

Java Parser for UML Class diagram

Name : Harsha Muktamath

SJSU ID : 011825138

YouTube Link – <https://youtu.be/nosPm36FwT8>

Github Link - <https://github.com/muktamath81/UML-Parser>

Kanban Link - <https://waffle.io/muktamath81/UML-Parser>

Initial setup -

- Install Java and use any available IDE for development. In my project, I have used Eclipse and to manage dependencies Maven.
- GraphViz must be installed on the system. It does not have plugins, hence it has to be installed (Link- <http://www.graphviz.org/Download..php>)

Instructions to execute the program

- 1) Download the package
- 2) Download the test files (test*.zip) and unzip it to a location (here – sourcefilepath)
- 3) Execute the code as below

• `java -jar <jar file/Renamed file> <sourcefilepath> <outputfilenameandpath>`

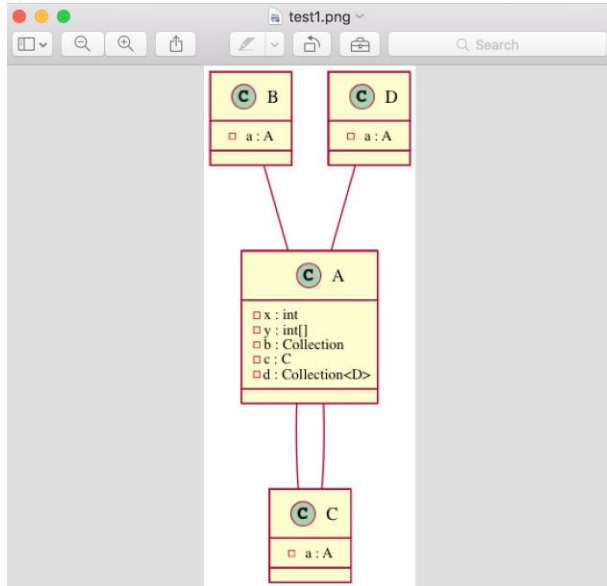
- sourcefilepath - Path to the default package folder.
- outputfilenameandpath – filename and path of the output image

In my example, I have executed as below (Here I have renamed *.jar file to UMLParser and executed)

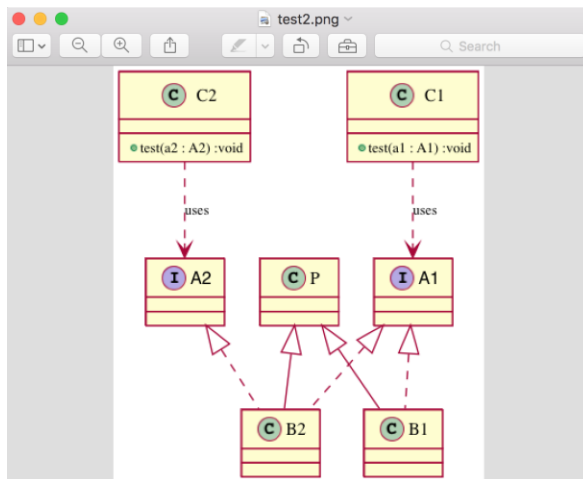
```
classDiagram
    class Component {
        +operation() String
    }
    class ConcreteDecoratorA {
        +addedState : String
        +operation() String
        +ConcreteDecoratorA(c : Component)
    }
    class ConcreteDecoratorB {
        +addedState : String
        +operation() String
        +ConcreteDecoratorB(hc : Component)
    }
    class Decorator {
        +component : Component
        +operation() String
        +Decorator(c : Component)
    }
    class ConcreteComponent {
        +operation() String
    }
    class Tester {
        +mainArgs : String[]
        +void
    }
    Component <|-- ConcreteDecoratorA
    Component <|-- ConcreteDecoratorB
    Component <|-- Decorator
    Component <|-- ConcreteComponent
    ConcreteDecoratorA --> Component : uses
    ConcreteDecoratorA --> Decorator : uses
    Decorator --> Component : uses
    ConcreteDecoratorB --> Decorator : uses
    ConcreteComponent ..> Component : uses
    Tester ..> ConcreteDecoratorA : uses
    Tester ..> Decorator : uses
```

