Week 5 Homework Q32

Telmen Enkhbold

San Fransico Bay University

CE480 - Java and Internet Application

Dr. Chang, Henry

10/12/2023

Author Note

The Question

Elevator on a selected floor. The constructor requires exactly one argument to represent the starting floor. Include a select operation that allows floors to be selected. For every floor, the message "going up" or "going down" should be displayed before the current floor of the elevator. Here is one sample output to give you an idea of what simulated elevators will look like on your
screen:

```
start on floor 1.
going up to 2
going up to 3
going up to 4
open at 4
```

The diagram that represents the Elevator class is

```
 ---------------------------------------------------------------------
 | Elevator                                                          |
 |   int initFloor;    // Initial floor. Default value is 1          |
 |   int currentFloor; // Current floor number. Default value is 1   |
 ---------------------------------------------------------------------
```

This class must have 4 types of member functions:
- Manager functions
- Access functions (get, set, predicate)
- Helping functions
- Implementor function
  + The "select" implementor member function is defined such that each floor passed produces the output "going up to " or "going down to " followed by the appropriate floor number. When the selected floor has been reached, display the message "Open at " followed by the selected floor. For example, the output generated by the main function

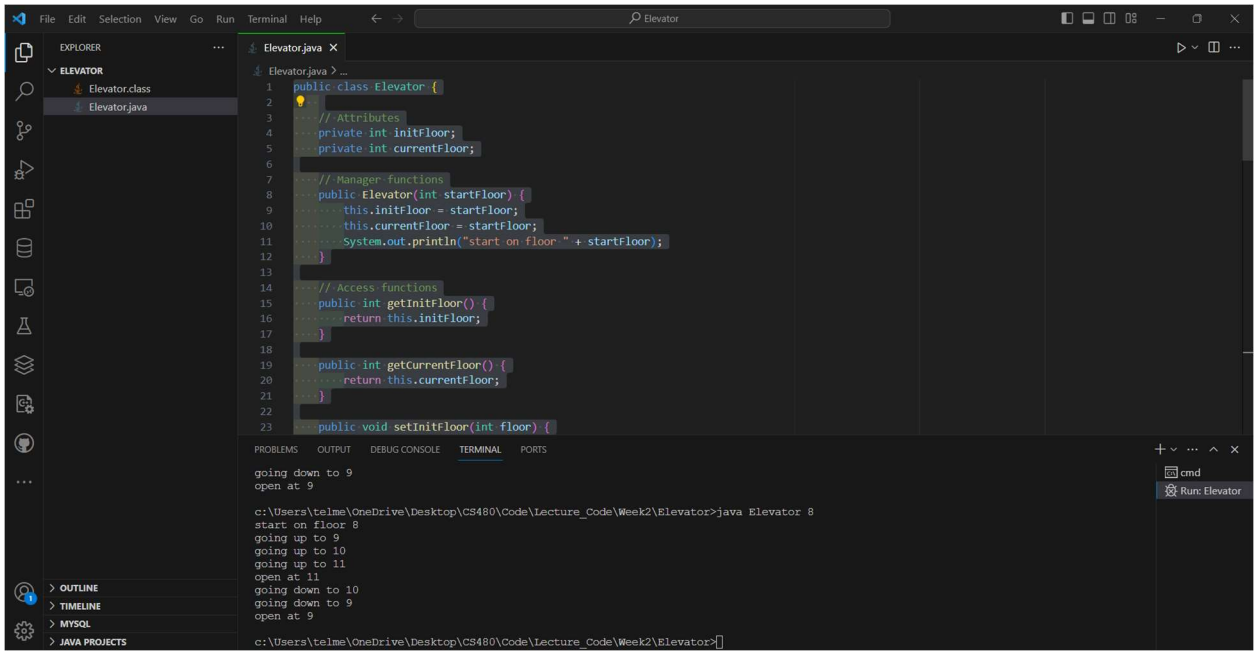
```
int main() {
   Elevator a(7); // Display "start on floor 7"
   a.select(11);  // Display "going up to 8"
                  //         "going up to 9"
                  //         "going up to 10"
                  //         "going up to 11"
                  //         "open at 11"
   a.select(9);   // Display "going down to 10"
                  //         "going down to 9"
                  //         "open at 9"
   return 0;
}
```

    are

```
start on floor 7
going up to 8
going up to 9
going up to 10
going up to 11
open at 11
going down to 10
going down to 9
open at 9
```

It looks like there is there needs to be one argument for the initial floor then the rest is coded in. I just need to keep in mind:

- Manager functions
- Access functions (get, set, predicate)
- Helping functions
- Implementor function



The Source Code

```java
public class Elevator {

    // Attributes
    private int initFloor;
    private int currentFloor;

    // Manager functions
    public Elevator(int startFloor) {
        this.initFloor = startFloor;
        this.currentFloor = startFloor;
        System.out.println("start on floor " + startFloor);
    }

    // Access functions
    public int getInitFloor() {
        return this.initFloor;
    }

    public int getCurrentFloor() {
        return this.currentFloor;
    }

    public void setInitFloor(int floor) {
        this.initFloor = floor;
    }

    public void setCurrentFloor(int floor) {
        this.currentFloor = floor;
    }

    // Predicate function to check if the elevator is on a specific floor
    public boolean isOnFloor(int floor) {
        return this.currentFloor == floor;
    }

    // Helping functions
    private void moveUp() {
        this.currentFloor++;
        System.out.println("going up to " + this.currentFloor);
    }
```

```
    private void moveDown() {
        this.currentFloor--;
        System.out.println("going down to " + this.currentFloor);
    }

    // Implementor function
    public void select(int targetFloor) {
        while (this.currentFloor != targetFloor) {
            if (this.currentFloor < targetFloor) {
                moveUp();
            } else {
                moveDown();
            }
        }
        System.out.println("open at " + this.currentFloor);
    }

    // Main function to demonstrate the behavior
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Please provide the initial floor as an
argument.");
            return;
        }

        try {
            int initialFloor = Integer.parseInt(args[0]);
            Elevator a = new Elevator(initialFloor);
            // For demonstration purposes, the select calls are hardcoded
            a.select(11);
            a.select(9);
        } catch (NumberFormatException e) {
            System.out.println("Invalid input. Please provide a numeric
floor value.");
        }
    }
}
```

Reference