CSE443 Homework #2 Report

1.1-) What happens if someone tries to clone a Singleton object using the clone() method inherited from Object? Does it lead to the creation of a second distinct Singleton object?

- Firstly, clone method is defined as protected in Object, so we have to override clone method in Singleton class to use it, otherwise we can’t use this method. Also for using clone method out Singleton class must implement Clonable interface otherwise whenever we use clone method it will throw CloneNotSupportedException. So if Singleton class implement Clonable and override clone method it can use clone method and it will lead to the creation of a second distinct Singleton object.

1.2-) Cloning Singletons should not be allowed. How can you prevent the clonning of a Singleton object?

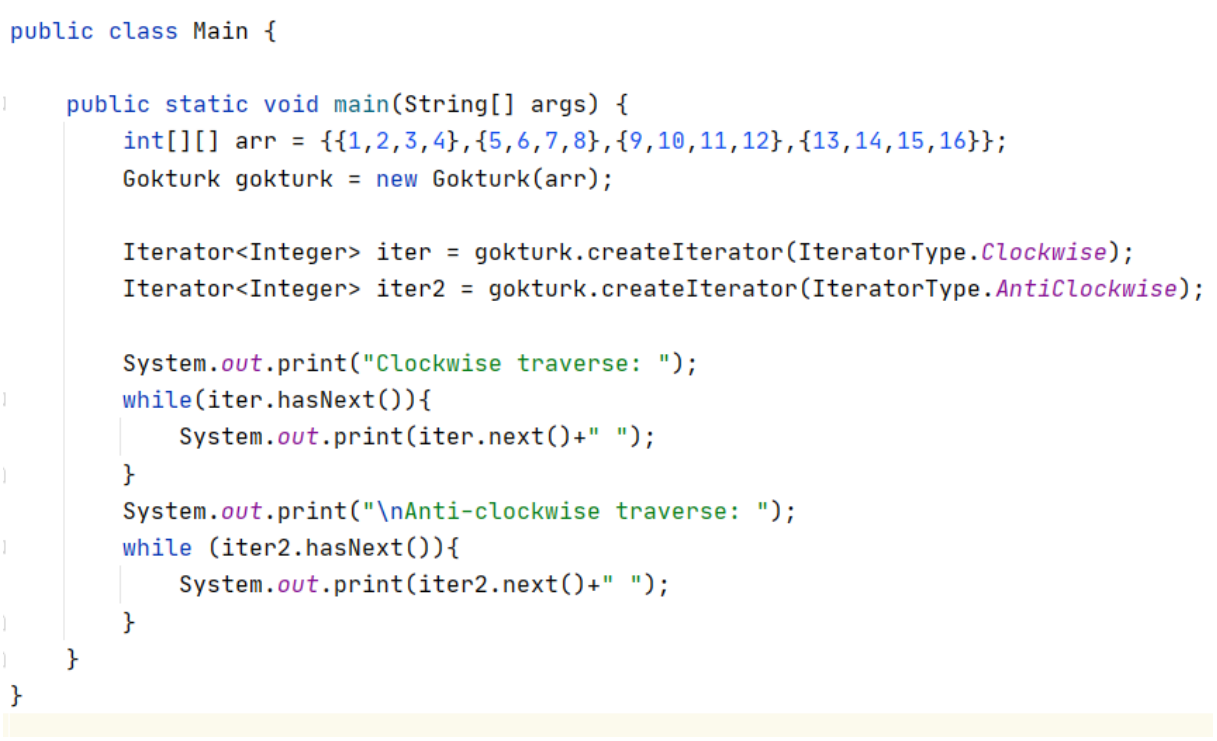
- If Singleton class not implement Clonable interface whenever we call clone it will throw CloneNotSupportedException so by this way we can prevent the clonning of a Singleton object. Second way we can override clone method as it will throw an exception whenever called or return same Singleton object instance.

1.3-) Let’s assume the class Singleton is a subclass of Parent, that fully implements the Clonable interface. How would you answer question 1 and 2 in this case?

- If parent class implement clonable then our Singleton class also implement this so it can use clone method as well and it lead to the creation of a second distinct Singleton object.
- For prevent this, in Singleton overrided clone method we can just throw an exception or just return same instance of Singleton so whenever we call clone it can throw a exception or return same Singleton object. With this way we can prevent the clonning of a Singleton object

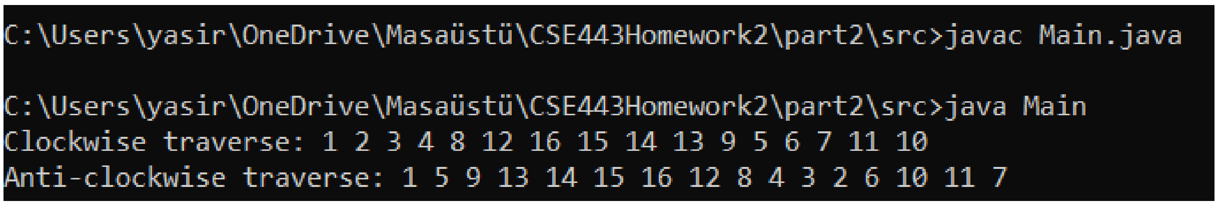
2-) I used iterator pattern fort his question. There is 2 iterator class for clockwise and anti-clockwise and there is a create iterator method in Gokturk class. So with this method you can create both of the iterators and by using them you can iterate array at clockwise or anti-clockwise.

In driver program I created same array in homework then traverse it both iterator with its hasNext and Next methods and print traverse results. If you want you can change array in main to another 2d array it will still work.

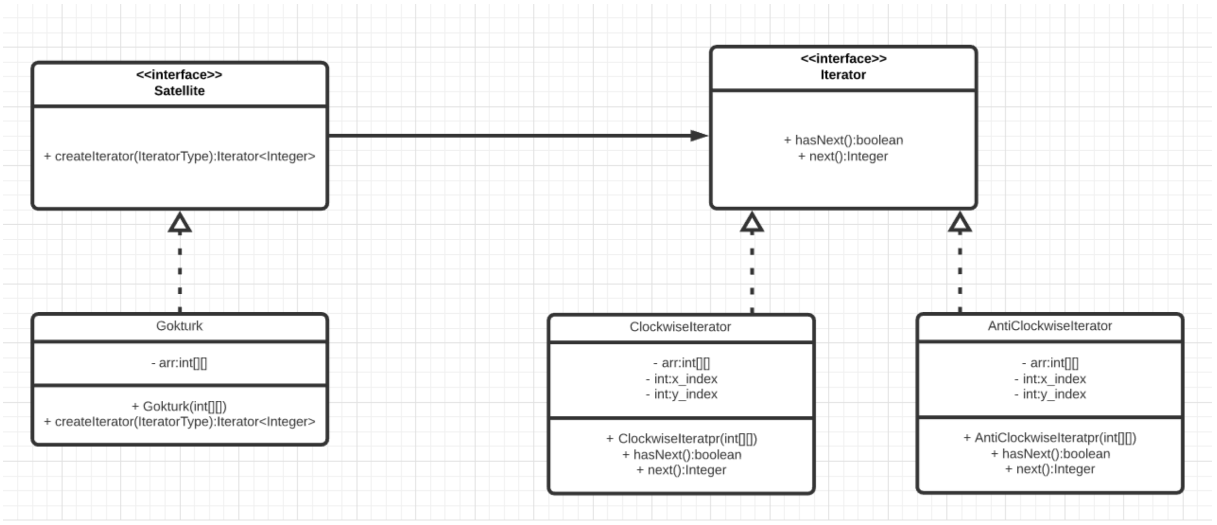


```java
public class Main {

    public static void main(String[] args) {
        int[][] arr = {{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
        Gokturk gokturk = new Gokturk(arr);

        Iterator<Integer> iter = gokturk.createIterator(IteratorType.Clockwise);
        Iterator<Integer> iter2 = gokturk.createIterator(IteratorType.AntiClockwise);

        System.out.print("Clockwise traverse: ");
        while(iter.hasNext()){
            System.out.print(iter.next()+" ");
        }
        System.out.print("\nAnti-clockwise traverse: ");
        while (iter2.hasNext()){
            System.out.print(iter2.next()+" ");
        }
    }
}
```

İter is a clockwise iterator so first it traverse array by clockwise and iter2 is anti-clockwise iterator it will iterate array as anti-clockwise.

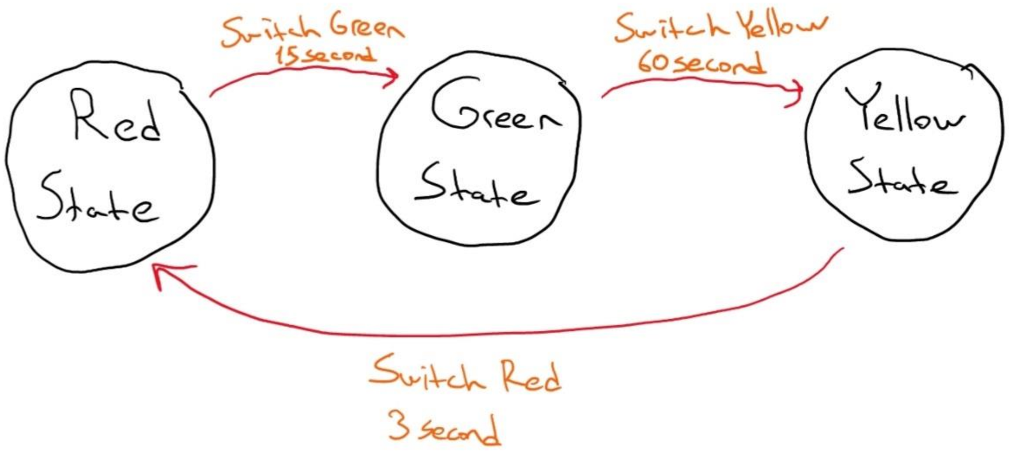


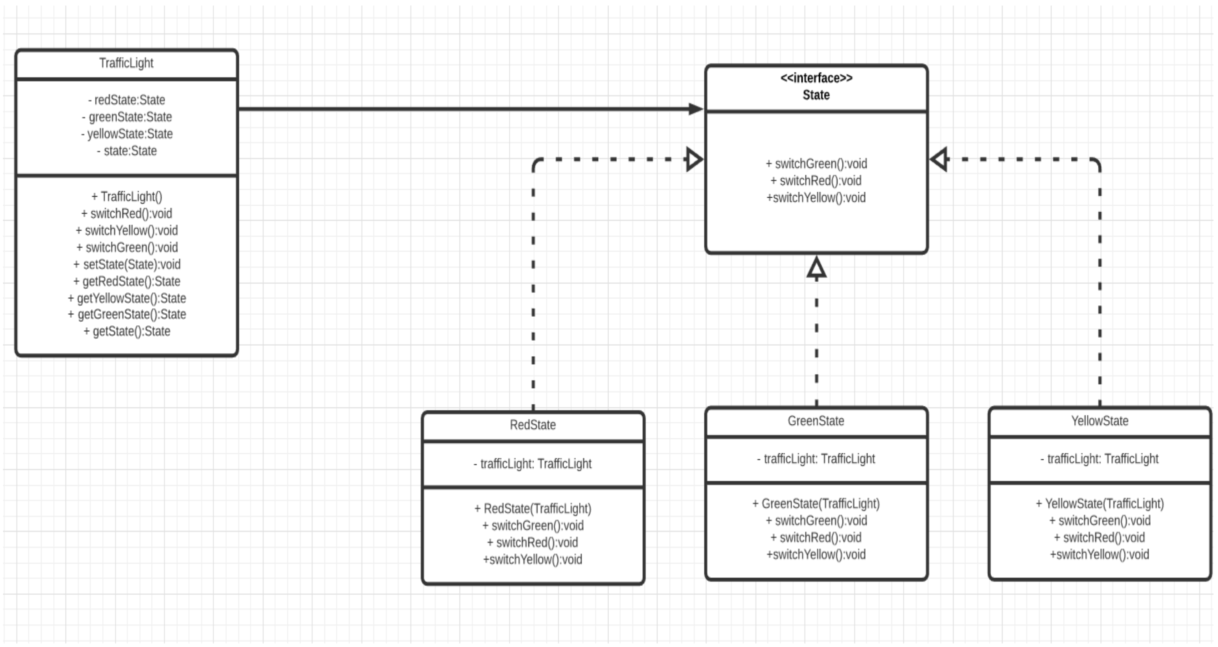
```
C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part2\src>javac Main.java

C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part2\src>java Main
Clockwise traverse: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
Anti-clockwise traverse: 1 5 9 13 14 15 16 12 8 4 3 2 6 10 11 7
```