Sample Output from my UML Parser code –

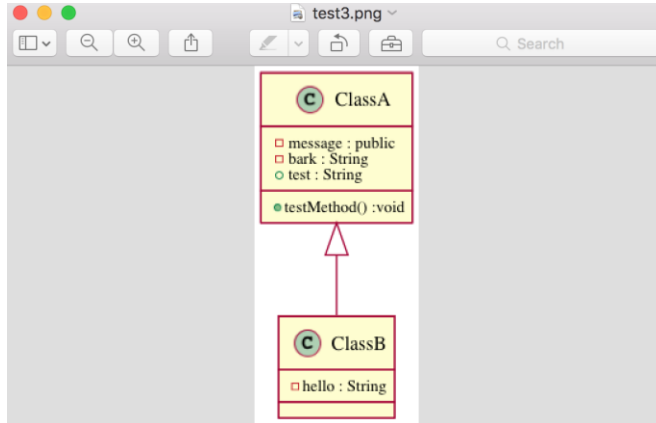
Test Case 1:



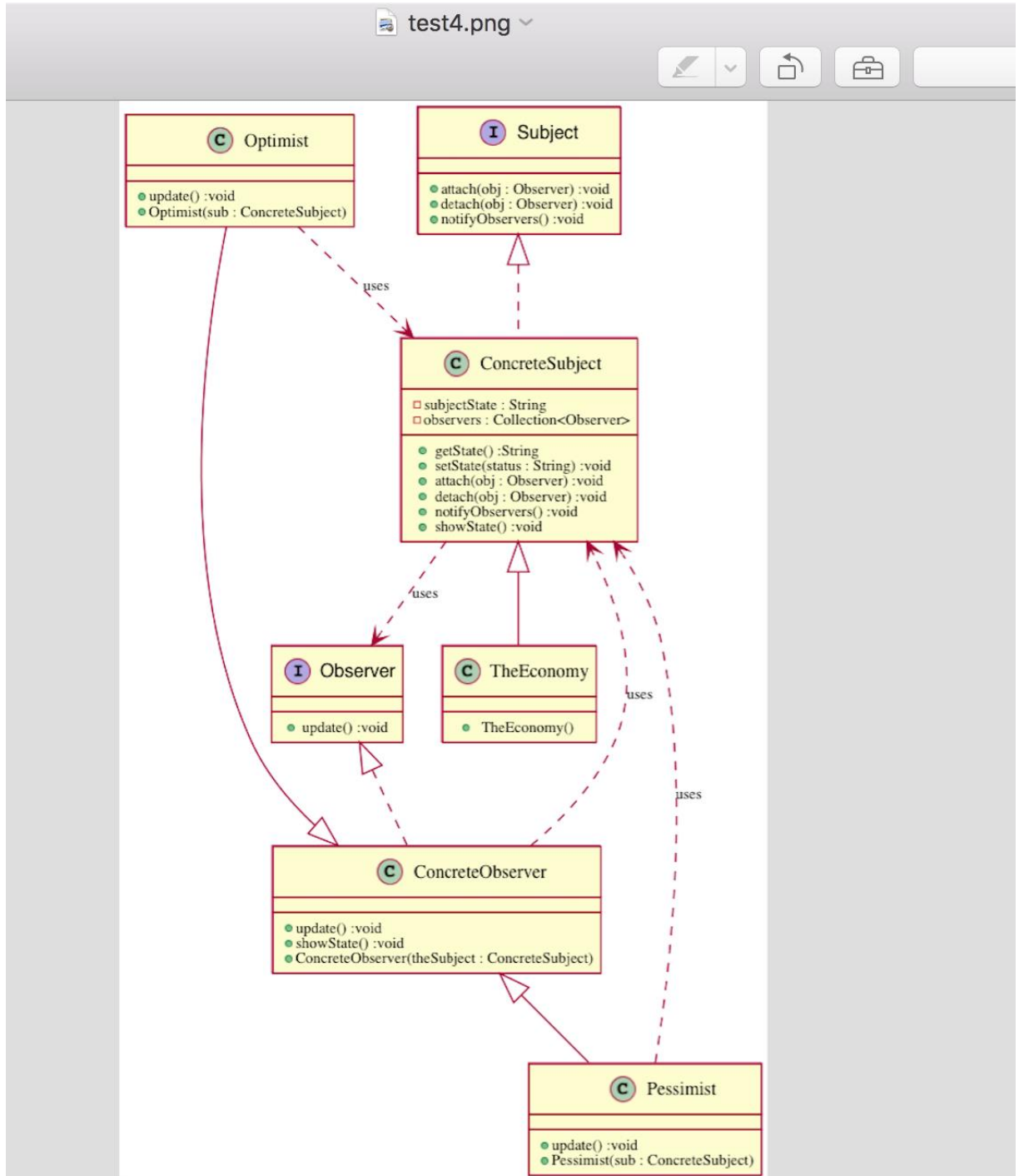
Test Case 2:



Test Case 3:



Test Case 4:



Test Case 5:

