



SLTC
Research University

**BSC. (HONS) IN ELECTRONICS AND TELECOMMUNICATIONS
ENGINEERING**

ECS2301-Software Engineering and Project

LAB ASSIGNMENT NO. : 04

INDEX NUMBER : 23UG1- 0152_Akindu Randira

15th FEBRUARY 2025

1. Create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes.

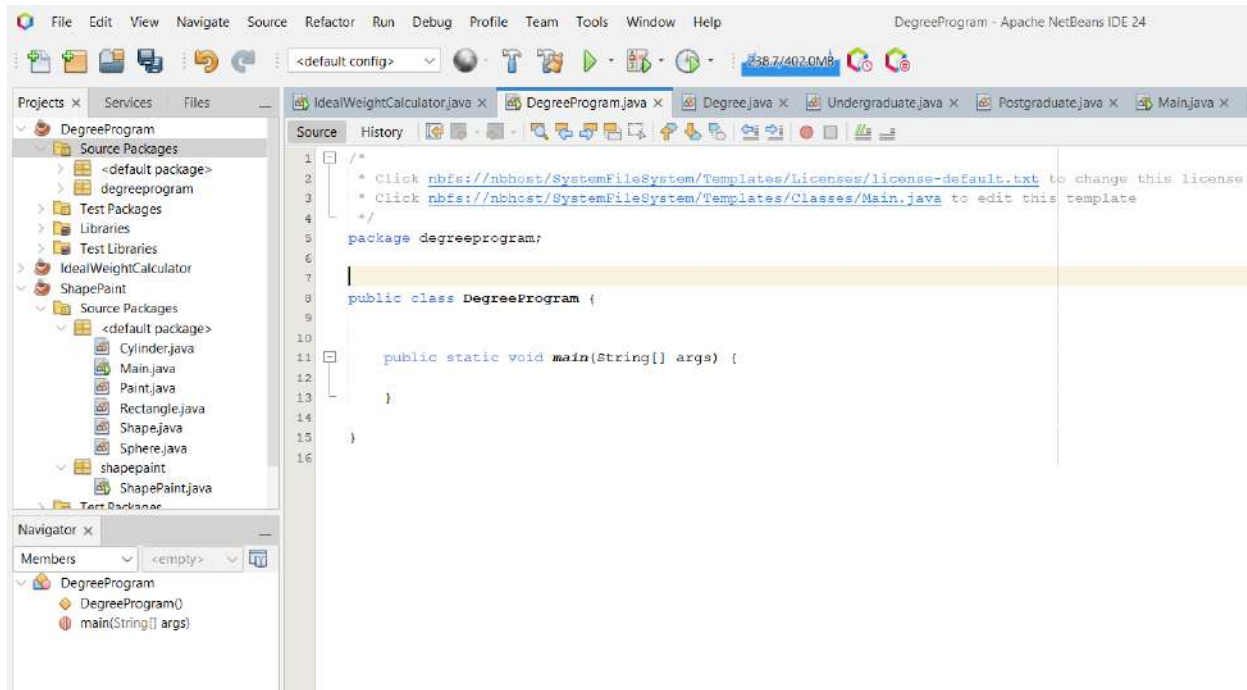


Figure 1 : DegreeProgram Class

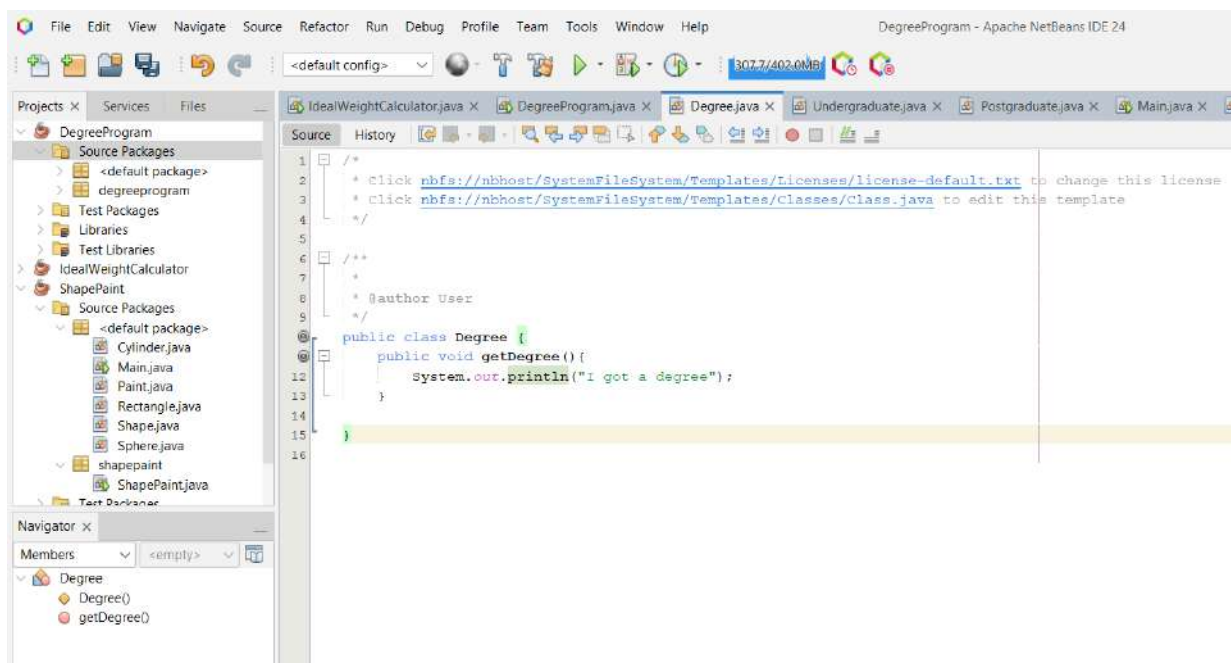


Figure 2 : Degree Class

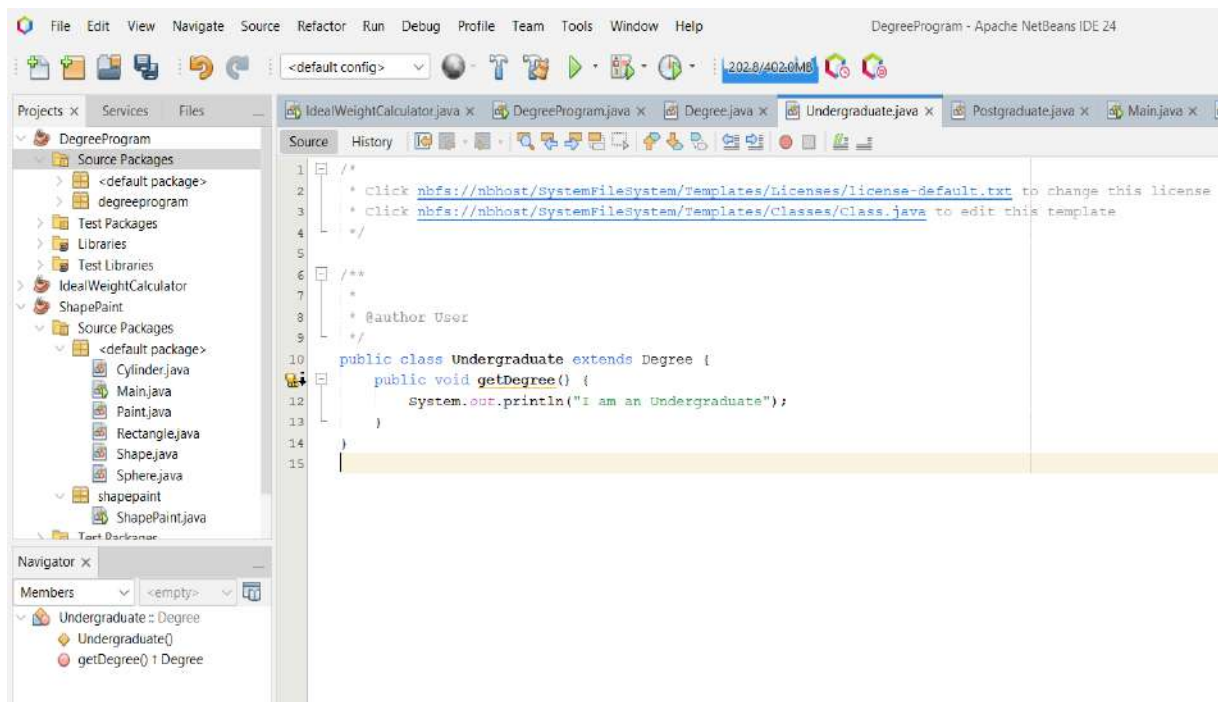


Figure 3 : Undergraduate Class

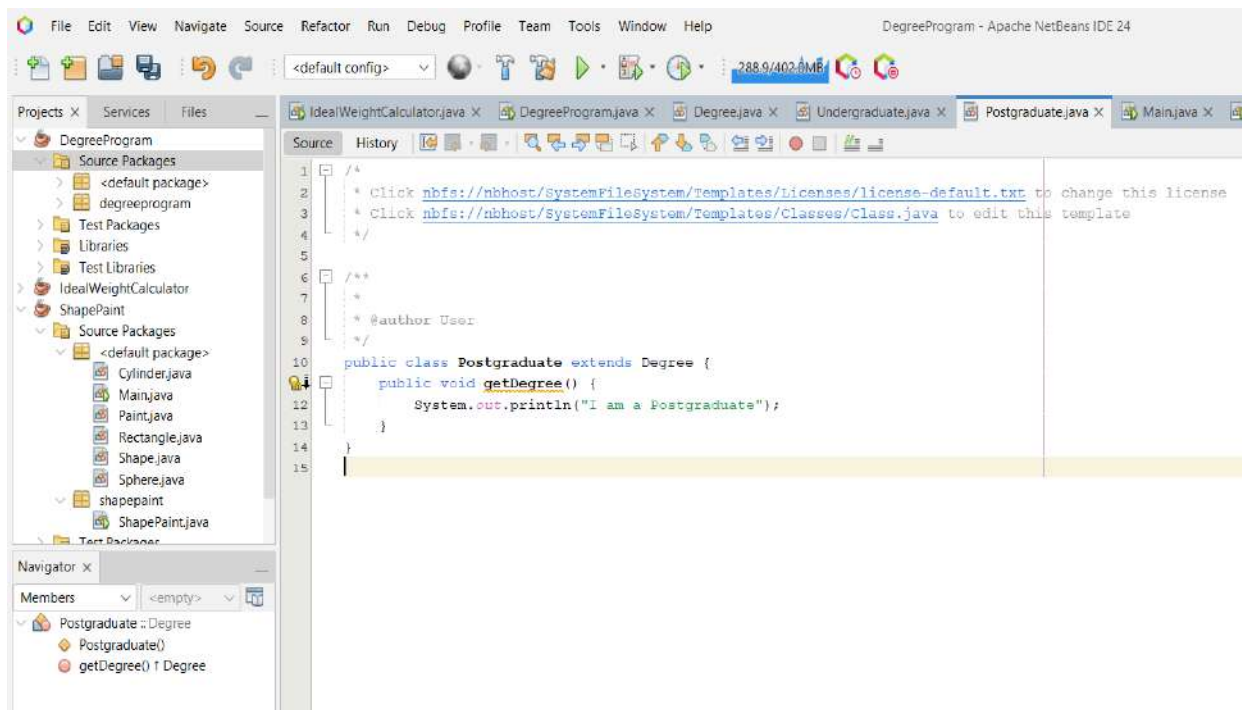


Figure 4 : Postgraduate Class

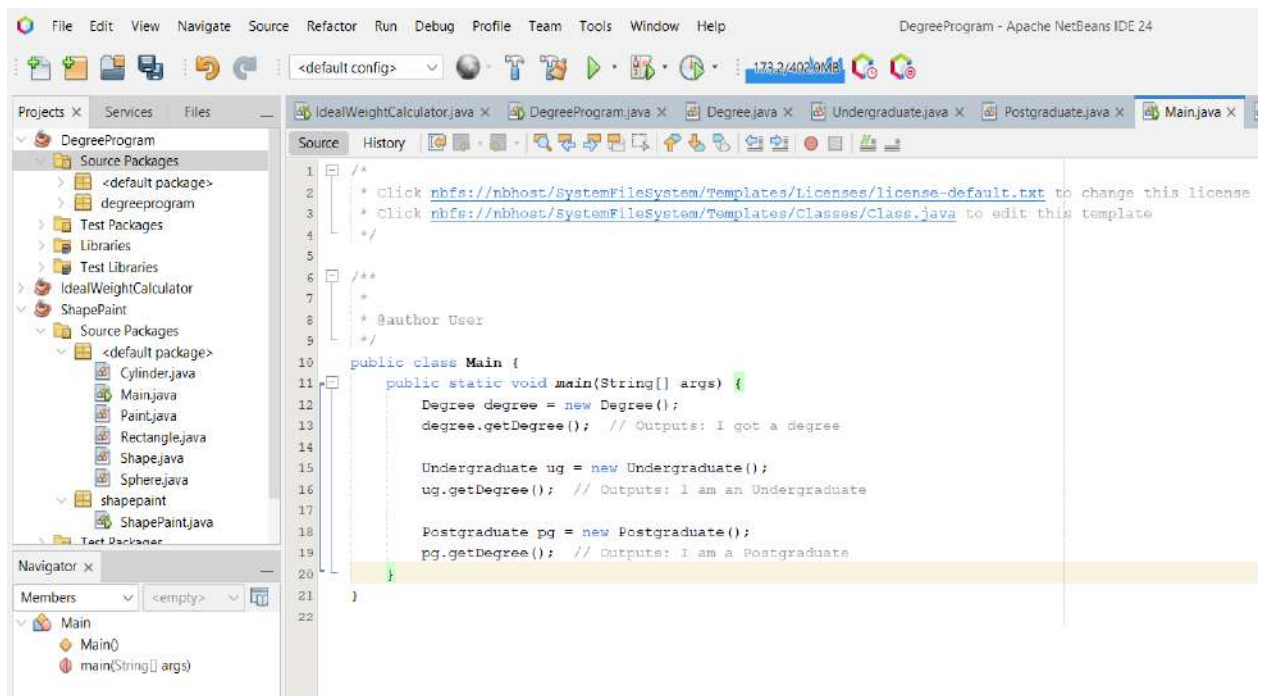


Figure 5 : Main Class

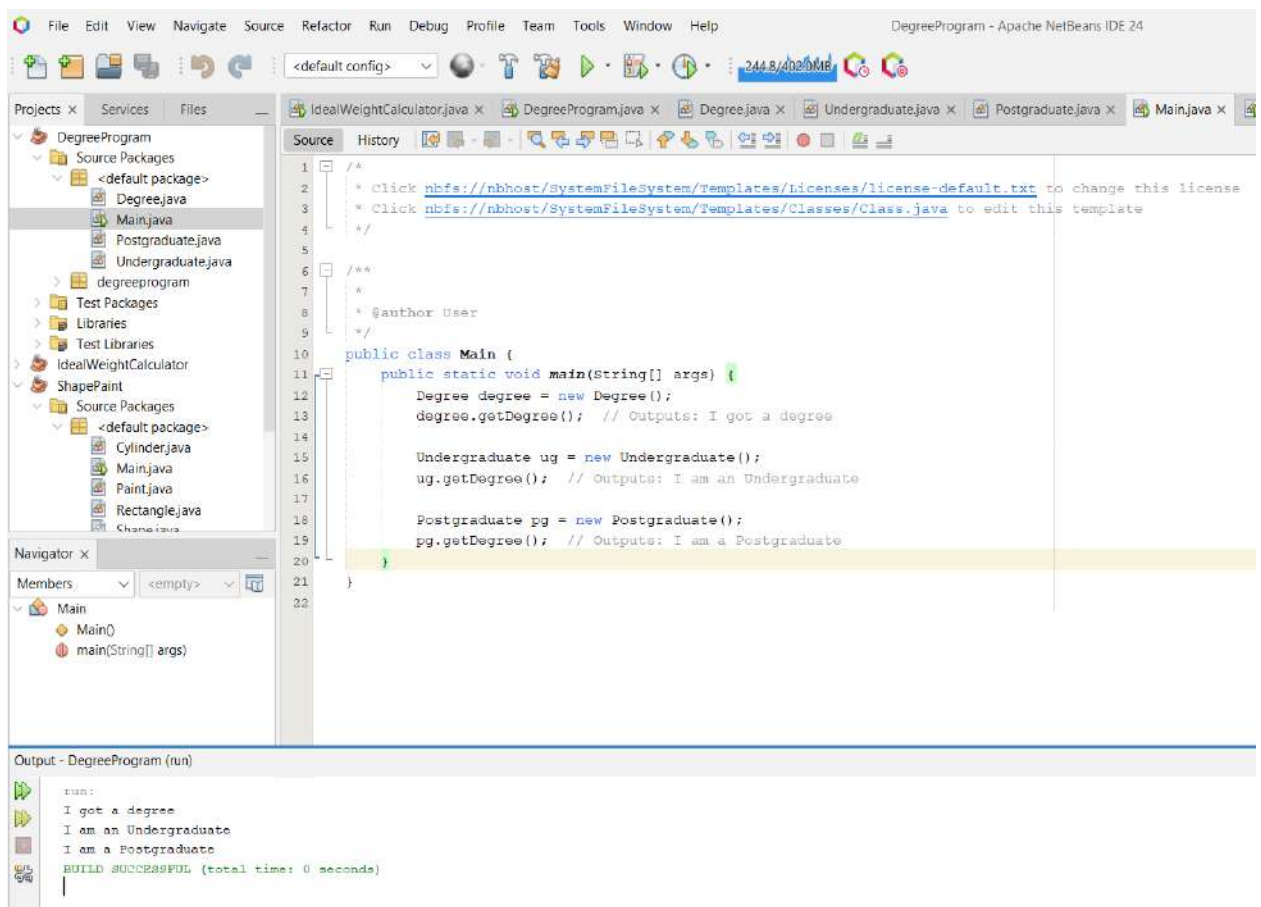


Figure 6 : Output

2. Develop a class hierarchy of shapes and write a program that computes the amount of paint needed to paint different objects. The hierarchy will consist of a parent class Shape with three derived classes - Sphere, Rectangle, and Cylinder. For the purposes of this exercise, the only attribute a shape will have is a name and the method of interest will be one that computes the area of the shape (surface area in the case of three-dimensional shapes). Do the following.
 - a) Write an abstract class Shape with the following properties:
 - An instance variable shapeName of type String
 - An abstract method area()
 - A toString method that returns the name of the shape