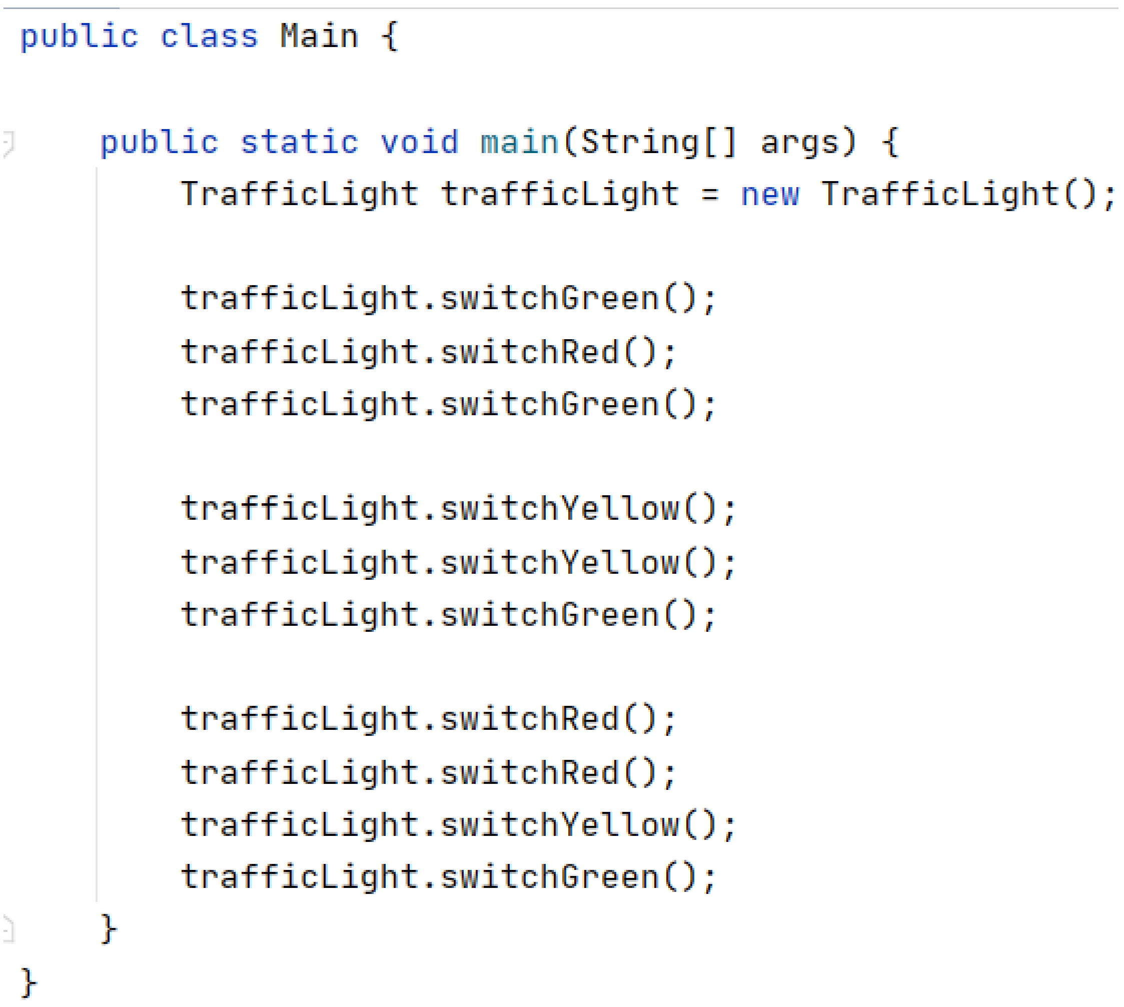
Here Iterator interface come from java default Iterator interface and I override its next and hasNext methods in ClockwiseIterator and AntiClockwiseIterator.

3-) Fort his question I used State pattern. Here I create 3 class for each state(Red,Green and Yellow) under a State Interface and there is a TrafficLight class. I used my states in this TrafficLight class. Also for first part of this question this is my state diagram and class diagram:

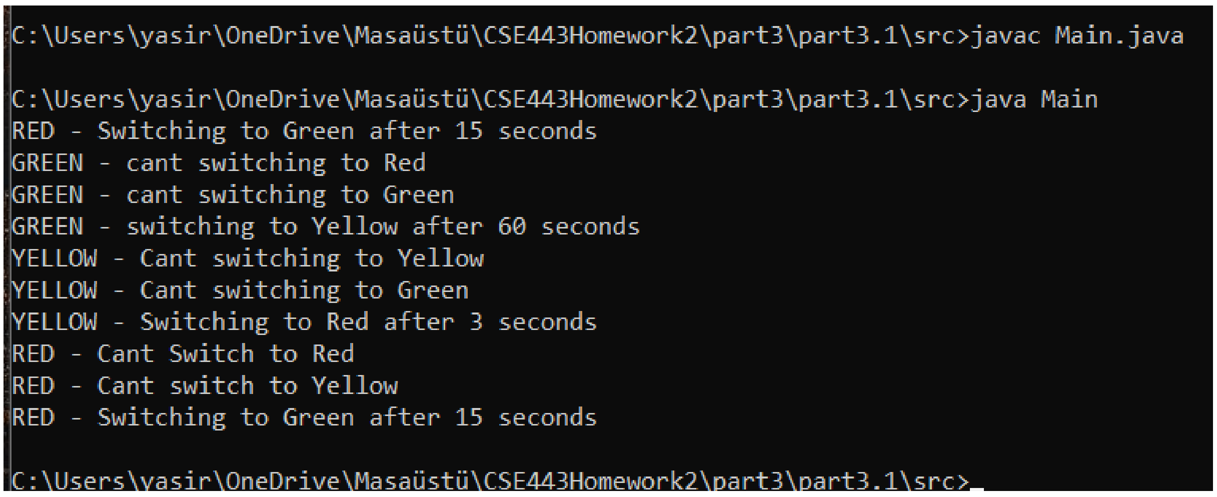




In driver, I create TrafficLight object and illustrate every state and every transition with text output on the terminal as mentioned in homework pdf.



