

HW3- Composite, Factories and Flyweight

Introduction

In this project, we are developing a reusable shape library which is open to extension and closed to modification. Our focus is to keep the library simple, modular and reusable.

Aim

Our aim to is to become familiar with Composite pattern, Factory Method pattern and Flyweight pattern.

UML Diagrams

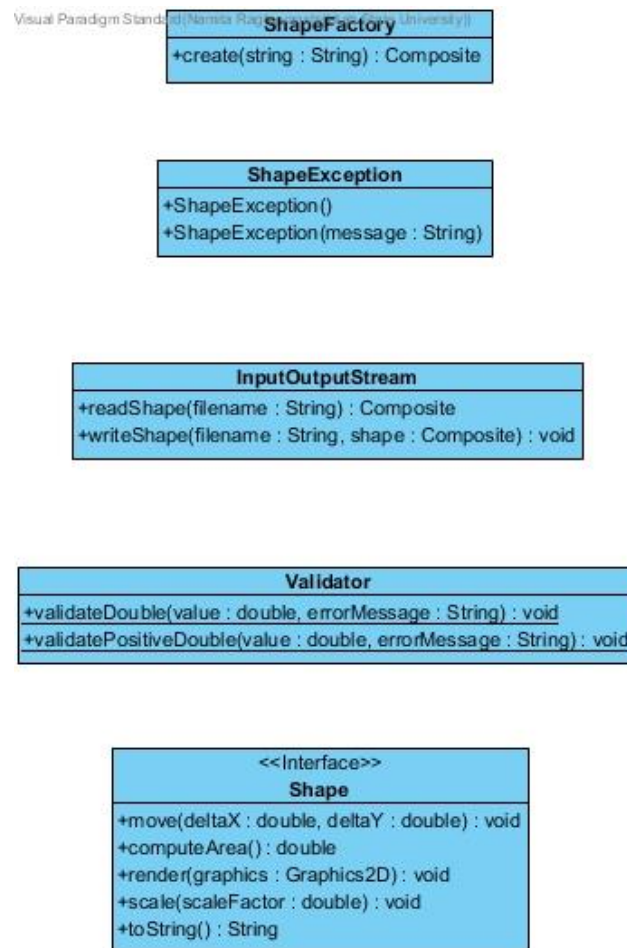


Fig. 1.1 Class Diagram of Shape Library

Visual Paradigm Standard (Namita Raghuvanshi/Jawahar University)