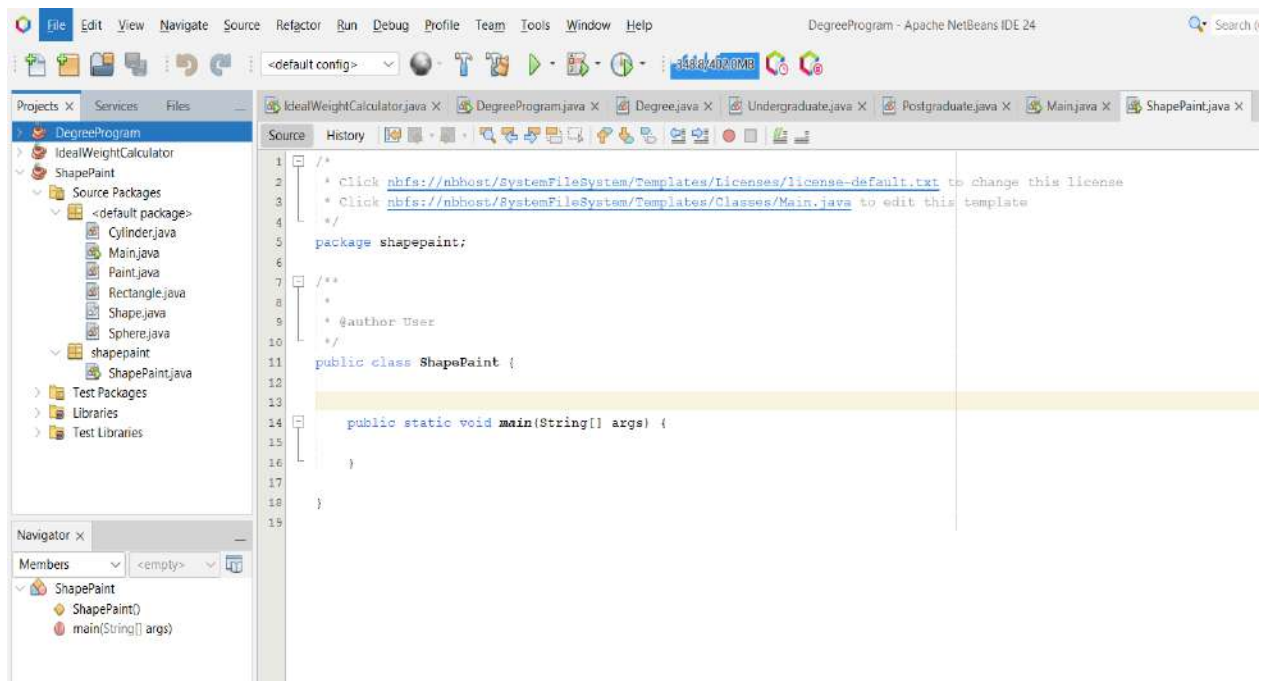


Figure 7 : ShapePaint Class

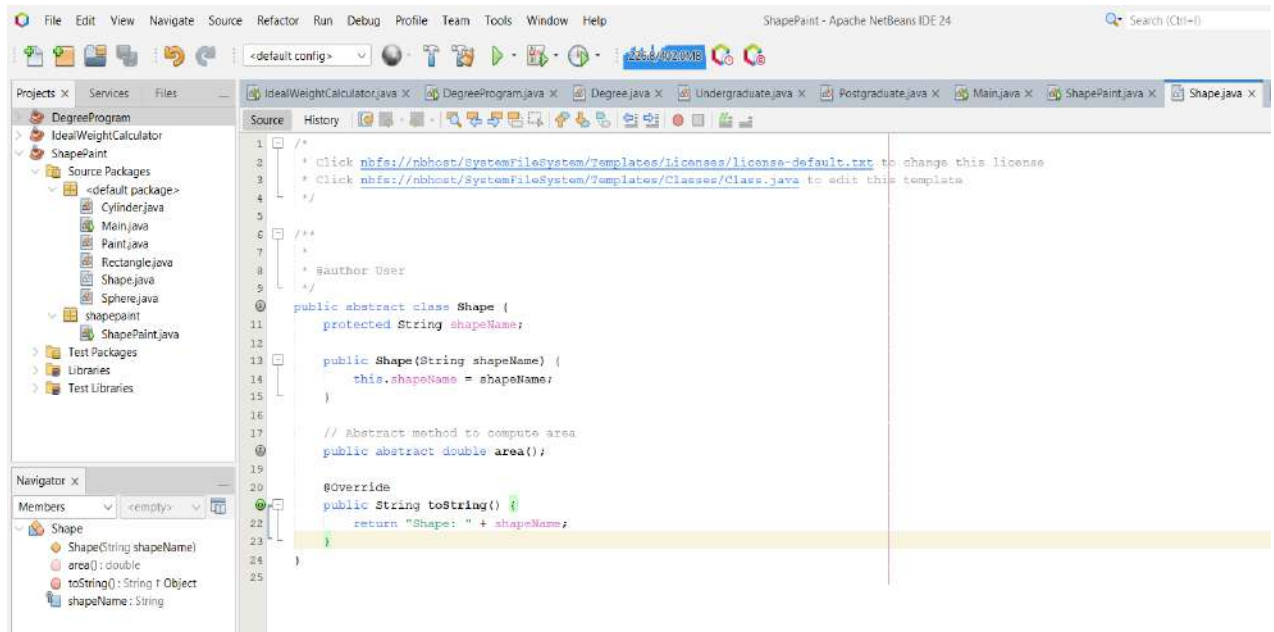


Figure 8 : Shape Class

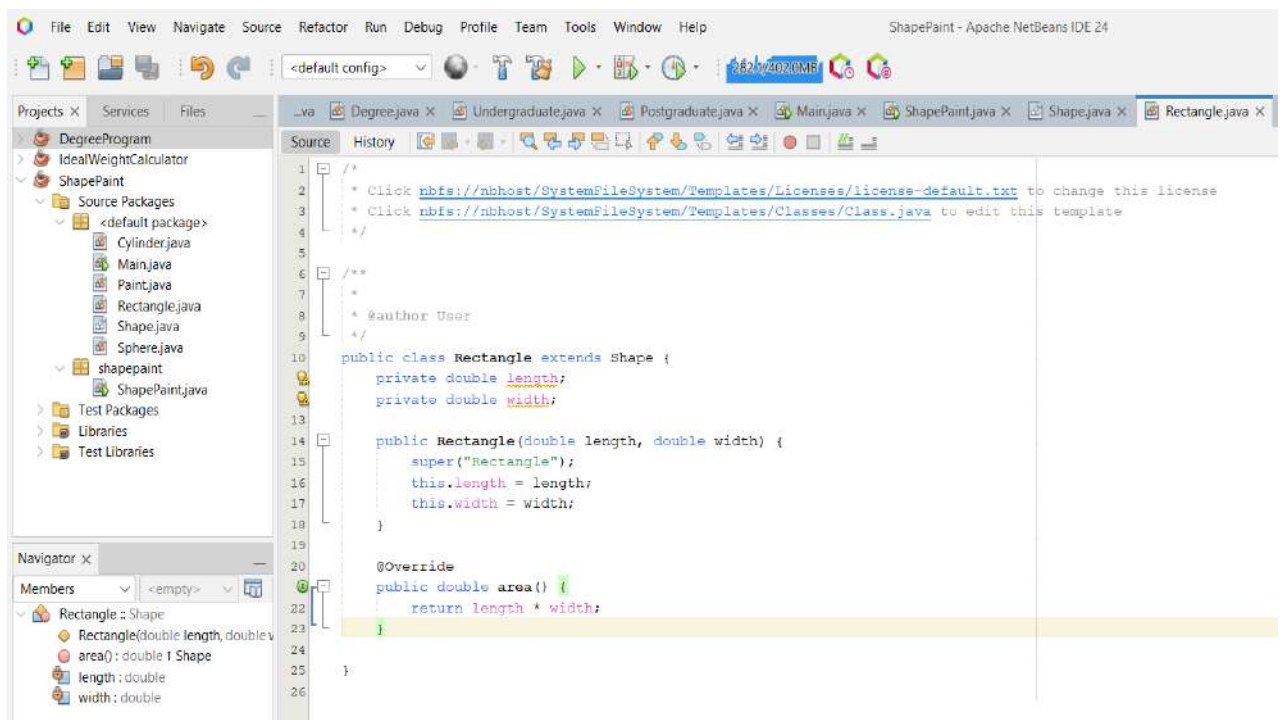


Figure 9 : Rectangle Class

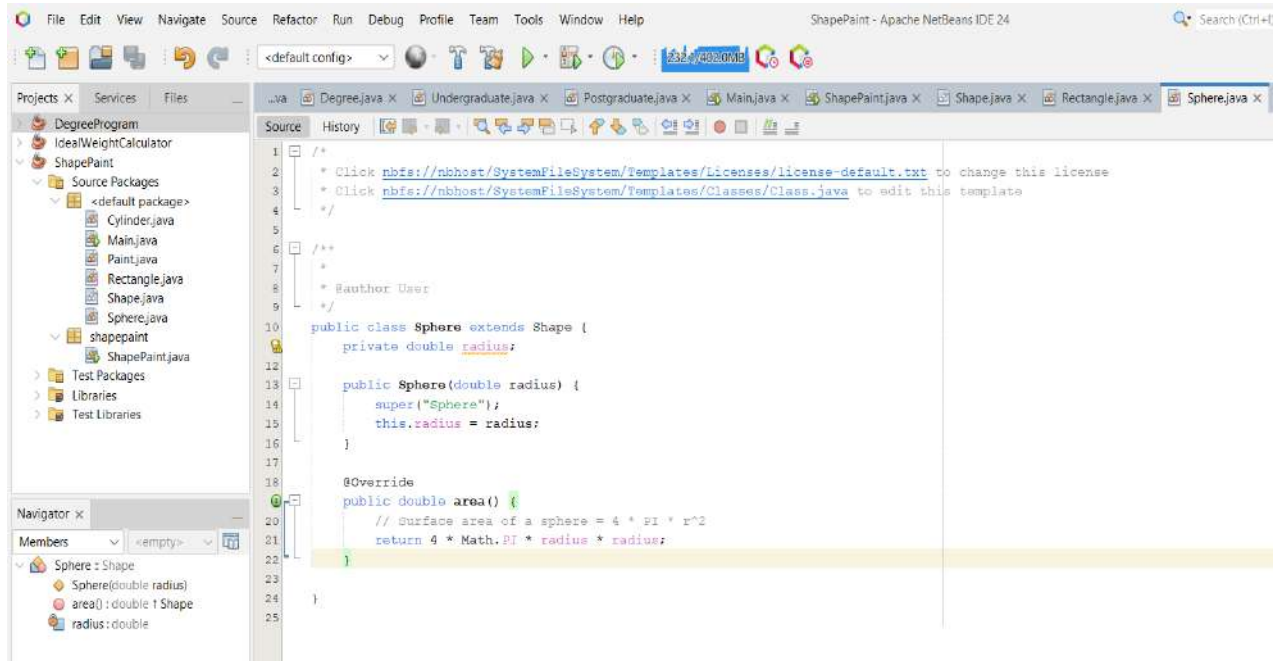


Figure 10 : Sphere Class

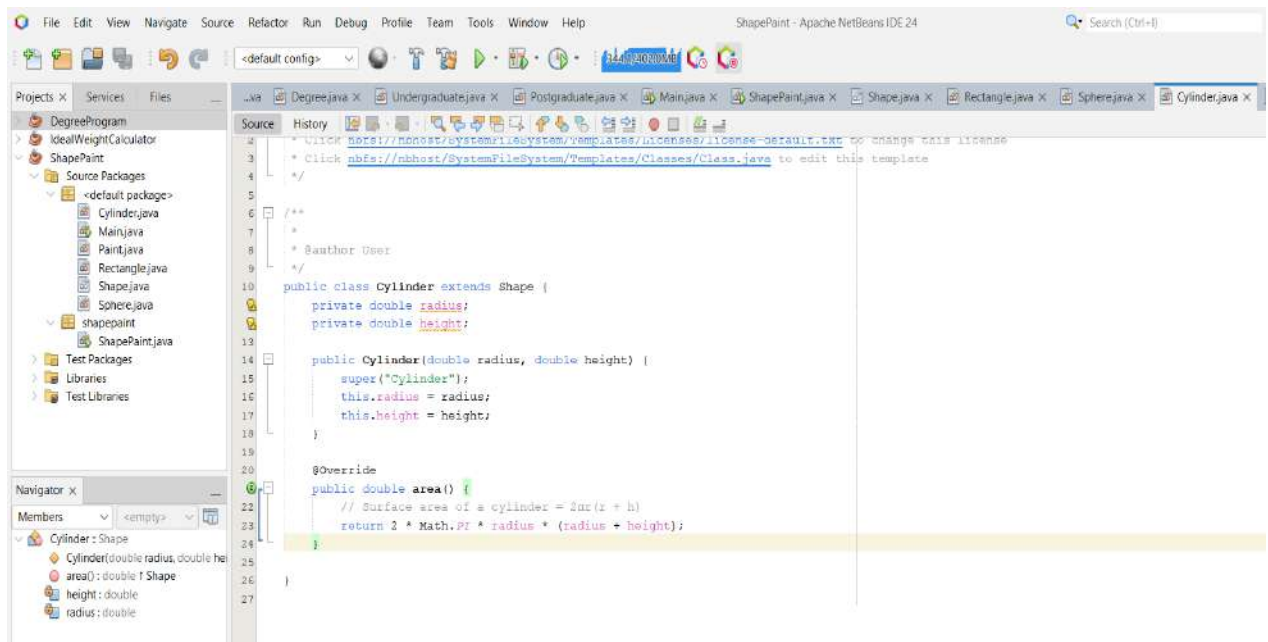


Figure 11 : Cylinder Class

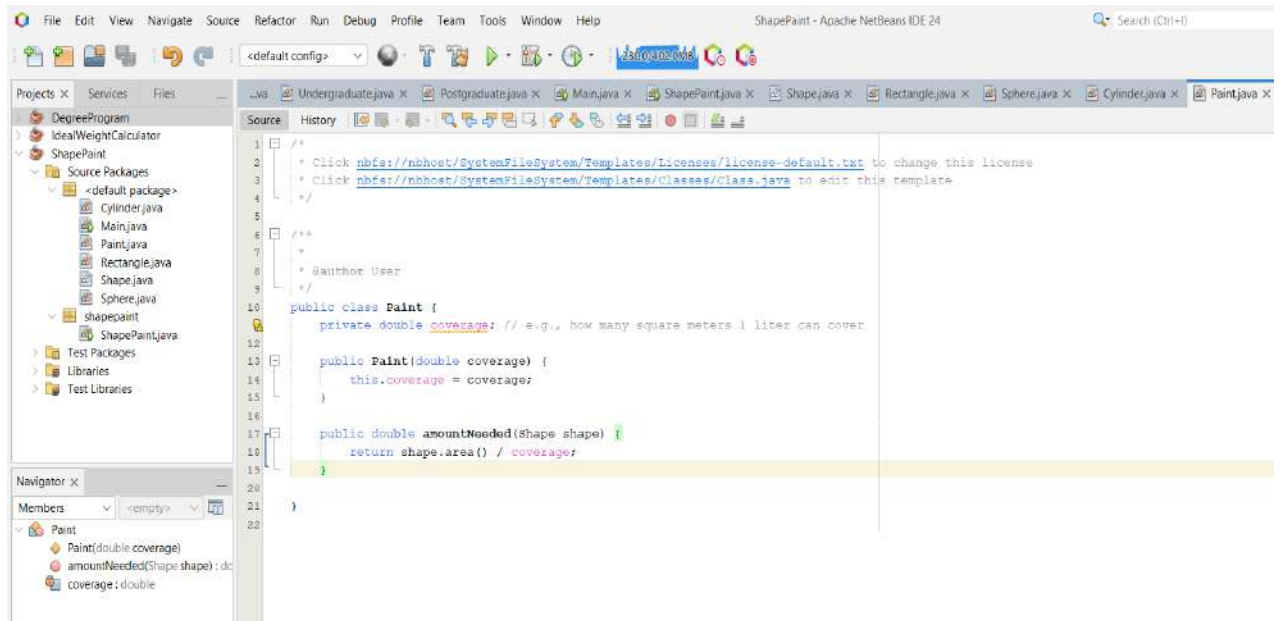


Figure 12 : Paint Class

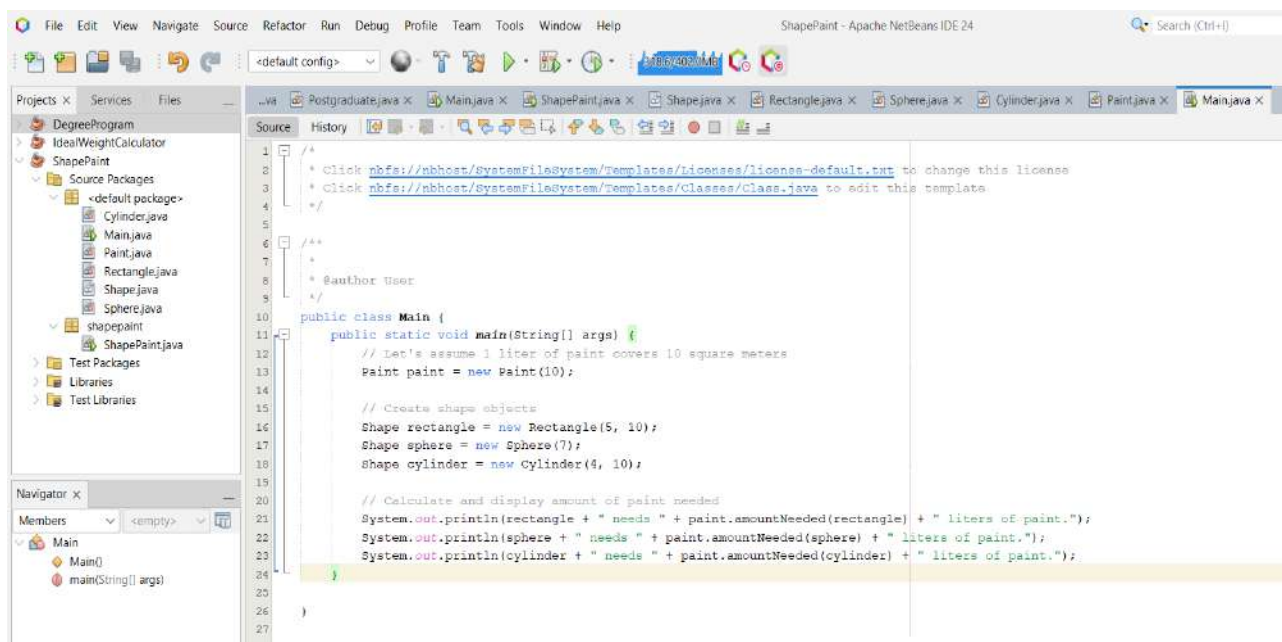


Figure 13 : Main Class

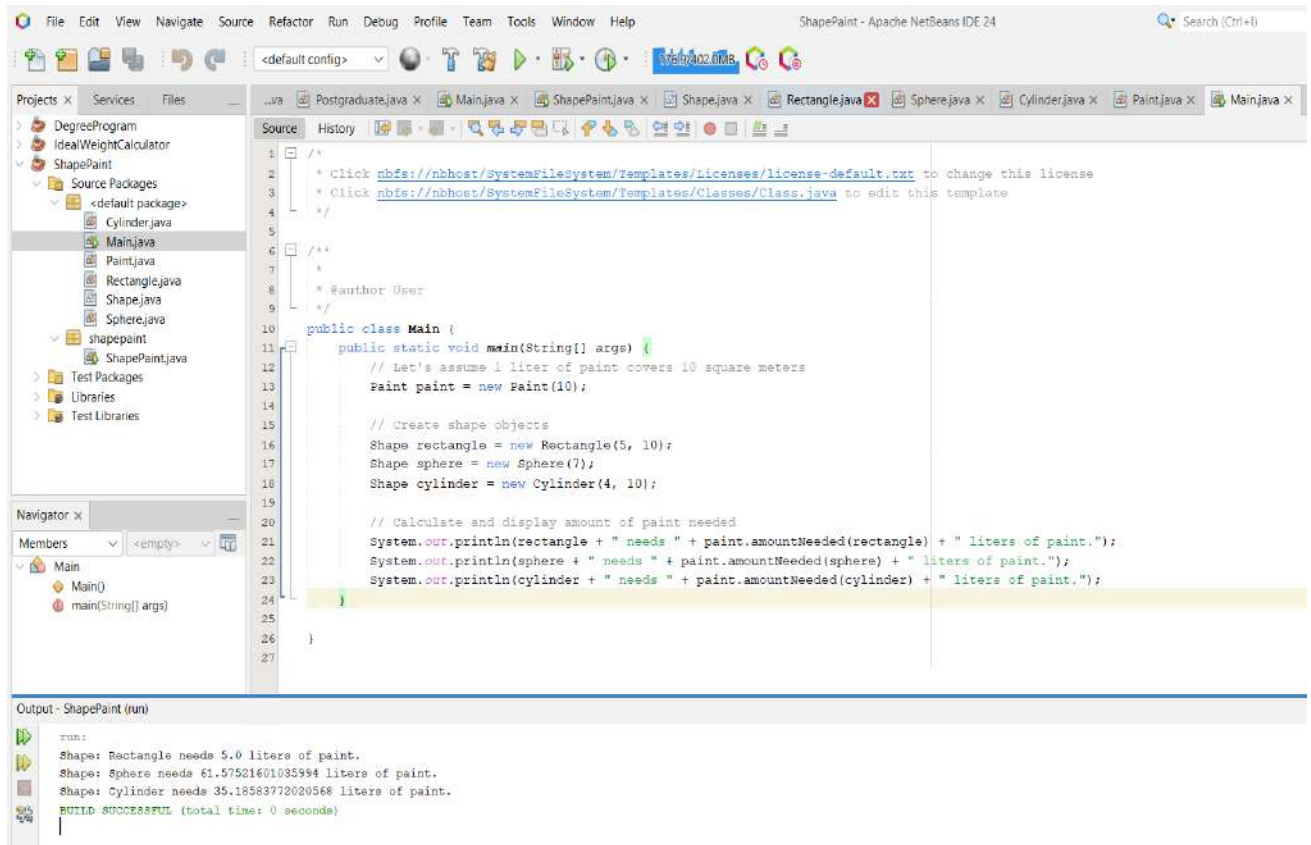


Figure 14 : Output

- b) The file Sphere.java contains a class for a sphere which is a descendant of Shape. A sphere has a radius and its area (surface area) is given by the formula $4\pi \text{radius}^2$. Define similar classes for a rectangle and a cylinder. Both the Rectangle class and the Cylinder class are descendants of the Shape class. A rectangle is defined by its length and width and its area is length times width. A cylinder is defined by a radius and height and its area (surface area) is $\pi \text{radius}^2 \text{height}$. Define the toString method in a way similar to that for the Sphere class.

