and blue values. See the video for tips.

6 Experimenting with other shapes

For other things you can do besides rectangles, see the complete reference in Brightspace, under Content→ Help→ Links, or here:

https://cs.stanford.edu/people/eroberts/jtf/javadoc/student/acm/graphics/package-summary.html
Recommended: GOval, GLine, GRect, GPolygon

7 Getting Started: Adding the acm.jar file to your project

(You can see how to do this in the video also). Once you have created your project in Netbeans, and added the StampIt.java file to the src folder, do the following to configure the acm.jar file:

1. Move the acm.jar file into your project folder.
2. In Netbeans, right click your StampIt project in the Projects panel at the top left. Choose "Properties"
3. In the Project Properties dialog that appears, click "Libraries" in the upper left.
4. On the right, click "Add JAR/Folder".
5. Browse for the acm.jar file.
6. Click OK.
7. Now you will be able to successfully build your project.

8 Sharing your project with classmates, family, and friends

Of course you will want to show everyone your cool new program! Here is how to do it:

1. In Netbeans, right click your StampIt project in the Projects panel at the top left. Choose "Properties"
2. In the Project Properties dialog that appears, click Build → Packaging.
3. Make sure all 3 checkboxes are checked.
4. Click OK.

5. Choose Run → Clean and Build Project.

6. Switch to a file/folder window on your computer, and look inside your project folder for the "dist" folder. Check in the "dist" folder to make sure you see StampIt.jar and a "lib" folder. If you do, great. Compress the "dist" folder by right clicking it. On Mac, choose "Compress", and on Windows choose "Send To → Compressed/Zipped Folder". See the video if you have trouble.

9 Style Requirements for all Projects

- Use a multi-line comment at the top of your program that contains the name of your program, what project it is, your name, and the due date. See the Example.java file for an example.
- In your multi-line ("header") comment, please include a sentence pledging that you have upheld the Non-Collaboration Policy. (See the Syllabus for details on the policy).
- Use single line comments to describe what your code is doing. You don't have to comment every line.
- Variable names are camelCase, starting with lowercase
- Variable names are descriptive
- Variables are defined early (near the top of the main() method, where the others are already defined)
- Variables are given an initial, default, value when they are declared. For example: `int amount = -1;`
- Indentation is correct (use Netbeans to do this automatically, with Source → Format)

10 Grading Criteria

Compiles with no syntax errors	10%
Header comment, program description, and pledged statement (see Example.java)	5%
Graphic is drawn when the user clicks	5%
Graphic is centered at the x,y coordinate where the mouse was clicked	10%
Graphic is different than the demo and the example code	10%
Graphic looks different in different screen quadrants	10%
Graphic does not go off the screen when the mouse is clicked at the screen edge	15%
Variables have meaningful names	10%
All variables initialized to default values	10%
Code is indented correctly	5%
Named constants follow naming conventions and are used correctly	5%
Inline comments are used to explain sections of the code	5%

11 Reminder: Project Handin Rules

1. Projects may be turned in up to a week late. You will lose 5% of the grade for each day late.