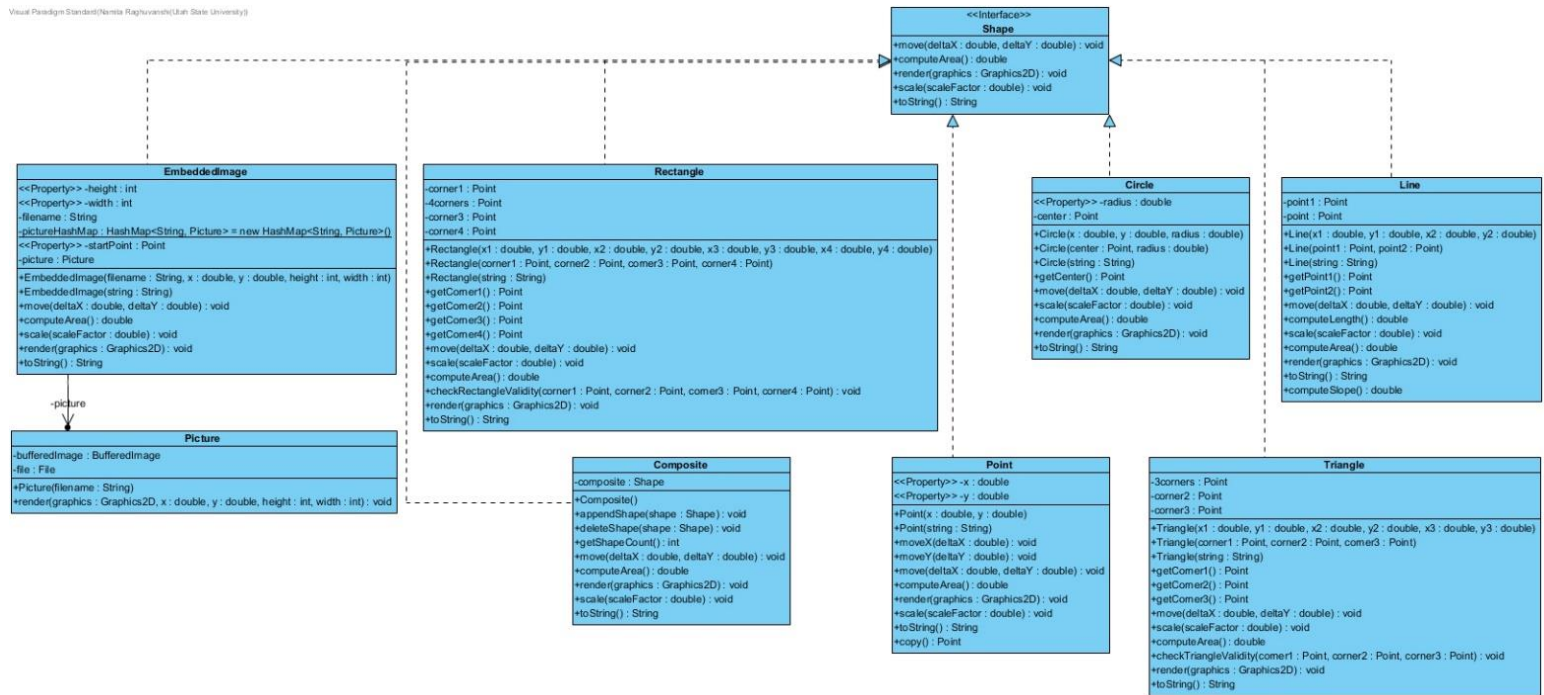


Fig. 1.2 Class Diagram of Shape Interface

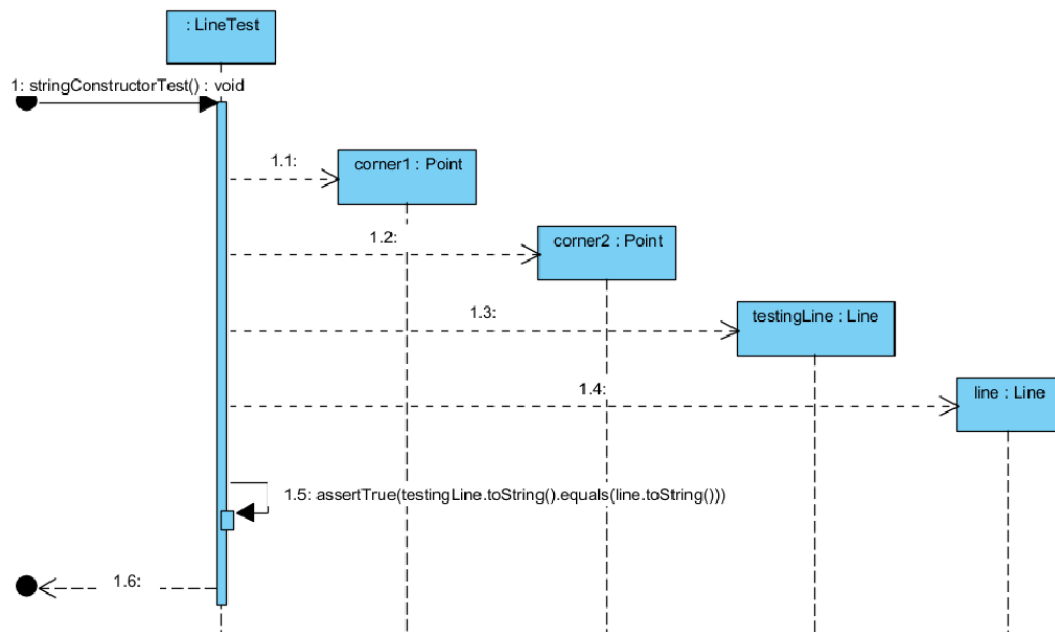


Fig 2.1 Interaction Diagram of Line Constructor.

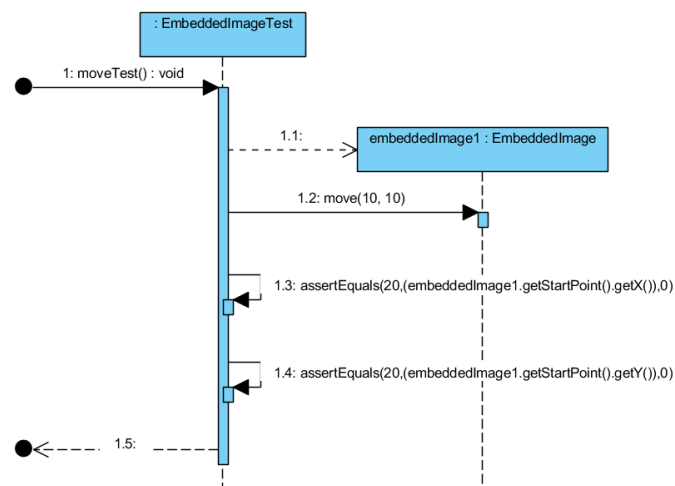


Fig 2.2 Interaction Diagram of Embedded Image move Testing.

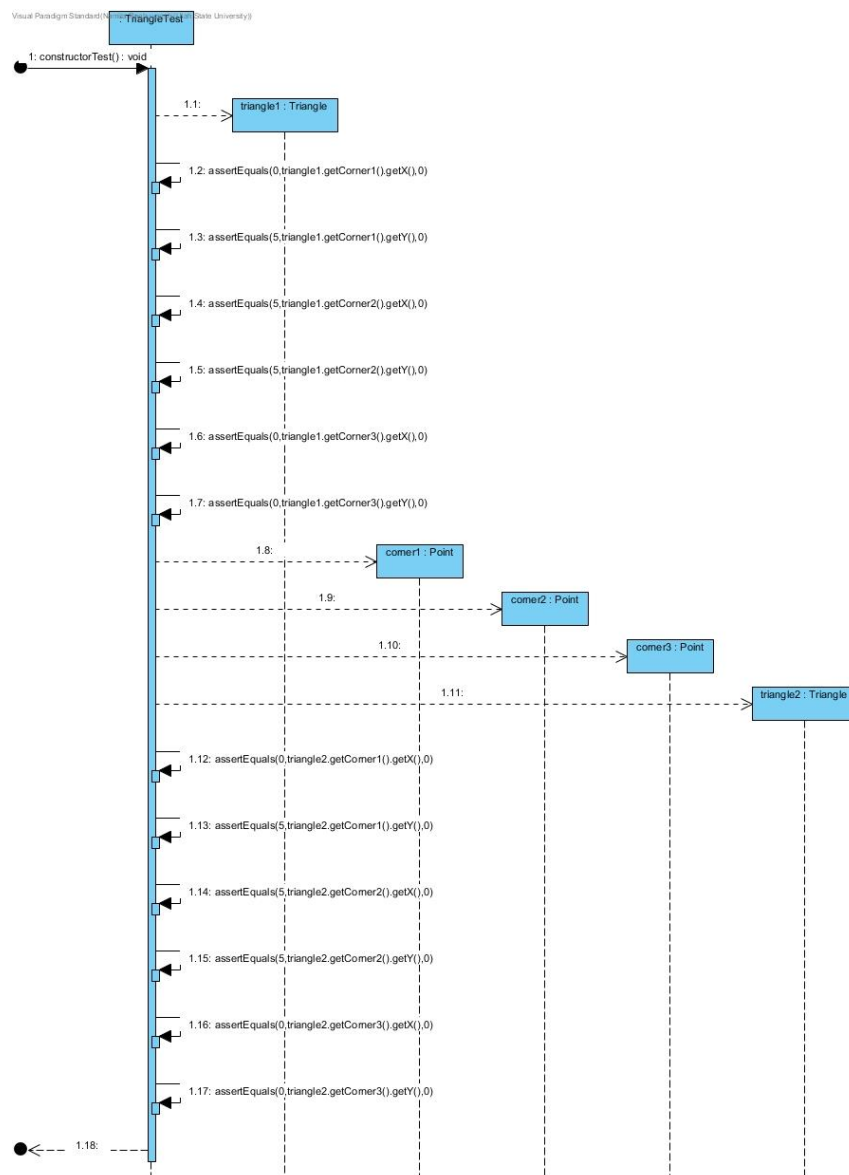


Fig 2.3 Interaction Diagram of Triangle constructor test.

Insights Uncovered

-While doing the project I have learned a lot about testing and I also know it now that creating a code with better modularity, encapsulation and abstraction leads to easily testable cases.

-I got introduced to Flyweight pattern and Composite pattern which are both quite unique and different than the previous patterns that we learned in class and it did clear a lot of concepts and it is also a very efficient way to create such ShapeLibrary.

-While trying to figure out the Flyweight and Composite patterns that the project demands, I have appended a lot into my knowledge domain of java. Also I have faced the fact that every time we should try to figure out new methods and ways rather than coding just with the pre-existing knowledge.

-In this assignment too we used Factory Pattern. Using this pattern again, cleared the concept even more and now this is a concept I will know exactly when to use.

-one of my bad practice that I tackled while doing this project is that we should always be open to new ideas and strategy rather than thinking in one direction. Because sometimes the only solution is to change the whole plan and get started again from a new point. I have wasted a lot of time trying to figure out things where all I needed was to think differently about a new solution.

-I have also figured out that the naming conventions are not to be taken lightly as it could make the code look messy and also it won't be readable for someone else. Also it is time wasting to debug a code which does not have proper names.

-Last point I learned is that we should always code in such a way that further add-ons and changes should not be a cumbersome task.

-In this assignment I have applied a flyweight so that it doesn't save the same image again and again and save memory and processing time.

- I have also applied Composite pattern with the help of which I can append shapes and make it count as one diagram.