Object Oriented Programming

IY4101

Tutor: Ido

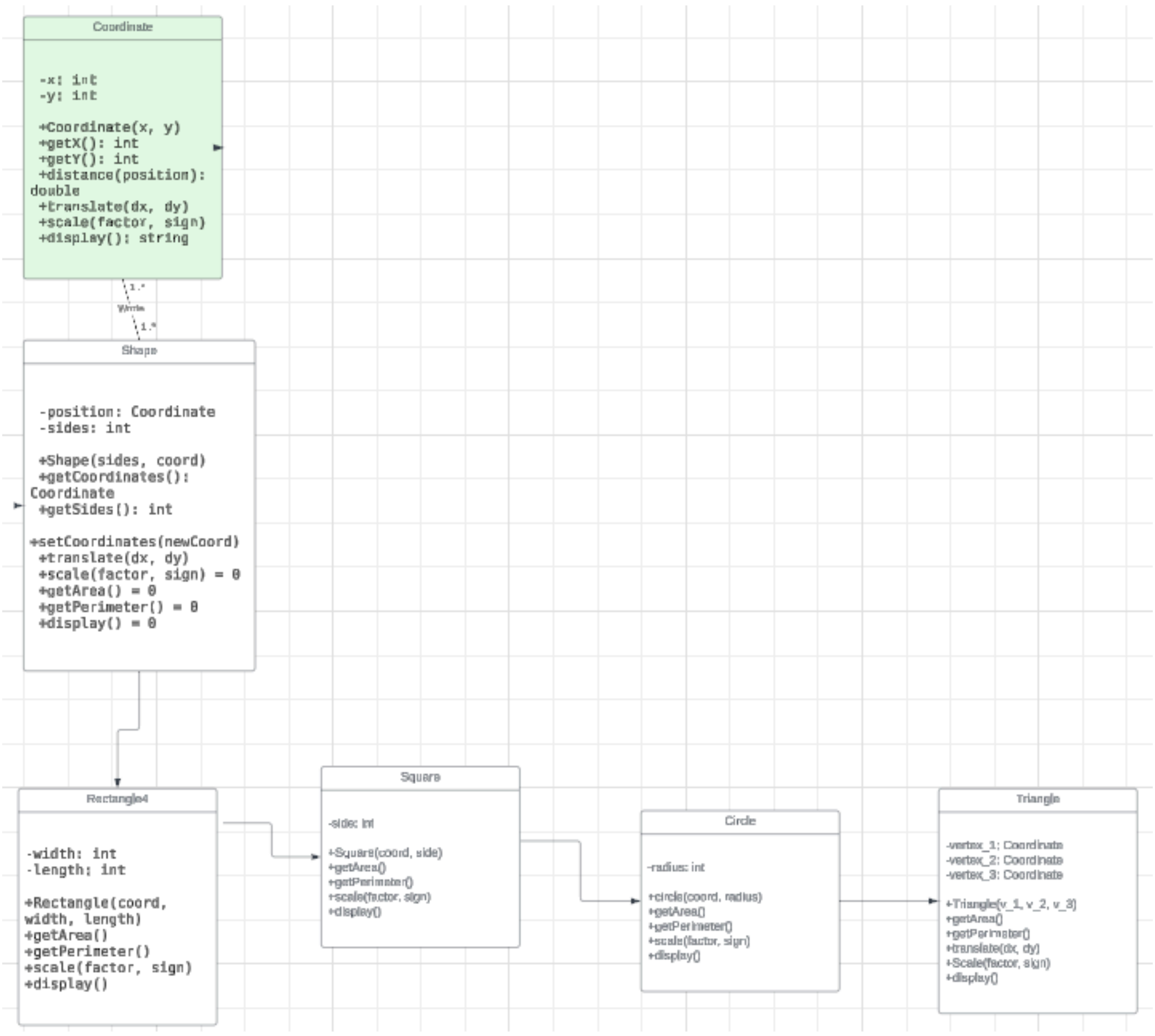
Student Name: Maco Wu

Student ID: P441802

I confirm that this assignment is my own work. Where I have referred to academic sources, I/we have provided in text citations and included the sources in the final reference list.

Table of Contents

UML diagram and their relationship



The concept of inheritance plays a significant role in object-oriented programming. Inheritance allows you to create a hierarchy of classes and derive the attributes and methods of the base class to the derived classes. In the context of shape classes, inheritance establishes a relationship between the base class "Shape" and its derived classes such as Rectangle, Square, Circle, and Triangle.

The Shape class, as the base class, defines the common attributes and methods that are shared by all the shapes. Some of the methods defined in the Shape class include getArea(), getPerimeter(), translate(), and scale(). These methods are then inherited by the derived classes, which can then implement their unique properties and behaviors.

For example, the Rectangle class inherits the attributes and methods of the Shape class and implements its specific properties such as length and width. Similarly, the Circle class inherits the attributes and methods of the Shape class and implements its specific properties such as radius.

In addition to the Shape class, the coordinate class is also an essential aspect of the shape management system. The coordinate class represents 2D on the canvas and provides a method for calculating the distances, translating, scaling, and displaying coordinates. The Shape class is used in conjunction with the coordinate class to store and adjust the position of the shape.

Furthermore, the ShapeList class is responsible for managing the list of shapes and provides methods to add, remove, translate, scale, and display the information. This class allows for easy manipulation and organization of the different shapes.

Finally, the ShapeManagement class serves as the entry point of the program and provides a user-friendly menu to interact with. This menu allows the user to create the shape, manipulate, and display the information for the different shapes through the menu options. With all these classes working together seamlessly, the shape management system provides a comprehensive solution for managing and manipulating different shapes.

Testing and Results

Action	Expected result	Actual result
Create a triangle whose vertices have the coordinates (50,50), (20,70), (70,70) and add it to the list of shapes	The triangle should be created with the vertices which have provide and added to the list.	As Expected
Create a Rectangle in position (100, 20) width 10, length 15 and add it to the list	The rectangle should be created with the width and length which is given and added to the list	As Expected
Create a circle in position (80, 100) and radius 25 and add it to the list of shapes	The circle should be created with the position and radius which is given and added to the list.	As Expected
Create a square in position (90, 40) witd side 20 and add it to the list of shapes	The square should be created withe the position with side which is given and added to the list	As Expected
Add three other shapes of your choice here	Additional shapes should be created and add to the list	As Expected
Show area and perimeter the second shape of the list	The area and perimeter should be displayed	As Expected
Display information of all the shapes	The info of all the shapes should displayed with no error	As Expected
Remove the third shape and check that no error occurs	The third shape should remove with no error	As Expected
Translate all the shapes by 10 for coordinate x and 15 for coordinate y	At the shapes should be translate by adding x = 10, and y = 15	As Expected
Display information of all the shapes	The information for all shapes should display with no error	As Expected
Increase all the shapes by a factor of 2	The shape should be increase by factor of 2 with no error	As Expected
Display all the shapes and see if they have changed in position and dimensions	It should show all the shape if they changed in position and dimension	As Expected
Add more Actions	Additional test can be added into the task	As Expected
Add actions for error handling such as: removing a shape in position 20, display the area of the	Error handling should be test without any error	As Expected