CSE – 443 OBJECT ORIENTED ANALYSIS AND DESIGN

HOMEWORK 2 REPORT

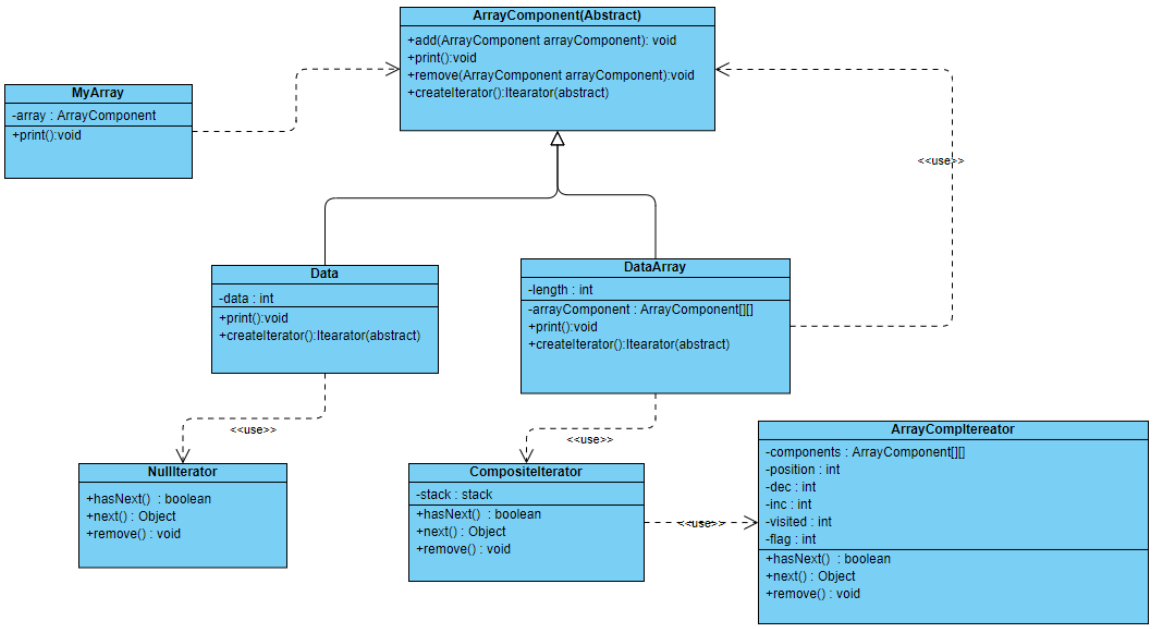
OĞUZHAN SEZGİN

1801042005

QUESTION 2

In this part of homework, I used **iterator design pattern**. Because client wants to print 2D array in a certain direction. This means that I have iterate array in a certain direction. I created **ArrayComponent** abstract class and derived Data class and **DataArray** class from this **ArrayComponent** class. array can have a data or also a data array. Then I created **ArrayCompIterator**. This class implements **java.util.Iterator** interface. This iterator spirally iterates simple 2D array clockwise direction. I created **NullIterator** class. This class for simple data. It iterates nothing. I also created **CompositeIterator** class. This class also implements java.util.Iterator interface. This iterator iterates main array. This means that this iterator iterates simple data or data arrays together. This iterator uses **ArrayCompIterator** and **NullIterator** while iterating. Finally, I created **MyArray** class. This is main array class it has array components and iterates this component with **CompositeIterator** class.

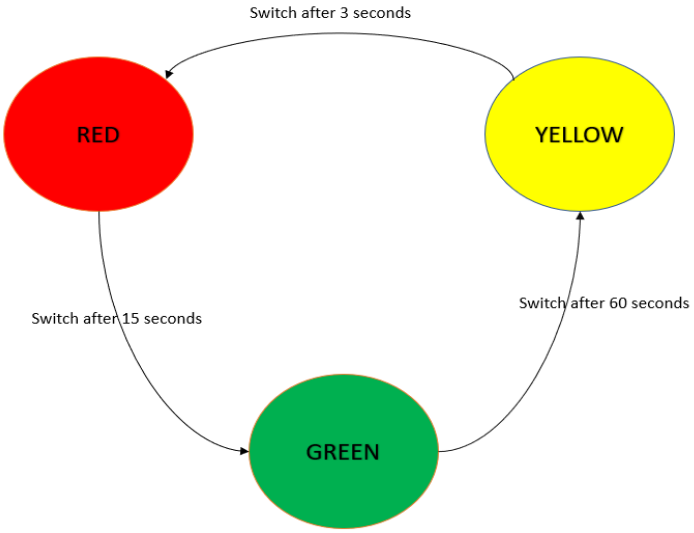
The class diagram showing below:



QUESTION 3

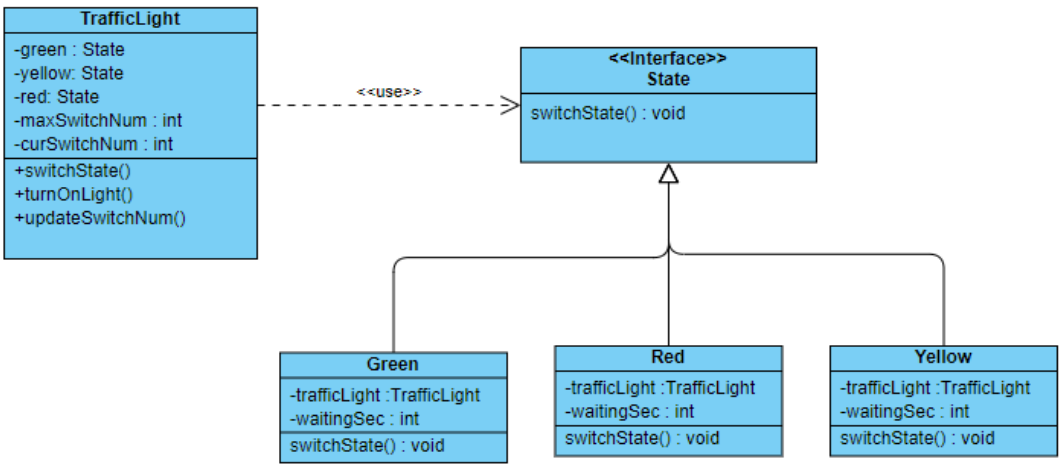
In this part of homework, I used state design pattern as stated in the question. Firstly, I created State interface. This interface has just **switchState** method. Then I created Green, Red and Yellow classes. This classes are state classes, so they implement State interface. This classes first execute their task (e.g. green light turns on) then switch state with using **switchState** method. This classes also has **waitingSec** variable this variable exists for waiting second in the states.

The state diagram showing below:



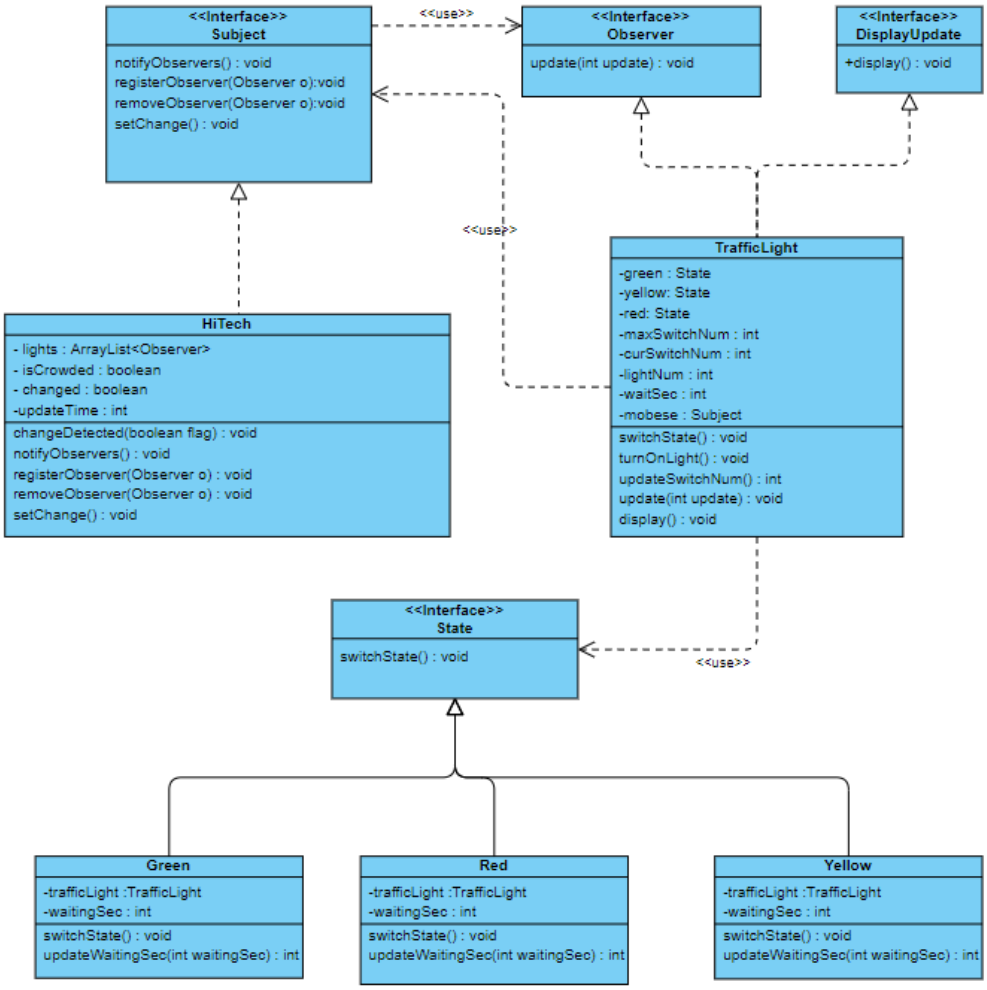
I created traffic light class to execute this states. This class has states which showing above. When the user call **turnOnLight** function states execute sequentially. The parameter given when creating to this class for prevent infinite loop. Traffic lights state execute as much as that number.

The class diagram showing below:



After these implementations I added new classes. This classes update green light stay on seconds if the traffic gets crowded. I used **Observer design pattern** for this part. I created a **HiTech** class. This class is a subject class and traffic lights subscribe this camera class. When camera detects a lot of traffic it calls **changeDetected** method with true parameter. Then the green light stay on seconds updates. If traffic returns to normal, then camera calls **changeDetected** method with true parameter again and execute second updates again. In every update traffic lights prints their new seconds.

The class diagram showing below:



All of these packages have their own Tester class.

Tester classes tests all methods of their package class.

Javadoc document is in doc file(index.html).