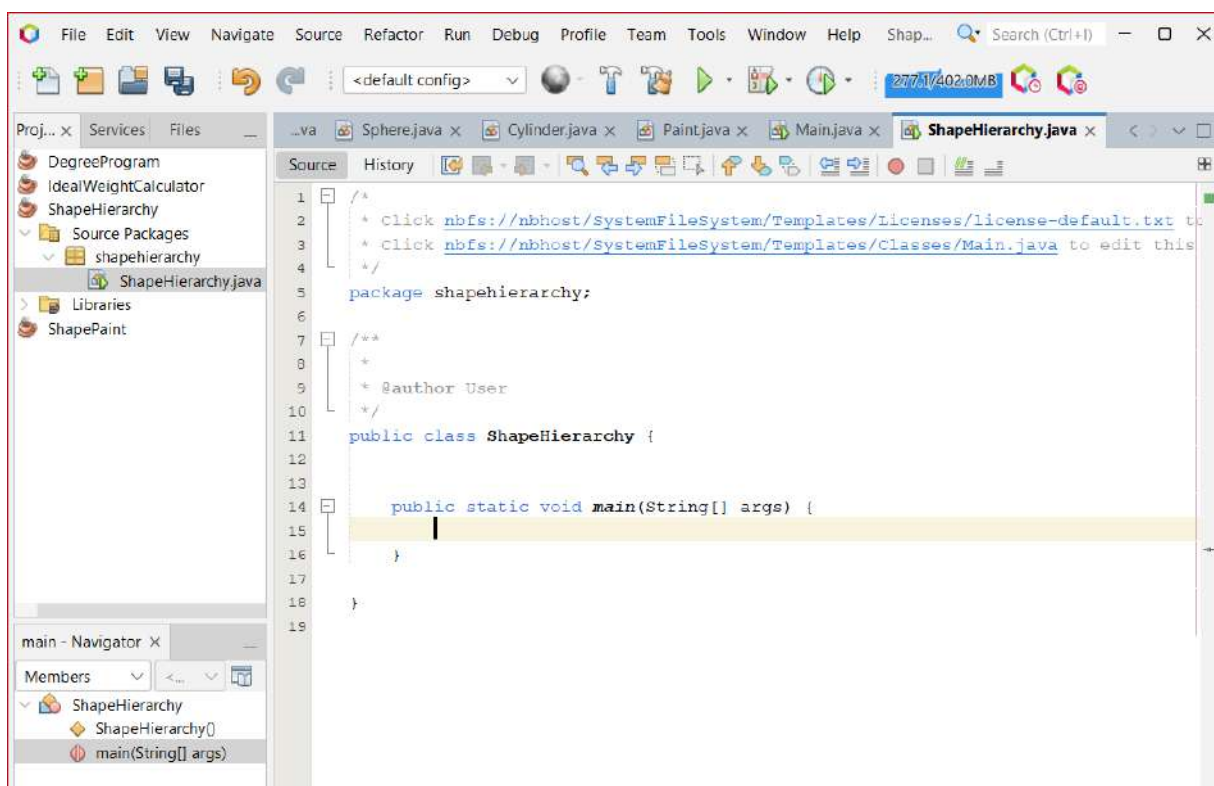


Figure 15 : ShapeHierarchy Class

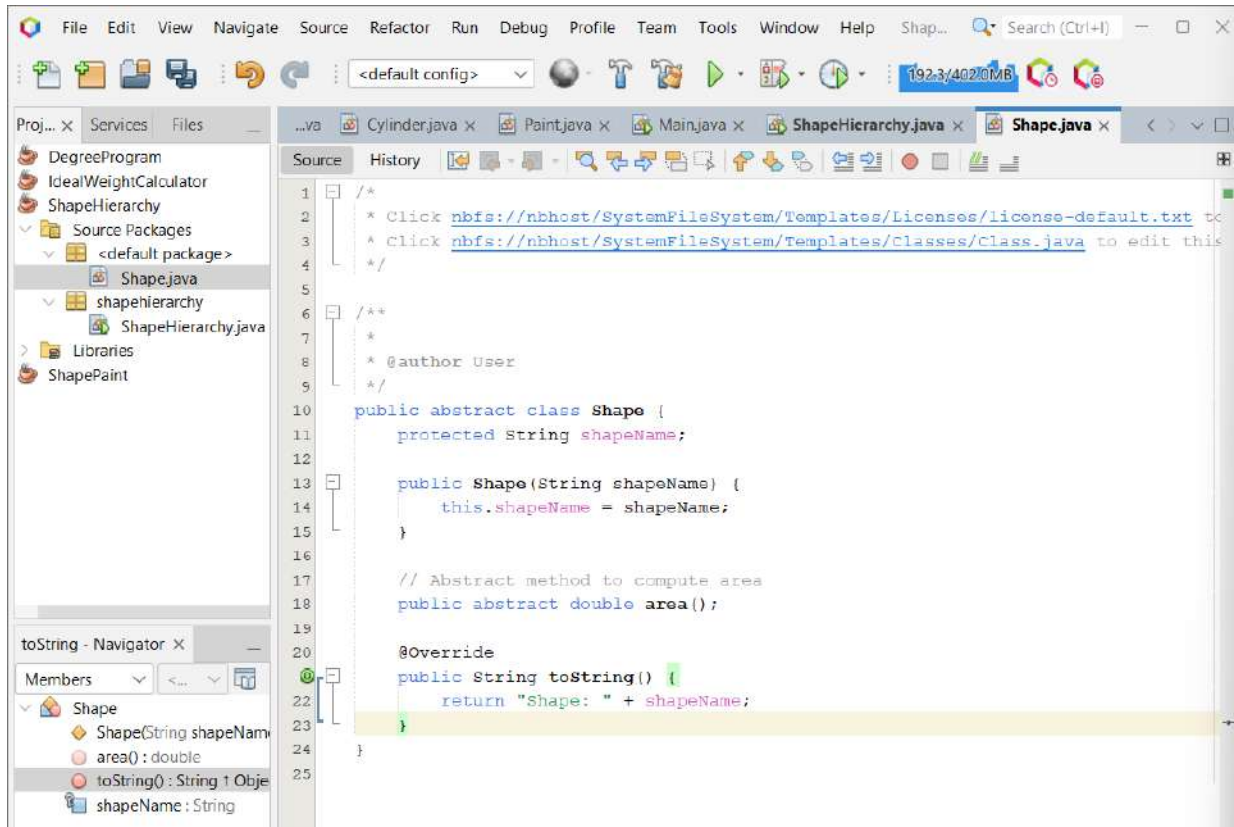


Figure 16 : Shape Class

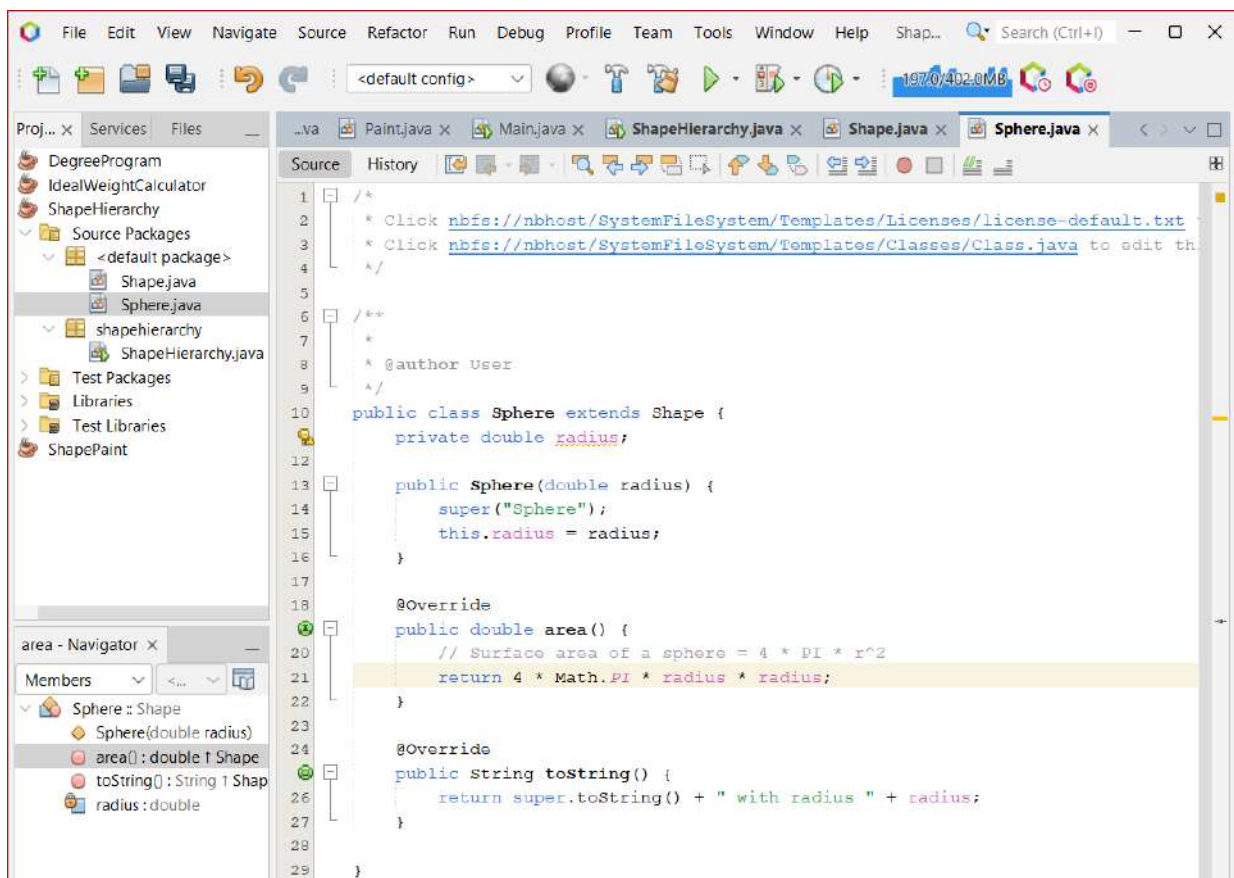


Figure 17 : Sphere Class

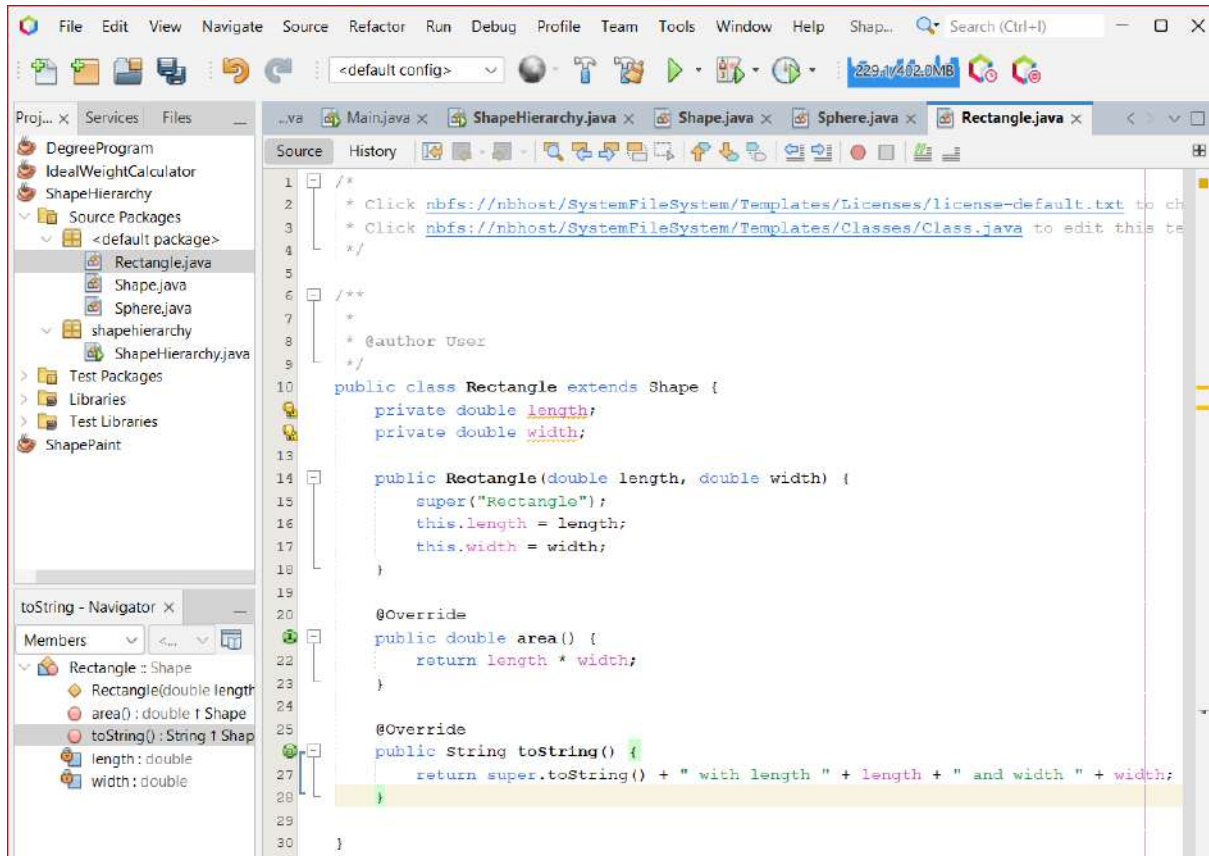


Figure 18 : Rectangle Class

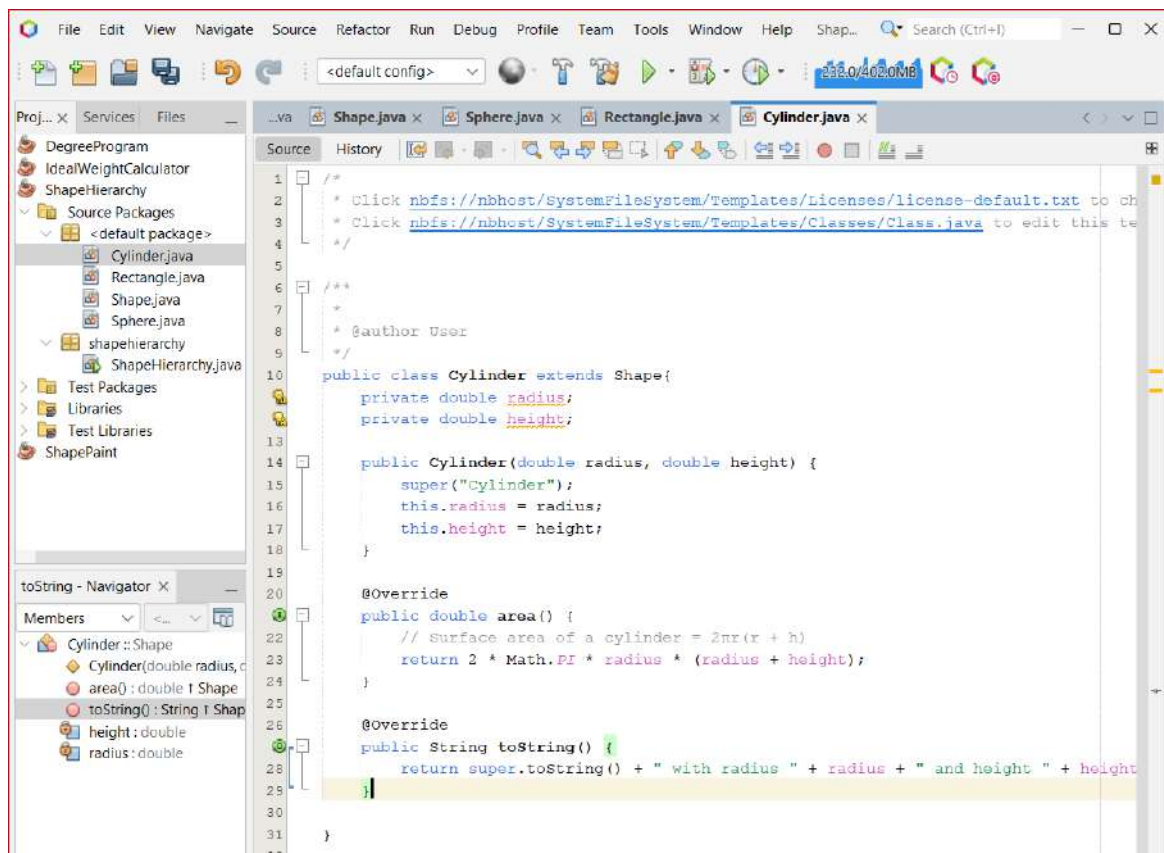


Figure 19 : Cylinder Class

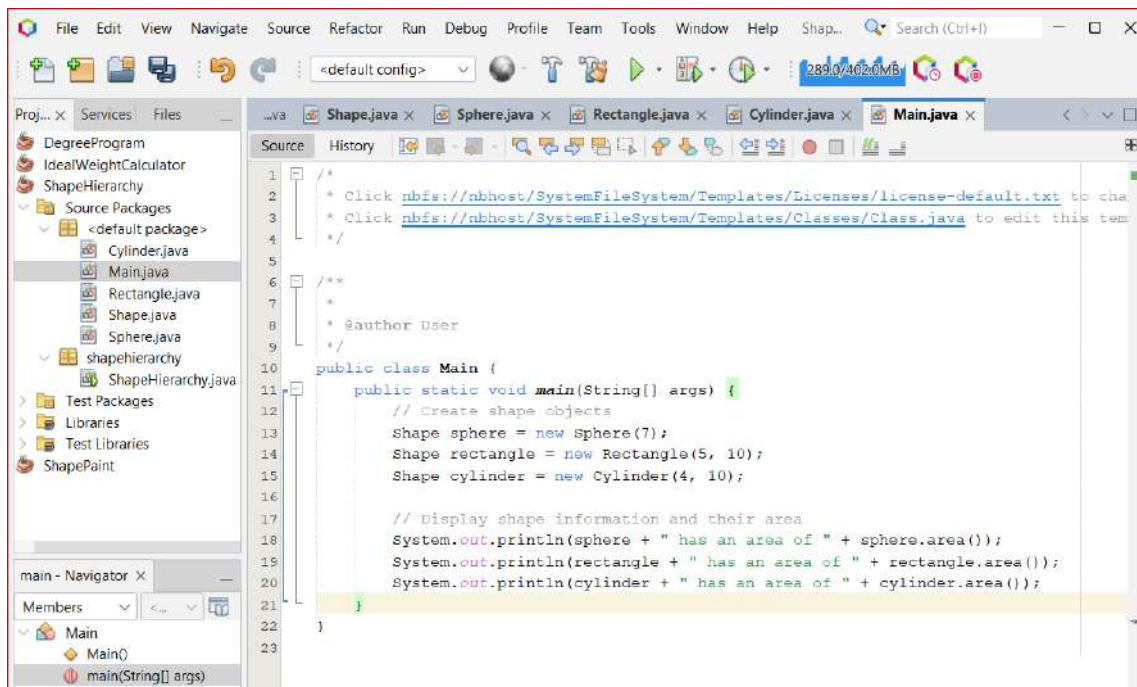


Figure 20 : Main Class

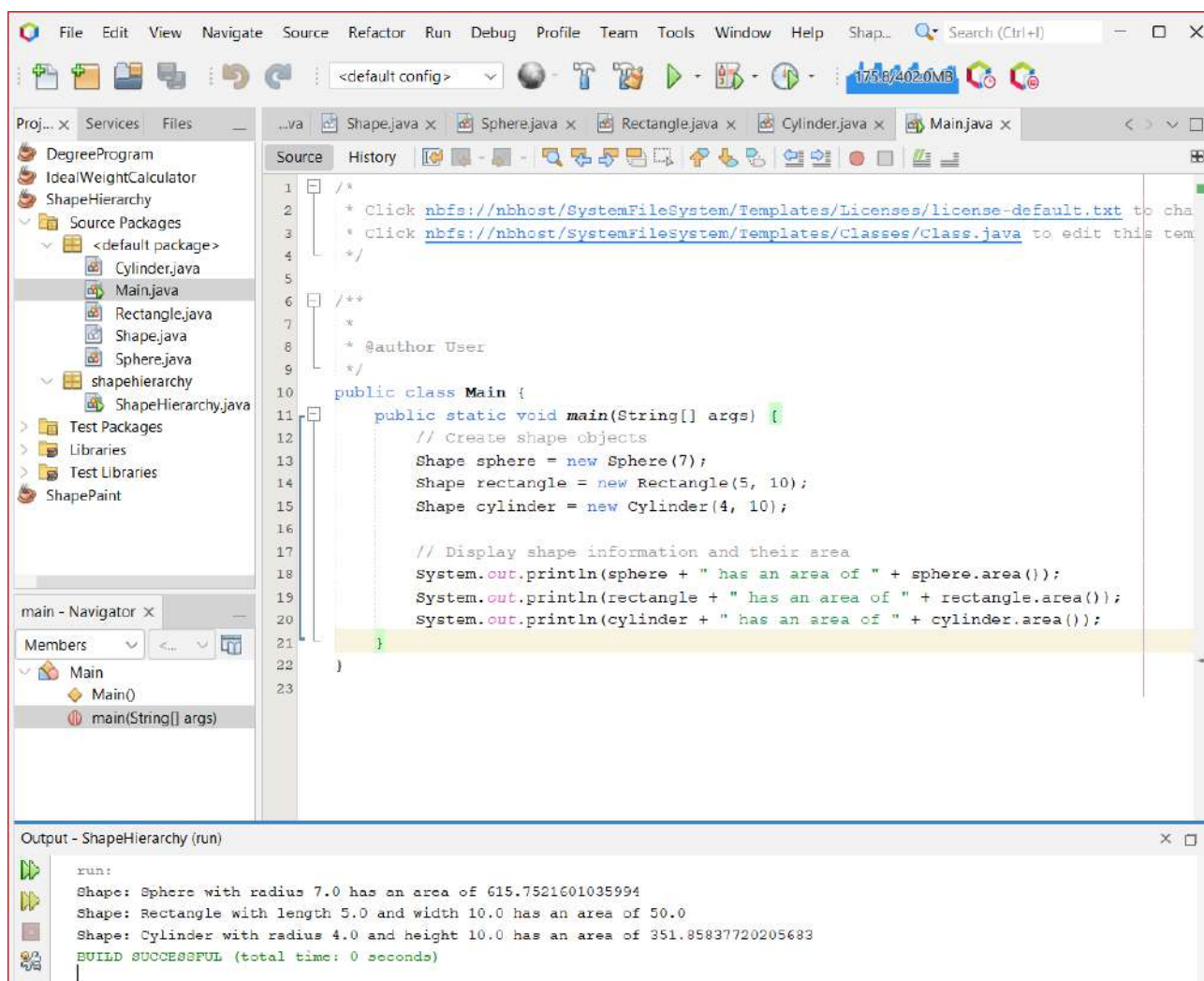


Figure 21: Output

- c) The file Paint.java contains a class for a type of paint (which has a "coverage" and a method to compute the amount of paint needed to paint a shape). Correct the return statement in the amount method so the correct amount will be returned. Use the fact that the amount of paint needed is the area of the shape divided by the coverage for the paint. (NOTE: Leave the print statement - it is there for illustration purposes, so you can see the method operating on different types of Shape objects.)

