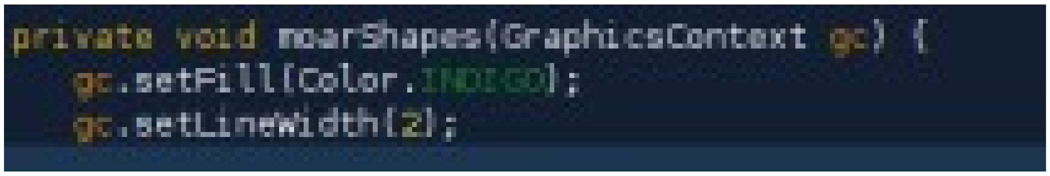
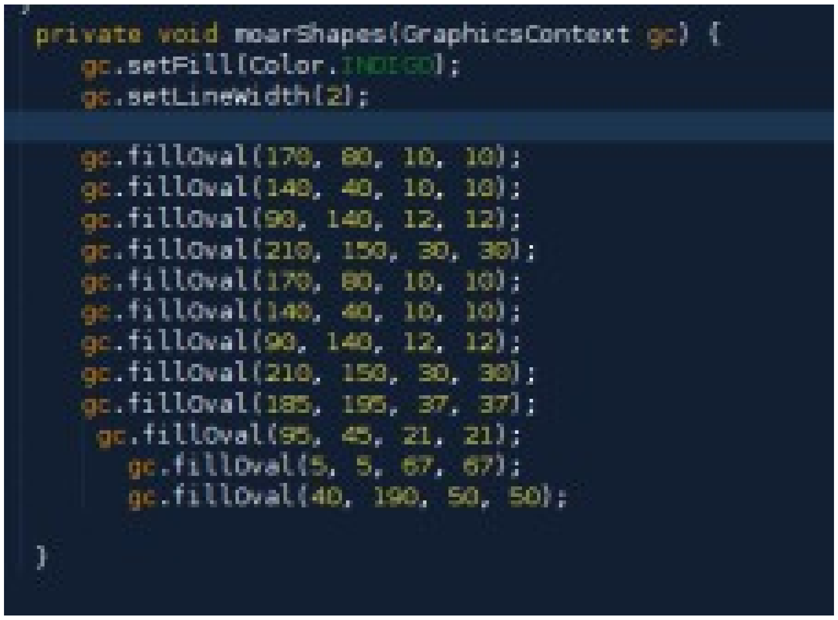
Method Madness

My program “Hernandez_3_challengeone.java” has been given life using the magical gift of methods. The code creates a pattern of blue and purple circles, either filled or empty within (like my soul). The program uses encapsulation through *private void patterns (gc);* public method because it calls upon a private method *private void Patterns (GraphicsContext gc) {*.

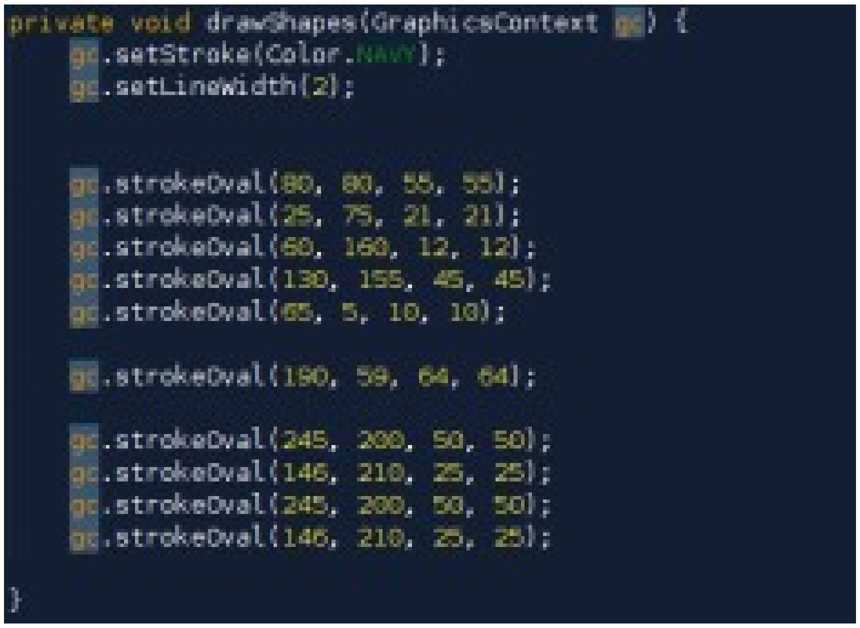


```
private void moarShapes(GraphicsContext gc) {
    gc.setFill(Color.INDIGO);
    gc.setLineWidth(2);
```