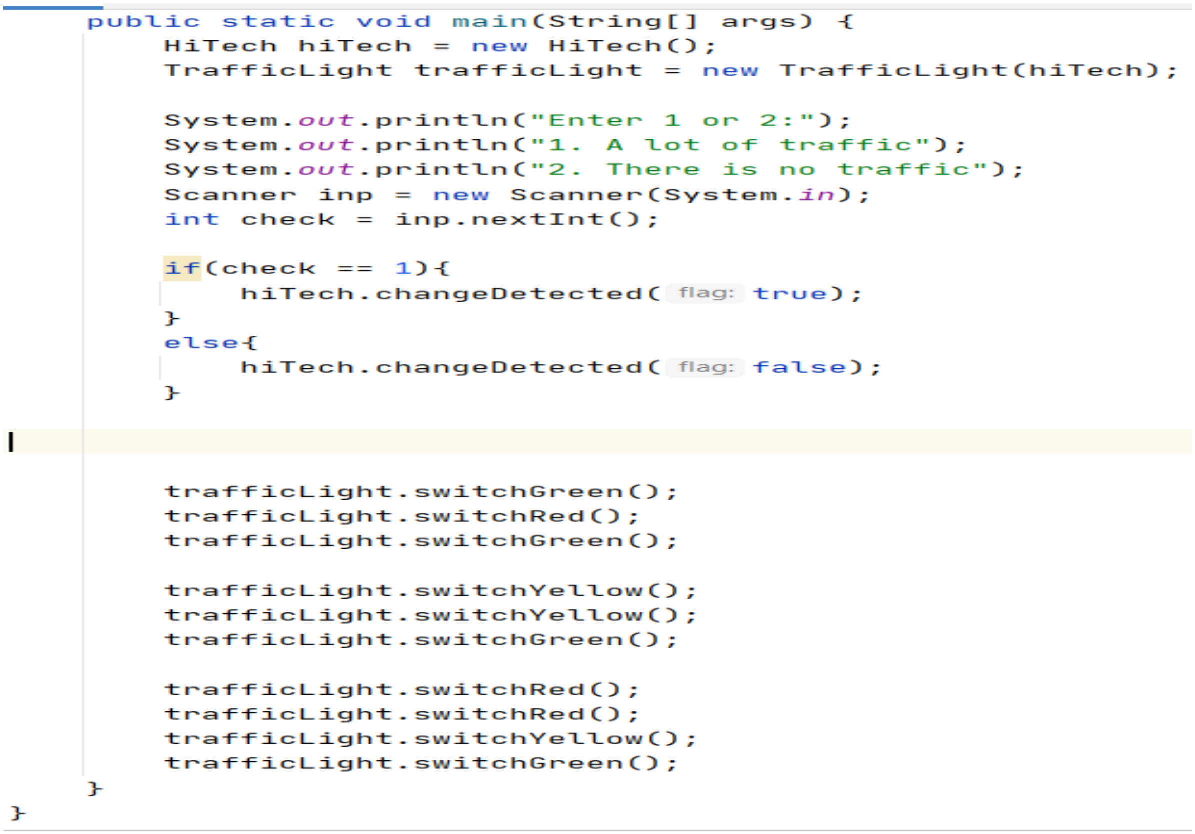
```java
public class Main {

    public static void main(String[] args) {
        TrafficLight trafficLight = new TrafficLight();

        trafficLight.switchGreen();
        trafficLight.switchRed();
        trafficLight.switchGreen();

        trafficLight.switchYellow();
        trafficLight.switchYellow();
        trafficLight.switchGreen();

        trafficLight.switchRed();
        trafficLight.switchRed();
        trafficLight.switchYellow();
        trafficLight.switchGreen();
    }
}
```



```
C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.1\src>javac Main.java

C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.1\src>java Main
RED - Switching to Green after 15 seconds
GREEN - cant switching to Red
GREEN - cant switching to Green
GREEN - switching to Yellow after 60 seconds
YELLOW - Cant switching to Yellow
YELLOW - Cant switching to Green
YELLOW - Switching to Red after 3 seconds
RED - Cant Switch to Red
RED - Cant switch to Yellow
RED - Switching to Green after 15 seconds

C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.1\src>_
```

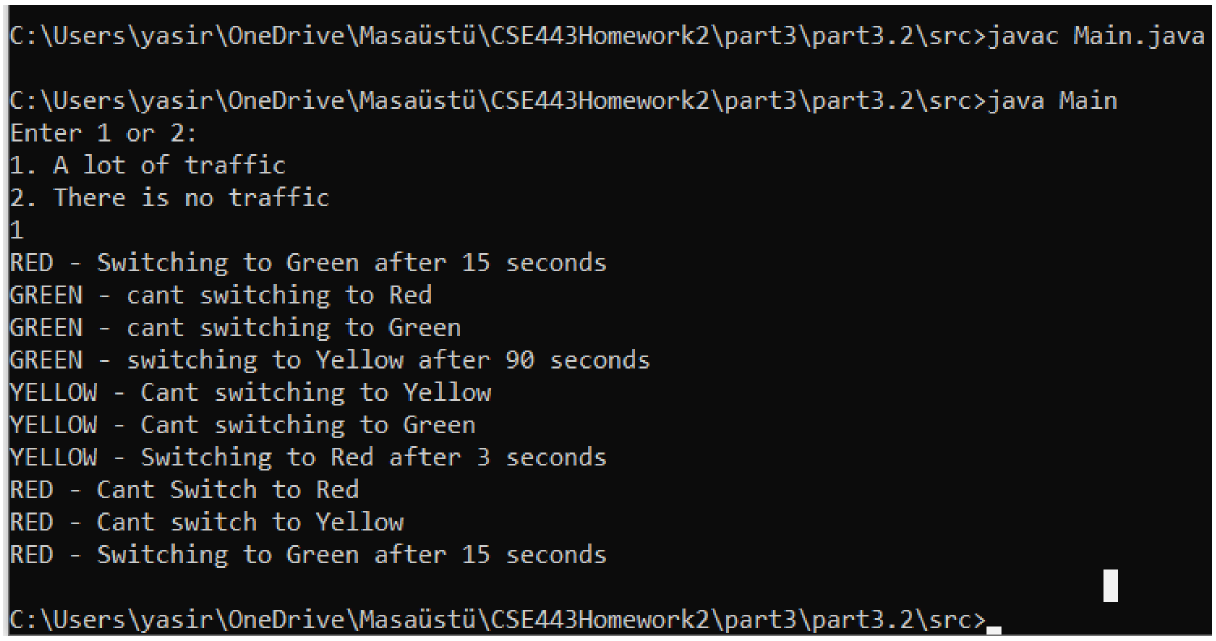
Here trafficLight initial state is red. Then I call switchGreen it will print red switching green after 15 seconds. Then I call switch red and switch green in green state. This will print cant switch because green state cant switch to red and green state. After calling switch yellow it will print switching yellow after 60 sec. Then in yellow state I try to switch yellow and green state but yellow cant switch those states so it will print cant switch yellow and green. Then I call switch red, so it will print switching red after 3 second and switch to red state. After red state I try to switch red and yellow but red cant switch red and yellow state so it will print cant switch those states. Then I call switch green. It will print switching green after 15 seconds because red can switch to green state and program terminated after showing all possible state changes.

