Software and Programming II - Coursework Three
2014-15

Please see the Moodle site for submission dates and conditions.

Assignment Overview

This assignment will give you more experience on the use of:

1. classes,
2. class methods,
3. object-oriented programming in general, and
4. working in teams (pairs).

In this assignment, you are going to simulate a simple elevator. You will also show how different strategies can affect the efficiency of an elevator.

Task

Your task is to implement the simple elevator in Java. You should implement your own "efficiency" strategy for dealing with customers. The default strategy is the simple "start at the bottom, go to the top, then go to the bottom". Can you write a better strategy, one that is more efficient?

Project Description / Specification

1. Create three classes: `Building`, `Elevator`, and `Customer`.
2. Equip the building with an elevator. Ask the user to decide upon the number of floors and the number of customers.
3. Your program should have error checking to make sure the user inputs are valid. For example, if a user gives non-integer inputs, notify the user that the inputs are incorrect and prompt again.
4. Each customer starts from a random floor, and has a random destination floor.
5. We are presuming we are in the USA so there is no floor numbered **thirteen** (should you building be that high)**.**
6. Each customer will use the elevator only once, i.e., when a customer leaves the elevator, he/she will never use it again.
7. When all customers have reached their destination floor, the simulation is finished.
8. Part of the grade on this project will be the appropriateness of your classes, methods, and any exceptions you use. The quality of the code matters and the performance.
9. All the classes require full `javadoc`, including the `private` methods (should there be any).
10. Provide appropriate unit tests to check your methods.
11. Implement both your own strategy and the default strategy and then compare them. Your strategy does not have to be better (your grade will not depend on it) but the comparison is required.

Submission

1. You should create a `github` repository name ~~ew3~~ `sp2-cw3-2014` and *commit often*.
2. We will clone your `github` repo at the due date and time.
3. You should submit electronically via Moodle by the due date and time.
4. No binary class files please.
5. Only one member of the pair should submit the file(s) via Moodle; you should both have the code in a repository on `github`.

Notes and Hints:

Here are some suggested attributes and methods. Note that you needn't follow these faithfully.

Class	Attribute/Method	Explanation
Building	numOfFloors	The number of floors
	customerList	The list of customers
	elevator	The elevator equipped in the building

Class	Attribute/Method	Explanation
Elevator	NUM_OF_FLOORS	The number of floors
	registerList	The list of customers in the elevator
	currentFloor	The current floor of the elevator
	direction	The direction of the elevator
	move	The method to move the elevator by 1 floor
	customerJoins(customer)	A customer goes into the elevator
	customerLeaves(customer)	A customer goes out of the elevator

Class	Attribute/Method	Explanation
Customer	currentFloor	The current floor of the elevator
	destinationFloor	The destination floor of the elevator
	ID	The customer's ID
	inElevator	The flag to denote whether he/she is in the elevator

	finished	The flag to denote whether he/she has reached their floor

In your main method of the driving program, at the beginning ask the user for the number of floors and the number of customers which can be used to create an instance of `Building`. Then we only need to call the method to start the elevator and method to output information, repeatedly in a loop.

Randomly select the floors (to and from) for each customer. Use the appropriate method(s) from the `Random` class (look it up in the Java documentation)

To compare efficiency of strategy, count the number of floors visited for your strategy versus the default strategy