The program is organized by the color and fill of the circles.

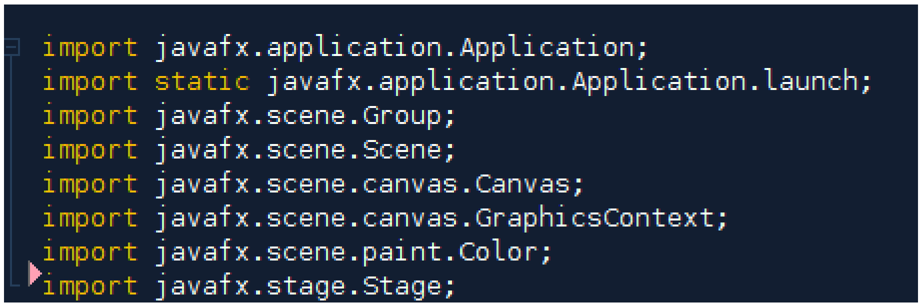


```
private void moarShapes(GraphicsContext gc) {
    gc.setFill(Color.INDIGO);
    gc.setLineWidth(2);

    gc.fillOval(170, 80, 10, 10);
    gc.fillOval(140, 40, 10, 10);
    gc.fillOval(90, 140, 12, 12);
    gc.fillOval(210, 150, 30, 30);
    gc.fillOval(170, 80, 10, 10);
    gc.fillOval(140, 40, 10, 10);
    gc.fillOval(90, 140, 12, 12);
    gc.fillOval(210, 150, 30, 30);
    gc.fillOval(185, 195, 37, 37);
     gc.fillOval(95, 45, 21, 21);
       gc.fillOval(5, 5, 67, 67);
       gc.fillOval(40, 190, 50, 50);

}
```



```
private void drawShapes(GraphicsContext gc) {
    gc.setStroke(Color.NAVY);
    gc.setLineWidth(2);


    gc.strokeOval(80, 80, 55, 55);
    gc.strokeOval(25, 75, 21, 21);
    gc.strokeOval(60, 160, 12, 12);
    gc.strokeOval(130, 155, 45, 45);
    gc.strokeOval(65, 5, 10, 10);

    gc.strokeOval(190, 59, 64, 64);

    gc.strokeOval(245, 200, 50, 50);
    gc.strokeOval(146, 210, 25, 25);
    gc.strokeOval(245, 200, 50, 50);
    gc.strokeOval(146, 210, 25, 25);

}
```