In second part of question 3 I used observer pattern for HiTech camera. It notify trafficLight and change green state seconds



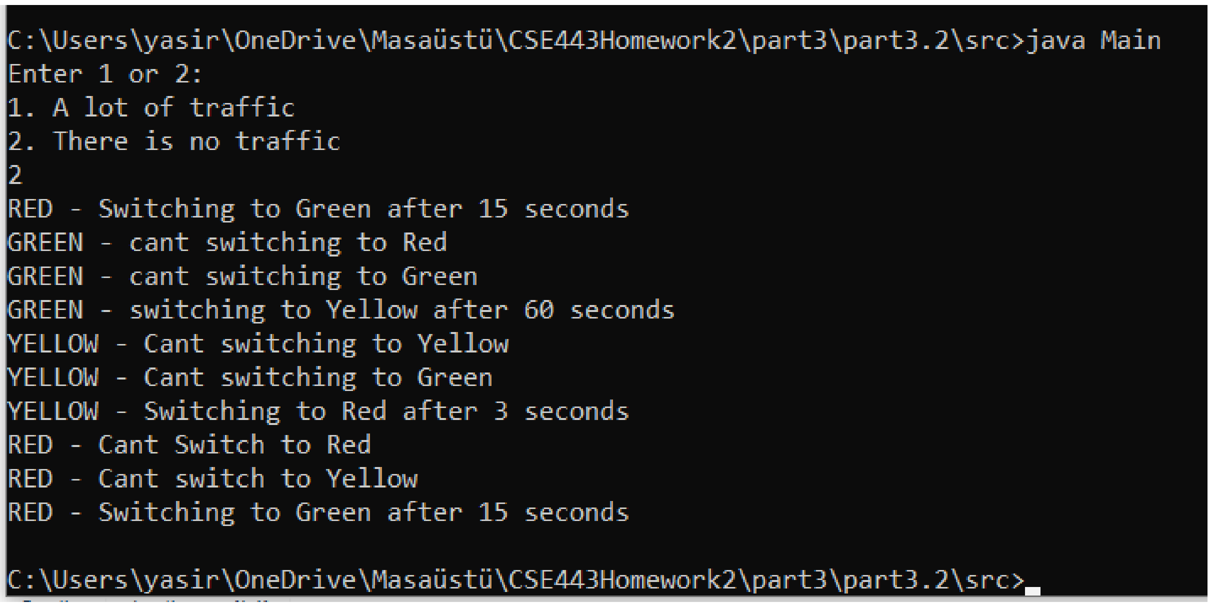
```
    public static void main(String[] args) {
        HiTech hiTech = new HiTech();
        TrafficLight trafficLight = new TrafficLight(hiTech);

        System.out.println("Enter 1 or 2:");
        System.out.println("1. A lot of traffic");
        System.out.println("2. There is no traffic");
        Scanner inp = new Scanner(System.in);
        int check = inp.nextInt();

        if(check == 1){
            hiTech.changeDetected( flag: true);
        }
        else{
            hiTech.changeDetected( flag: false);
        }


        trafficLight.switchGreen();
        trafficLight.switchRed();
        trafficLight.switchGreen();

        trafficLight.switchYellow();
        trafficLight.switchYellow();
        trafficLight.switchGreen();

        trafficLight.switchRed();
        trafficLight.switchRed();
        trafficLight.switchYellow();
        trafficLight.switchGreen();
    }
}
```

In driver, I ask user to enter 1 or 2. If its enter 1 camare change detected will be true then it will notify trafficLight and green seconds will be 90 otherwise it will be 60 seconds. After that I switch states like in first part of question 3.



```
C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.2\src>javac Main.java

C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.2\src>java Main
Enter 1 or 2:
1. A lot of traffic
2. There is no traffic
1
RED - Switching to Green after 15 seconds
GREEN - cant switching to Red
GREEN - cant switching to Green
GREEN - switching to Yellow after 90 seconds
YELLOW - Cant switching to Yellow
YELLOW - Cant switching to Green
YELLOW - Switching to Red after 3 seconds
RED - Cant Switch to Red
RED - Cant switch to Yellow
RED - Switching to Green after 15 seconds

C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.2\src>
```

