Addison Correll

Computer Science (4)

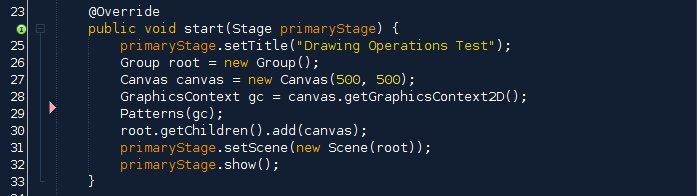
10/8/15

**Methods Madness Essay**

My project, “Correll\_4\_JavaFXintro.java”, creates a grid of different coloured shapes using methods. A for loop that I created will count a number up to 25, in this case *i*, and will check to see if *i* is divisible by a certain number, and will create a shape if the statement is found true. For example, if *i* is divisible by 3, then the *drawGreenTriangle* method is called to create the triangle on the canvas. My canvas is set up like an invisible table, placing all of the shapes in a 5x5 grid. Therefore, up to 25 integers are represented by their shapes on the canvas at a time.

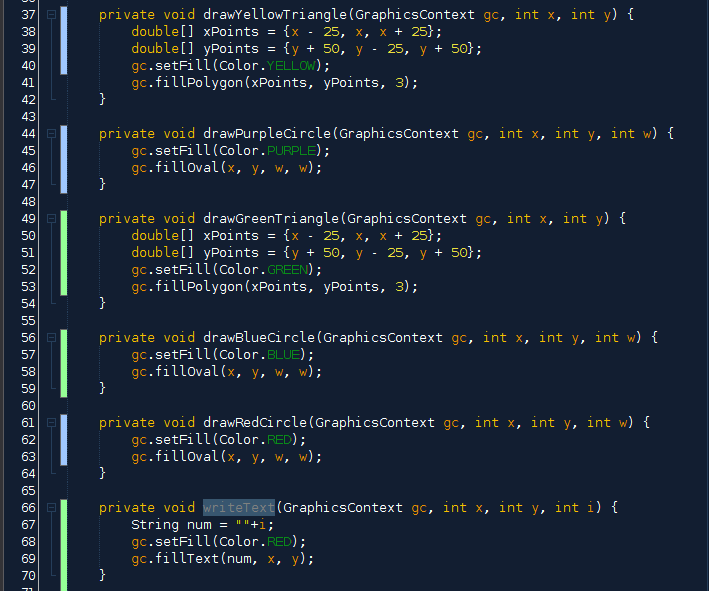
I think that my art is cool and unique because it incorporates the methods from the last project to create the different shapes based on their factors, creating a unique pattern instead of a specific picture.

You can see how my project shows evidence of encapsulation through *private void Patterns(gc);* public method because it is calling upon a private method *private void Patterns(GraphicsContext gc) {*.



correll_4_methodsmadness_pic3_part2.png

Here are all of the methods I used:



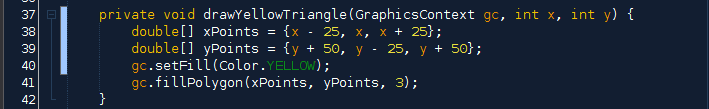
The values passed through each method like *gc.fillOval(x, y, w, w);* are the coordinates that set the location and the size of the circle. I used another 2 for loops to generate the x and y coordinates of the shapes to put them in the correct location on the canvas.

correll_4_methodsmadness_pic2_part3.png

correll_4_methodsmadness_pic3_part4.png

The method *gc.setFill(Color.YELLOW);* passses a color to the shape instead of numerical values like the x and the y coordinates. This method sets the color of the shape created, so in this case *private void drawYellowTriangle(GraphicsContext gc, int x, int y) {* is able to become yellow with this method included.

correll_4_methodsmadness_pic2_part5.png



All of the colors are provided by *import javafx.scene.paint.Color;*.

correll_4_methodsmadness_pic1_part7.png

The canvas is where all of the images are shown, and the canvas grows downward and to the right from (0, 0). My canvas is 500x500, and the canvas is provided by *import javafx.scene.canvas.Canvas;*.

correll_4_methodsmadness_pic1_part8.png

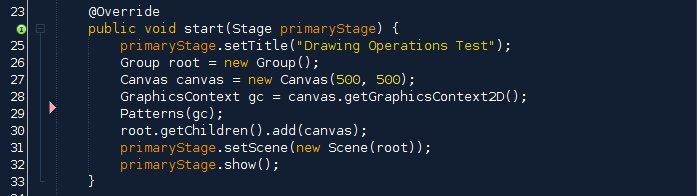
correll_4_methodsmadness_pic1_part9.png

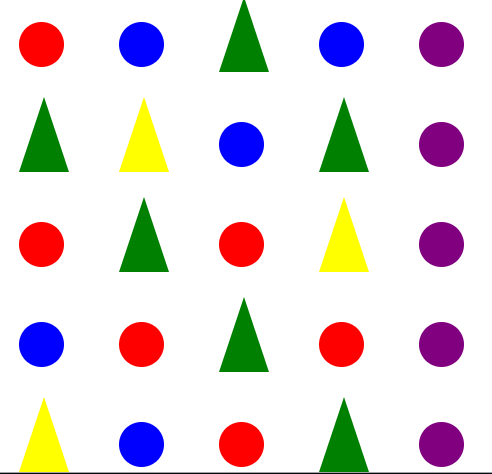
Even though I have more than 5 methods in “Correll\_4\_JavaFXintro.java”, there is only one method included in the main(): *launch(args);*. This is due to the fact that the program is being represented on a canvas, and not through numbers or words.

correll_4_methodsmadness_pic1_part10.png

In the program, I used both public and private access modifiers. An example of this is the private method *private void Patterns(GraphicsContext gc) {* is called upon in the public method *public void start(Stage primaryStage) {*.

correll_4_methodsmadness_pic3_part2.png



Together, my methods created this grid-like structure of shapes to represent 1-25 and their primary factors:

In the end, my project came out very close to what I had originally planned it to be with some few minor tweaks and fixes to get it the way I wanted. I was able to successfully have the for loop for *i* and have the shapes be printed in a 5x5 grid to represent *i*’s factors by the different shapes from the methods. I have learned a lot while creating this project, even though I was quite confused along the way. I now have a better understanding of the process of creating art in JavaFX, and I believe I will be able to do more things now that I have the knowledge to do so.