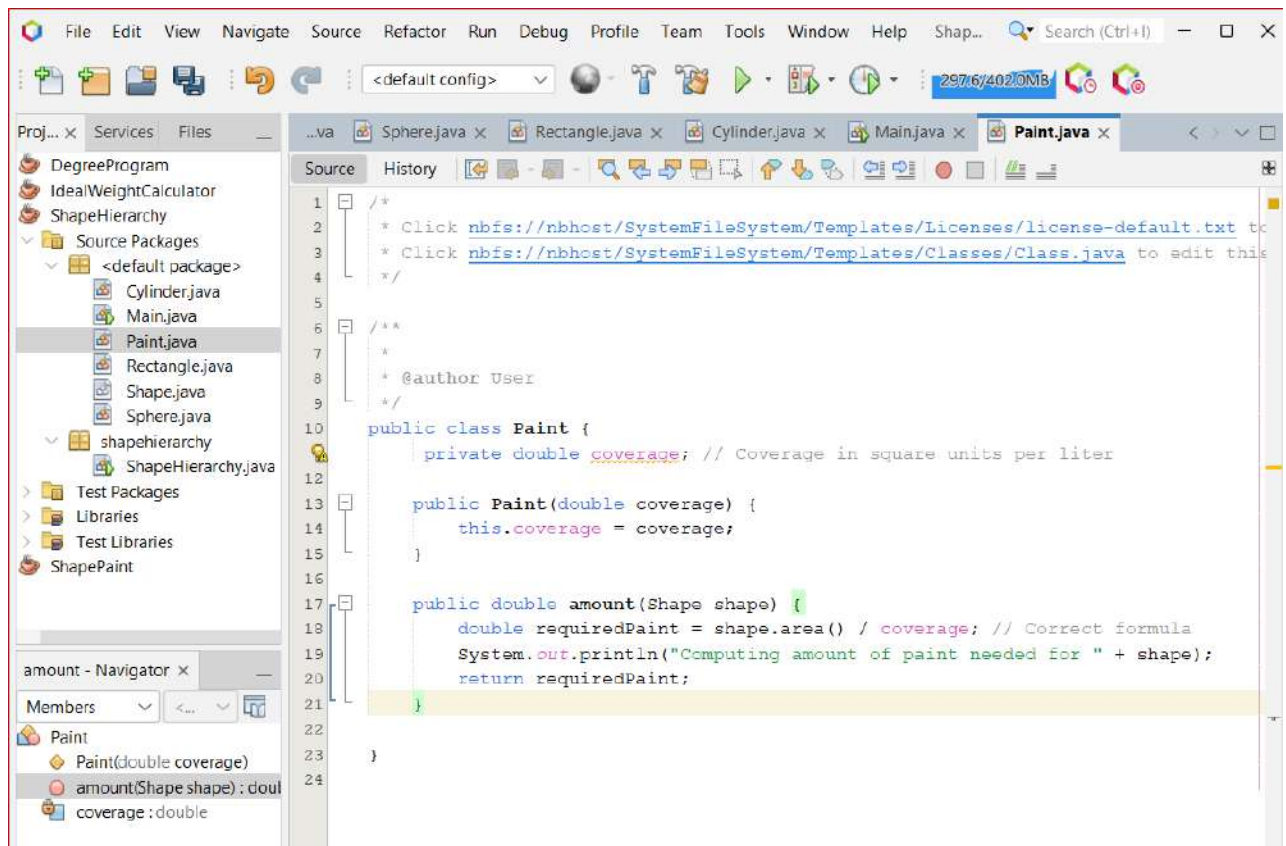


Figure 22 : Paint Class

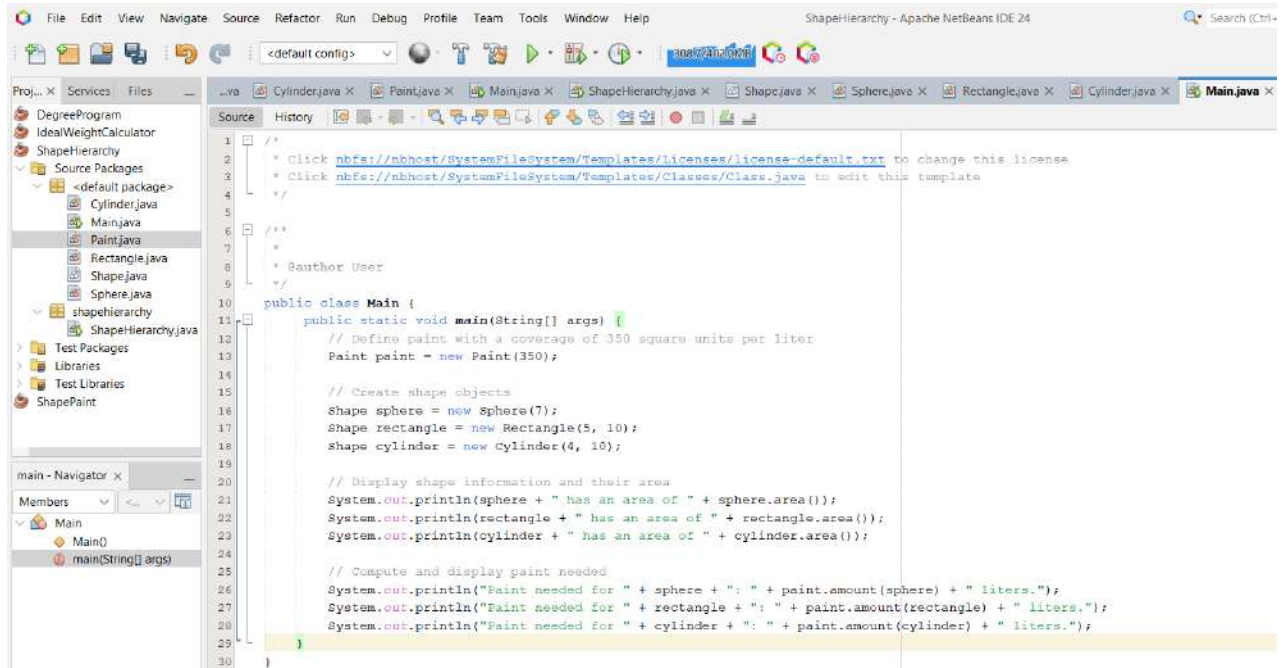


Figure 23 : Main Class

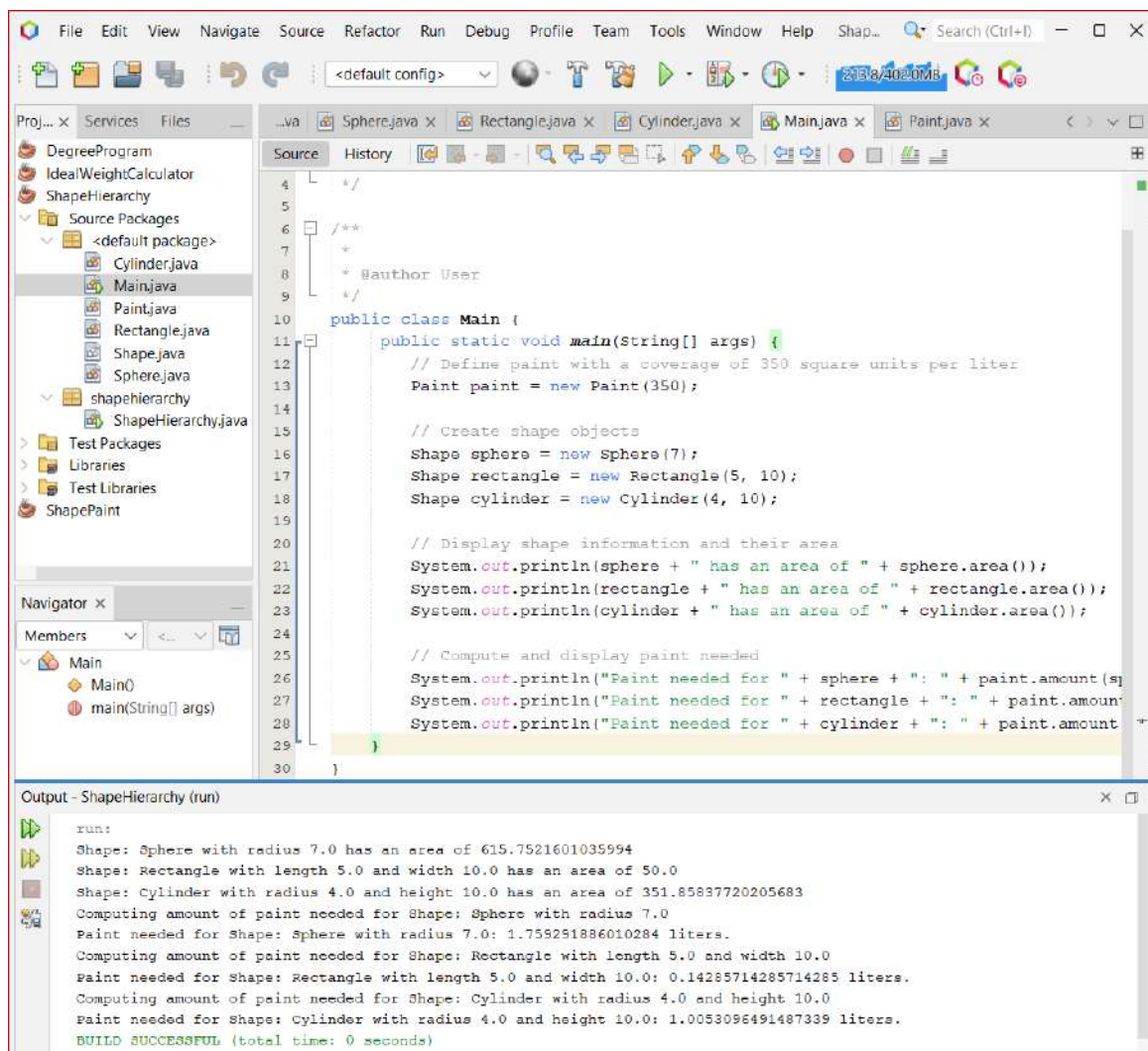
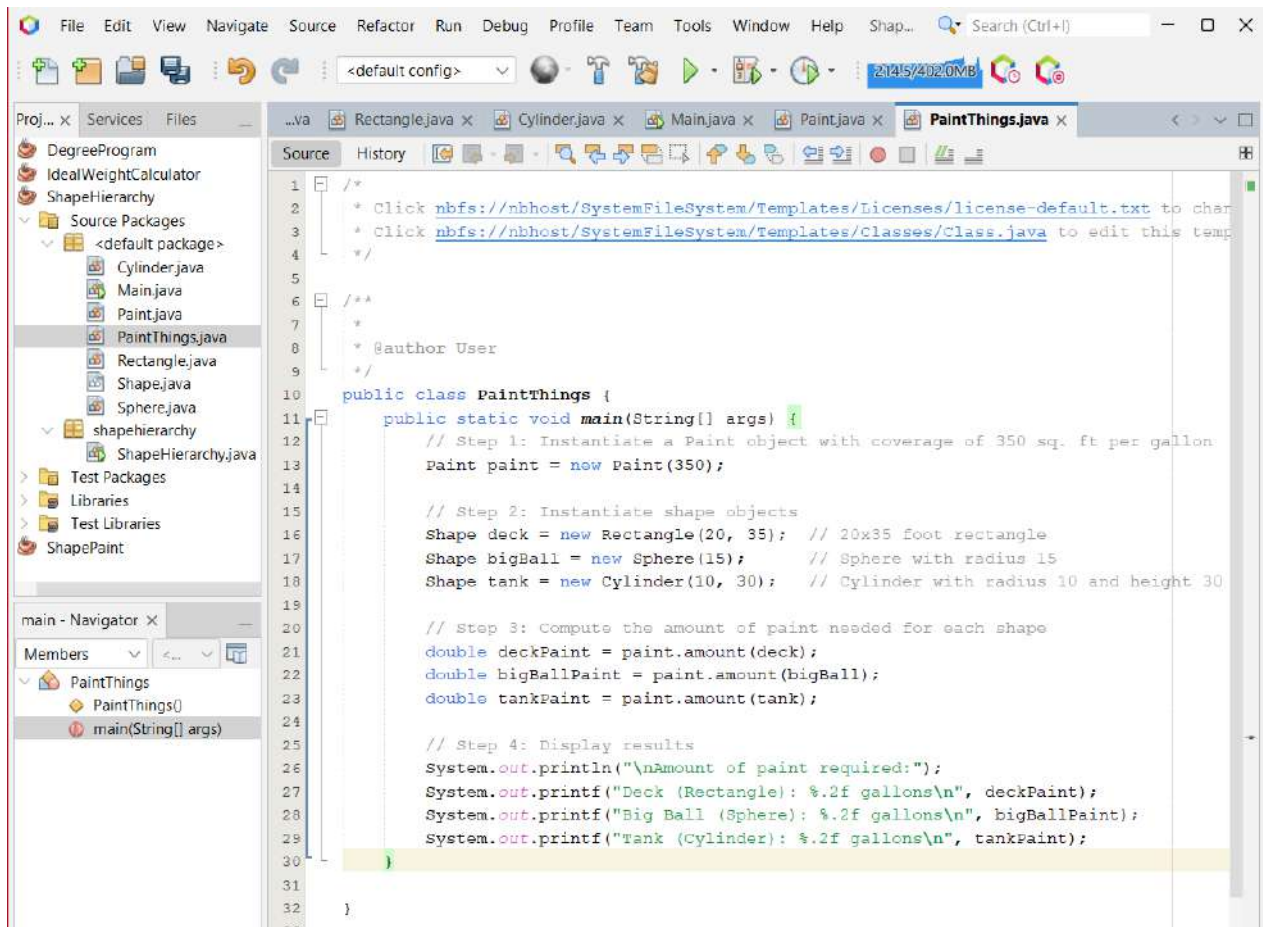


Figure 24 : Output

- d) The file `PaintThings.java` contains a program that computes the amount of paint needed to paint various shapes. A paint object has been instantiated. Add the following to complete the program: Instantiate the three shape objects: deck to be a 20 by 35 foot rectangle, bigBall to be a sphere of radius 15, and tank to be a cylinder of radius 10 and height 30. Make the appropriate method calls to assign the correct values to the three amount variables. Run the program and test it. You should see polymorphism in action as the amount method computes the amount of paint for various shapes.



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6  /**
7   *
8   * @author User
9   */
10 public class PaintThings {
11     public static void main(String[] args) {