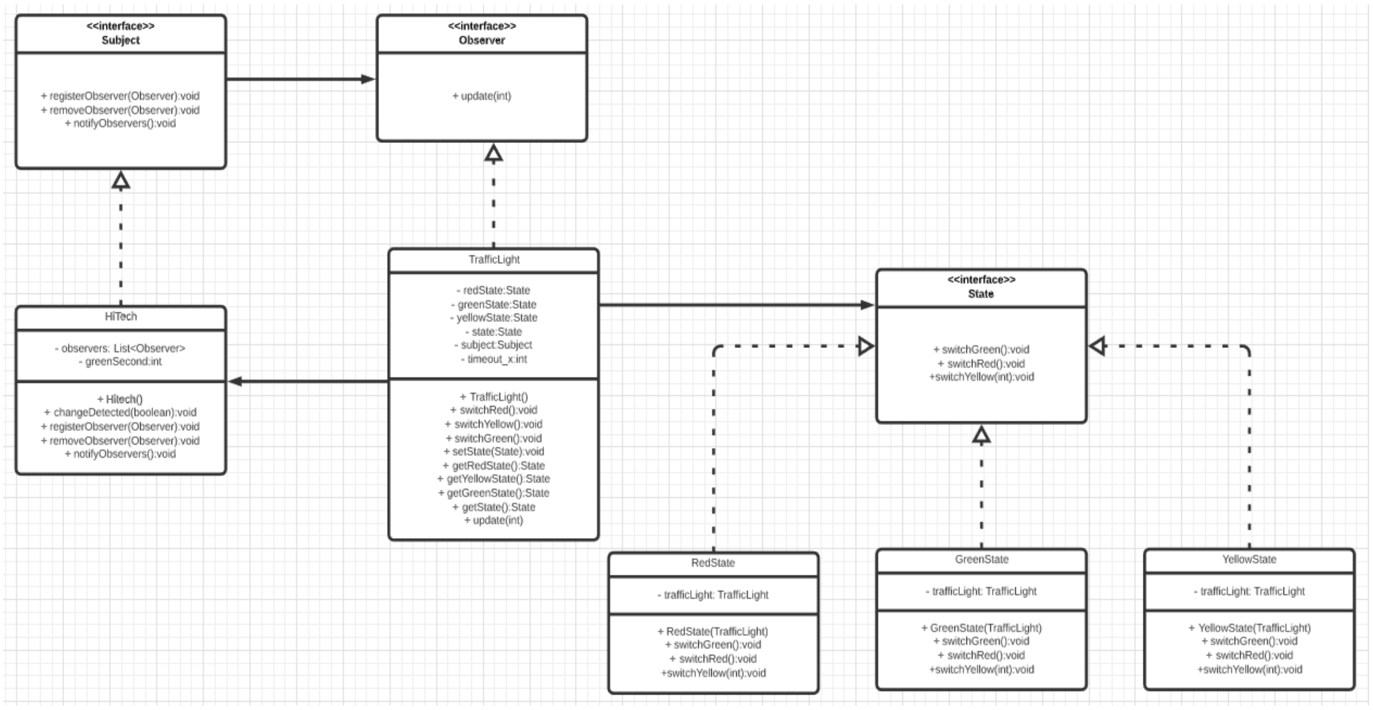
For example I enter 1 here, so changeDetect will be true and it will notify trafficLight. So green switching time will be 90 seconds as you can see.

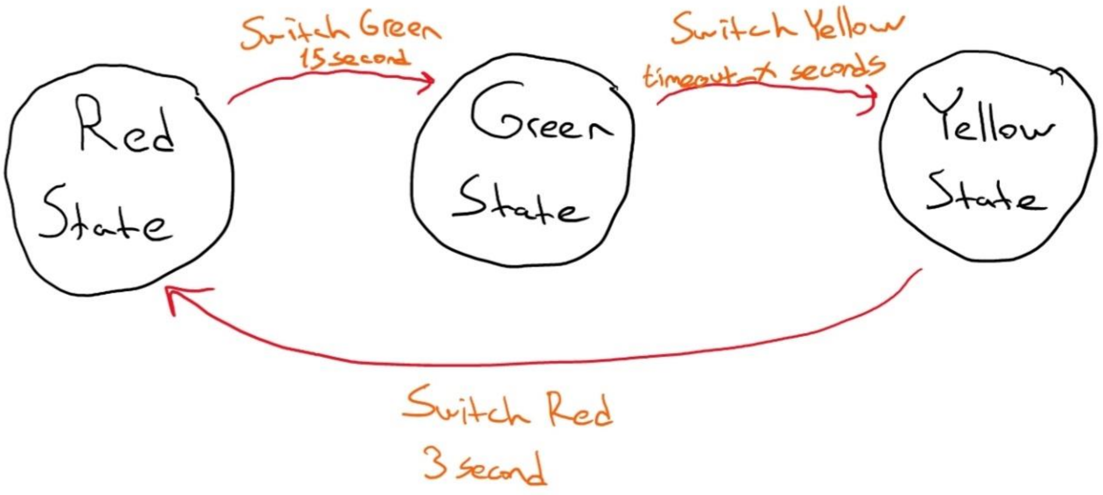


```
C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.2\src>java Main
Enter 1 or 2:
1. A lot of traffic
2. There is no traffic
2
RED - Switching to Green after 15 seconds
GREEN - cant switching to Red
GREEN - cant switching to Green
GREEN - switching to Yellow after 60 seconds
YELLOW - Cant switching to Yellow
YELLOW - Cant switching to Green
YELLOW - Switching to Red after 3 seconds
RED - Cant Switch to Red
RED - Cant switch to Yellow
RED - Switching to Green after 15 seconds

C:\Users\yasir\OneDrive\Masaüstü\CSE443Homework2\part3\part3.2\src>
```

If user enter 2 it means changeDetect will be false so trafficLight green switch will be 60 seconds like in screenshot.

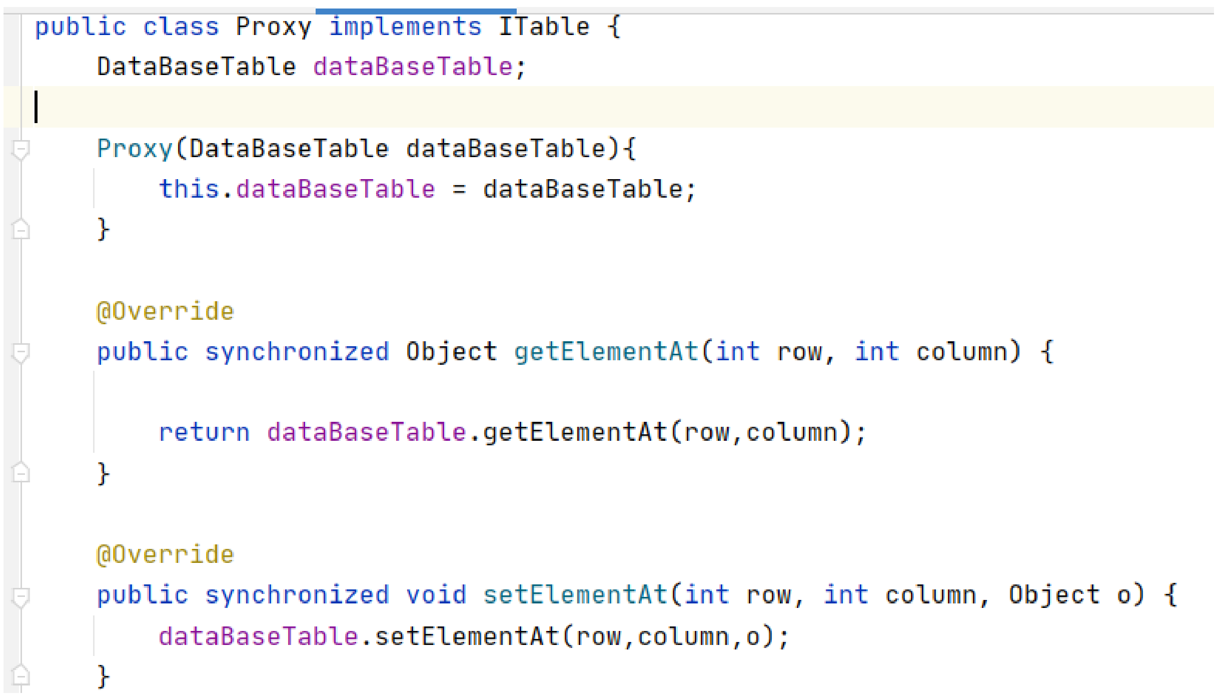
This is my class diagram and state diagram for part 2 of question 3:





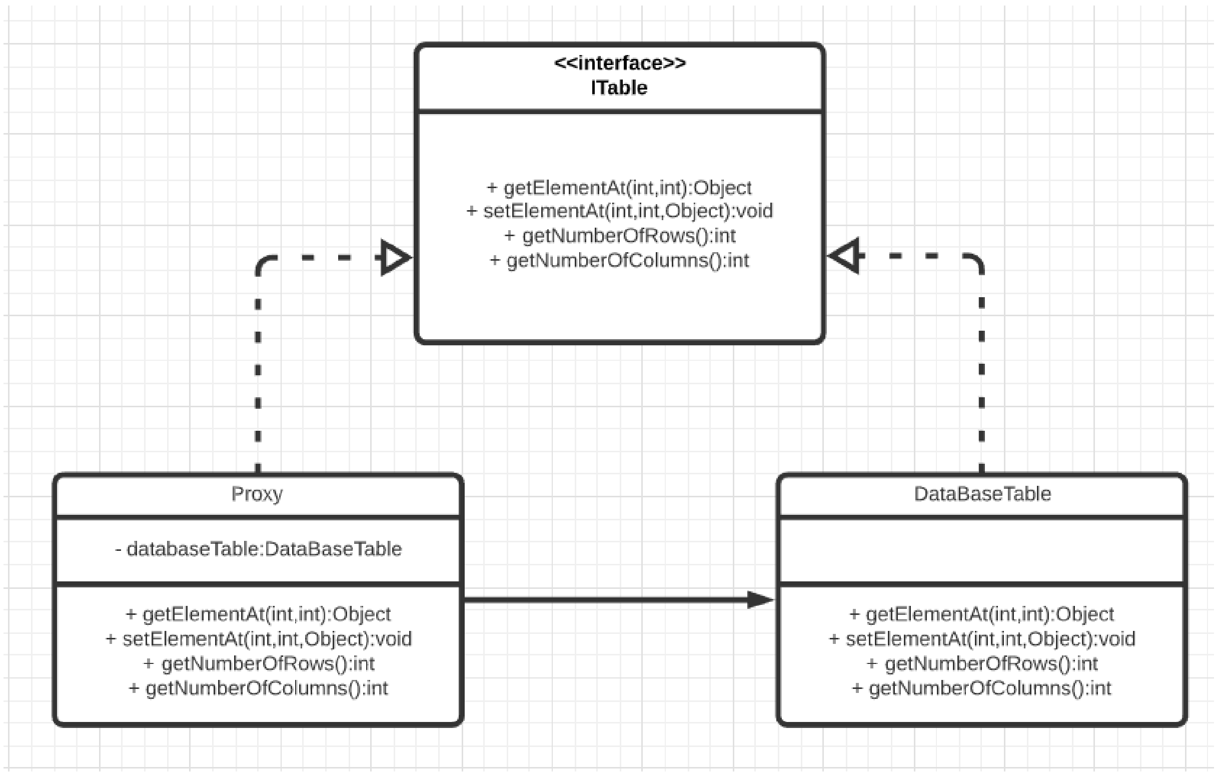
4-)

a) In this question I use Proxy pattern and synchronize getElementAt and SetElementAt methods in Proxy class. So when a thread in getElementAt or SetElementAt, other threads cant be in those methods. In methods I just call DatabaseTable methods but because of synchronize in Proxy class this getElementAt and SetElementAt methods have synchronization capability.



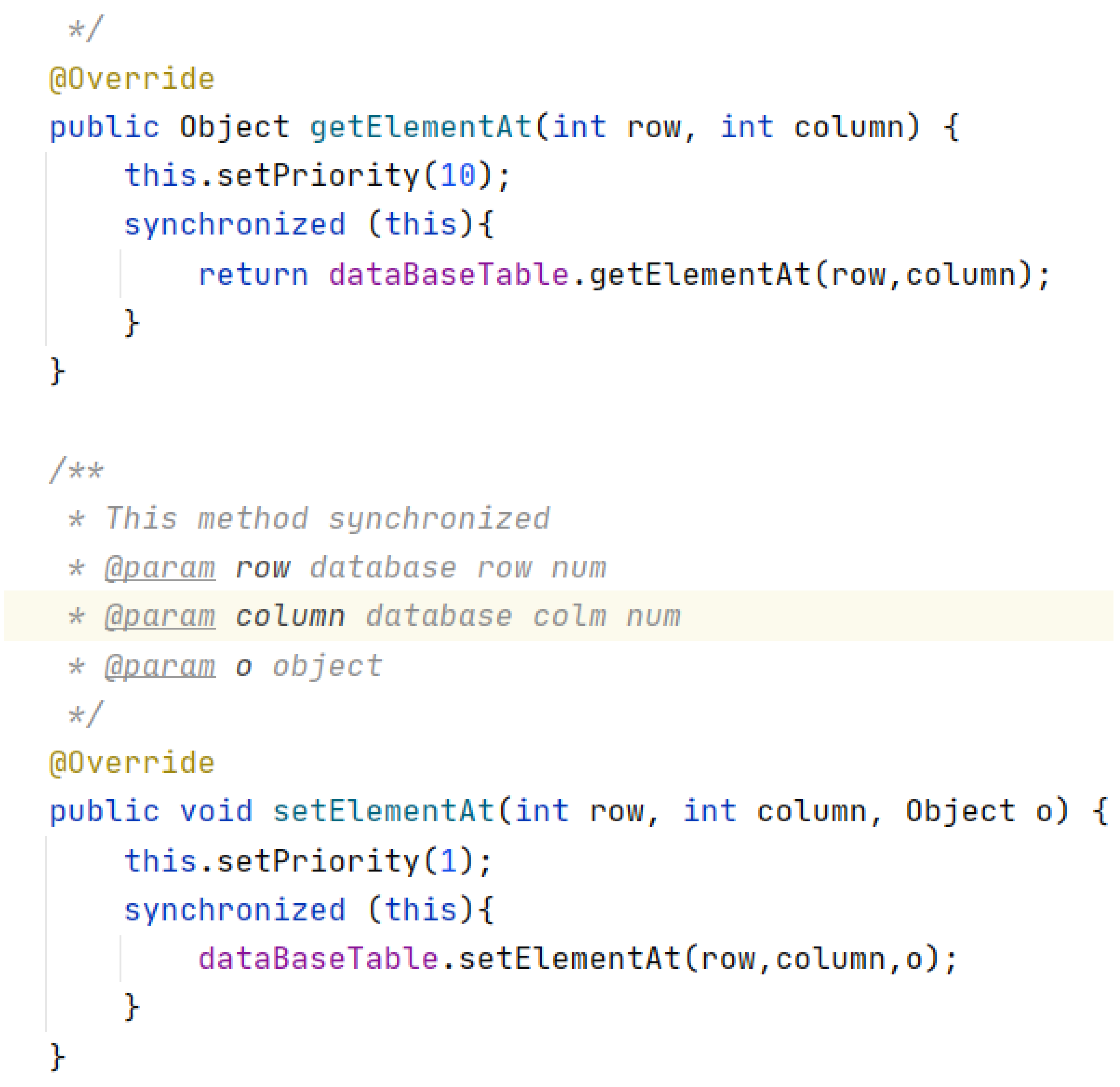
```java
public class Proxy implements ITable {
    DataBaseTable dataBaseTable;

    Proxy(DataBaseTable dataBaseTable){
        this.dataBaseTable = dataBaseTable;
    }

    @Override
    public synchronized Object getElementAt(int row, int column) {

        return dataBaseTable.getElementAt(row,column);
    }

    @Override
    public synchronized void setElementAt(int row, int column, Object o) {
        dataBaseTable.setElementAt(row,column,o);
    }
```