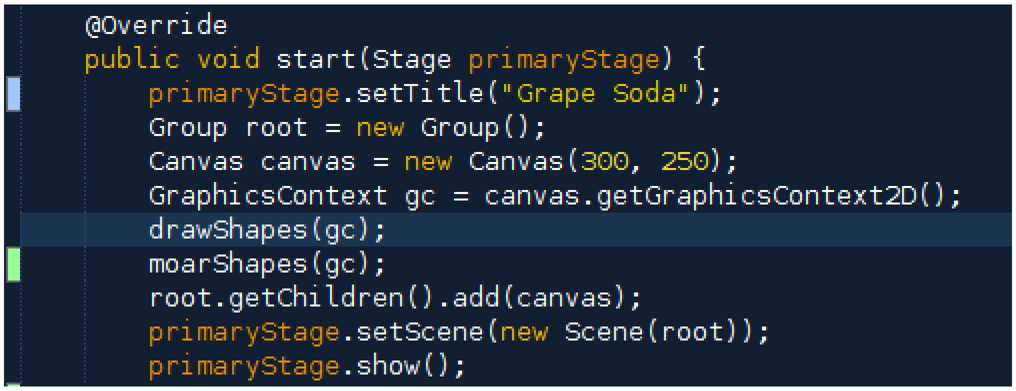
The methods added were given the purpose of separating the color of the circles and whether or not the circles are filled with the color. The values passed are *gc.fillOval(x, y, w, w)* and *gc.strokeOval(x, y, w, w). FillOval* creates ovals that are filled with color, while *strokeOval* creates ovals that only contain colors along the border. X is the horizontal coordinate and Y is the vertical coordinate.W is the radius value of the ovals. For each circle, their W values are the same so they would create a circle instead of an oval. The filled circles are purple while the empty ones are a navy blue.



```
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
```

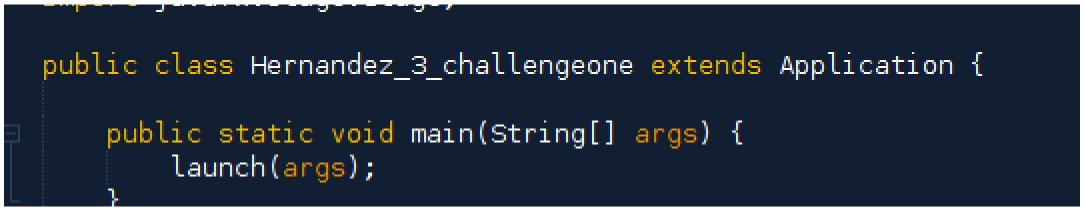
The colors are provided by *import javafx.scene.paint.Color;*

The canvas used to create the artwork is 300X250 in size and is provided by *import javafx.canvas.Canvas.*



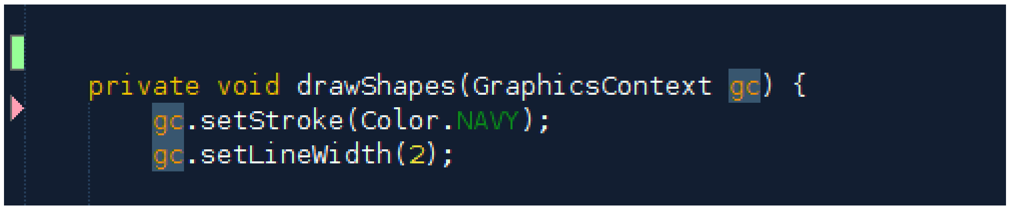
```
    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Grape Soda");
        Group root = new Group();
        Canvas canvas = new Canvas(300, 250);
        GraphicsContext gc = canvas.getGraphicsContext2D();
        drawShapes(gc);
        moarShapes(gc);
        root.getChildren().add(canvas);
        primaryStage.setScene(new Scene(root));
        primaryStage.show();
```

Out of the 3 methods used in the program, only one is in the main(): *launch(args);*. This is primarily because the program is being expressed via canvas.

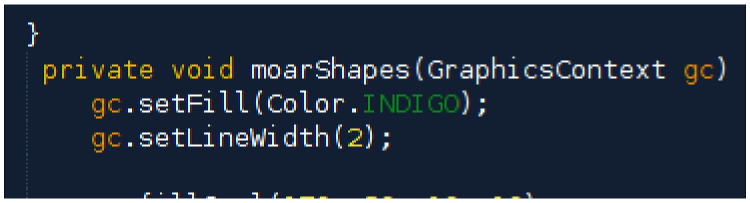


```
public class Hernandez_3_challengeone extends Application {

    public static void main(String[] args) {
        launch(args);
    }
```