12         // Step 1: Instantiate a Paint object with coverage of 350 sq. ft per gallon
13         Paint paint = new Paint(350);
14
15         // Step 2: Instantiate shape objects
16         Shape deck = new Rectangle(20, 35); // 20x35 foot rectangle
17         Shape bigBall = new Sphere(15); // Sphere with radius 15
18         Shape tank = new Cylinder(10, 30); // Cylinder with radius 10 and height 30
19
20         // Step 3: Compute the amount of paint needed for each shape
21         double deckPaint = paint.amount(deck);
22         double bigBallPaint = paint.amount(bigBall);
23         double tankPaint = paint.amount(tank);
24
25         // Step 4: Display results
26         System.out.println("\nAmount of paint required:");
27         System.out.printf("Deck (Rectangle): %.2f gallons\n", deckPaint);
28         System.out.printf("Big Ball (Sphere): %.2f gallons\n", bigBallPaint);
29         System.out.printf("Tank (Cylinder): %.2f gallons\n", tankPaint);
30     }
31 }
32
33 }
```

Figure 25 : *PaintThings Class*

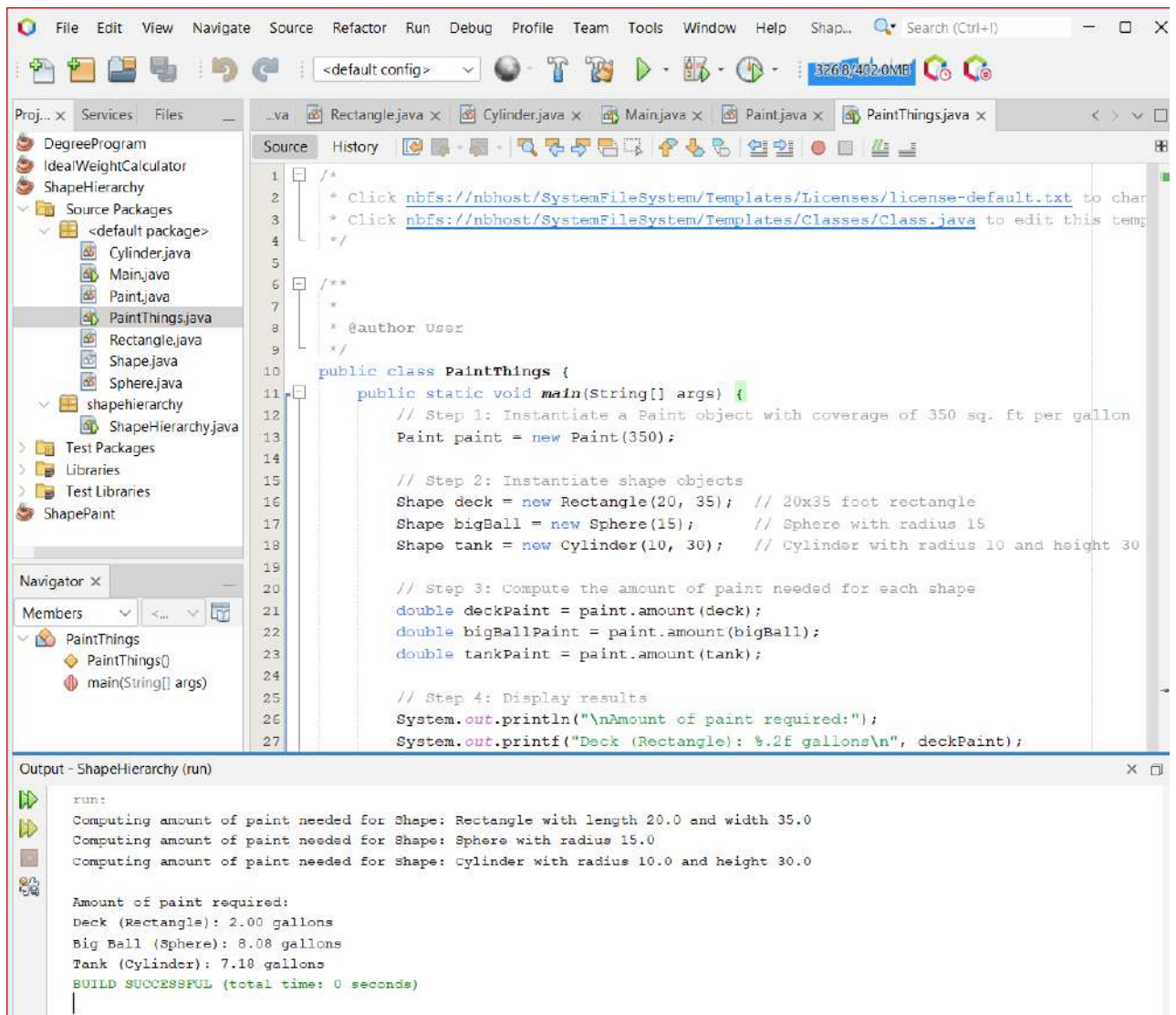


Figure 26 : Output