

CS 4773

Homework Assignment 4: An elevator simulator

Due: 11:59pm Sunday, December 5

Note: You may work in teams of up to 4 people. Put the names of your team members in the REAME.TXT file described below. Your code must compile and run successfully on the Computer Science department's Fox servers using your make file and shell script.

You may submit as many times as you wish up to the deadline. Only the last submission by the deadline will be graded. However, all Blackboard submissions for your team must be from the same team member.

DESCRIPTION

In this assignment, you will implement an Elevator. The elevator can travel to floors 1, 2, or 3. It has 3 buttons: 1, 2, and 3.

Pressing 1 when on floor 1 outputs “Nothing happens”. Pressing 2 or 3 when on floor 1 moves the elevator to that floor.

Pressing 2 when on floor 2 outputs “Nothing happens”. Pressing 1 or 3 when on floor 2 moves the elevator to that floor.

Pressing 3 when on floor 3 outputs “Nothing happens”. Pressing 1 or 2 when on floor 3 moves the elevator to that floor.

When an elevator arrives at its destination floor, the doors open. Pressing buttons 1, 2, or 3 close the doors.

The elevator will start on floor 1 with the doors open when your program launches. Have your program output messages whenever a button is pushed, the elevator starts to move up or down, the doors open or close, and the elevator arrives at a new floor. Also output a “Nothing happens” message when pressing a button does not move the elevator.

You must use the State and Mediator patterns in this assignment. You must submit a UML state diagram of your implementation of the State pattern, AND a UML class diagram of your implementation of the State and Mediator patterns clearly identifying your State and Mediator patterns.

Use our clean-coding standards. Keep your class sizes ≤ 200 LOC and functions ≤ 20 LOC.

Create a make file named Makefile that builds your project. Your make file should place all of your class files in one or more jar files and delete the existing class files from the file system.

Create a bash script in a file named hw4 that runs your program. Make sure (by using `ls -l`) that your hw4 script is executable. If it is not, use the `chmod` command to make it executable.

Example:

The following statements in a main method:

```
elevator.press2();  
elevator.press1();  
elevator.press3();  
elevator.press3();  
elevator.press1();
```

Should produce output similar to this:

```
2 pressed
Doors are closed
Going up...
*dینگ* The elevator arrives at Floor 2
Doors are open
1 pressed
Doors are closed
Going down...
*dینگ* The elevator arrives at Floor 1
Doors are open
3 pressed
Doors are closed
Going up...
*dینگ* The elevator arrives at Floor 3
Doors are open
3 pressed
Doors are closed
Nothing happens
1 pressed
Going down...
*dینگ* The elevator arrives at Floor 1
Doors are open
```

DELIVERABLES

A zip file named abc123.zip using the ID of one of your team's members. This zip file should contain:

A UML class diagram named assignment4UMLclass

A UML state diagram named assignment4UMLstate

A README.TXT file containing the names of all of your team members

A directory containing your source code

A make file named Makefile to build your project

A bash script named hw4 to run your project

RUBRIC

20 points: All functionality implemented correctly including make file and bash script

20 points: All patterns implemented correctly

20 points: Classes and responsibilities are well designed

20 points: Code is readable. No Javadoc is required

20 points: UML class and state diagrams