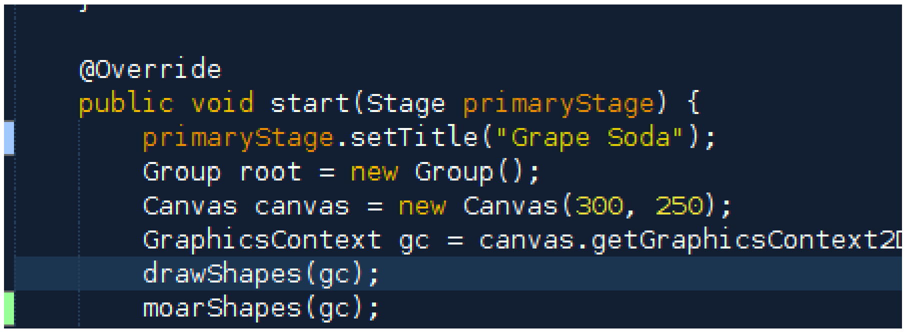
This programs uses both public and private access modifiers. Examples of this are *private void moarShapes(GraphicsContext gc) {* and *private void drawShapes(GraphicsContext gc) {* are both called upon in the public method *public void start(Stage primaryStage) {*.



```
    private void drawShapes(GraphicsContext gc) {
        gc.setStroke(Color.NAVY);
        gc.setLineWidth(2);
```

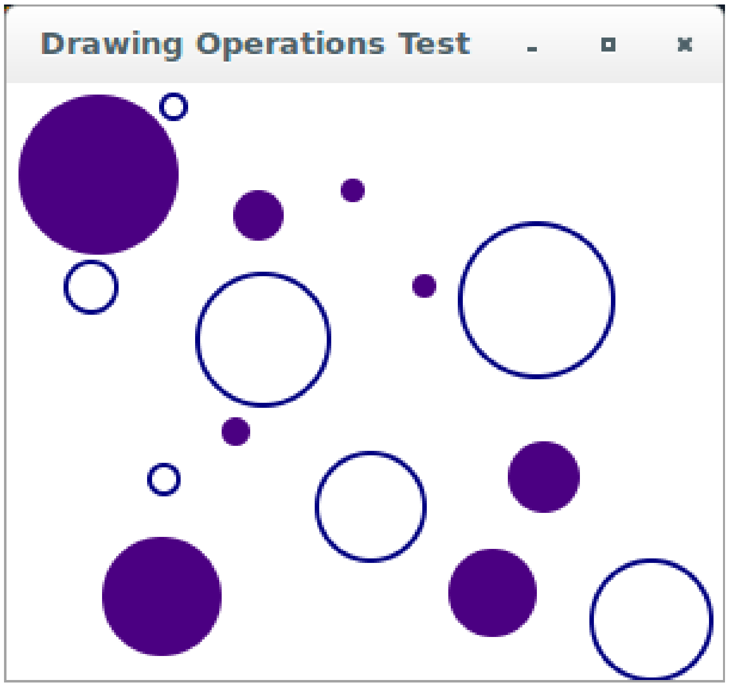


```
}
private void moarShapes(GraphicsContext gc)
   gc.setFill(Color.INDIGO);
   gc.setLineWidth(2);
```



```
    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Grape Soda");
        Group root = new Group();
        Canvas canvas = new Canvas(300, 250);
        GraphicsContext gc = canvas.getGraphicsContext2D
        drawShapes(gc);
        moarShapes(gc);
```

Assembled, these ingredients mixed together to create the bubbliest bubbles your eyes could ever see!



Well, the design is pretty simple but simplicity never killed anyone. This project brought a better understanding of methods and how javaFX works. I feel like I have just dunked my head in the waters of enlightenment, and I can feel the refreshing knowledge seep into my cranium. Something like that. Circles are easy to do, but at least I've grown to learn the ways of canvas and methoding my way through tasks.