Class diagram fort his design:



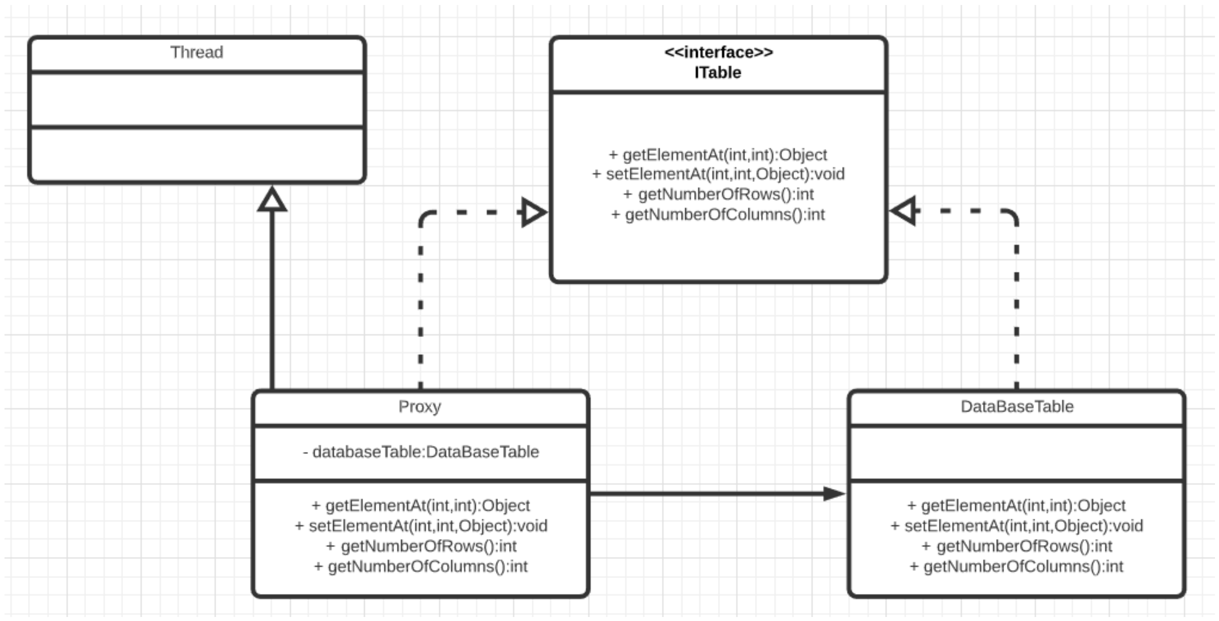
<<interface>>
ITable
+ getElementAt(int,int):Object
+ setElementAt(int,int,Object):void
+ getNumberOfRows():int
+ getNumberOfColumns():int
Proxy
- databaseTable:DataBaseTable
+ getElementAt(int,int):Object
+ setElementAt(int,int,Object):void
+ getNumberOfRows():int
+ getNumberOfColumns():int
DataBaseTable
+ getElementAt(int,int):Object
+ setElementAt(int,int,Object):void
+ getNumberOfRows():int
+ getNumberOfColumns():int

b) For priority of writers of DataBase, in Proxy class I change my synchronized method to synchronize blocks and by using java Thread class setPriority method I change priority of setElementAt method to 1 and getElementAt method to 10. By default their priority was 5. So with this change SetElementAt method has a priority in Thread run against getElementAt



```java
  */
@Override
public Object getElementAt(int row, int column) {
    this.setPriority(10);
    synchronized (this){
        return dataBaseTable.getElementAt(row,column);
    }
}

/**
 * This method synchronized
 * @param row database row num
 * @param column database colm num
 * @param o object
 */
@Override
public void setElementAt(int row, int column, Object o) {
    this.setPriority(1);
    synchronized (this){
        dataBaseTable.setElementAt(row,column,o);
    }
}
```

For using setPriority method now my Proxy class also extends java Thread class so my class diagram became like this:


Thread
<<interface>>
ITable
+ getElementAt(int,int):Object
+ setElementAt(int,int,Object):void
+ getNumberOfRows():int
+ getNumberOfColumns():int
Proxy
- databaseTable:DataBaseTable
+ getElementAt(int,int):Object
+ setElementAt(int,int,Object):void
+ getNumberOfRows():int
+ getNumberOfColumns():int
DataBaseTable
+ getElementAt(int,int):Object
+ setElementAt(int,int,Object):void
+ getNumberOfRows():int
+ getNumberOfColumns():int

Muhammed Yasir Fidan
161044056