Computer Science 3620

Assignment 1 - Due Tuesday February 6 @midnight

Problem 1 - Decorator Pattern

Tom's Cruise Line is taking reservations for its spring 7-day Alaskan cruise. The ship offers four room types. Pricing is all per passenger based on double occupancy:

- **Interior** room - $1000
- **Ocean** view room - $1500
- **Balcony** room - $2000
- **Suite** - $3000

In addition, there are a number of add-on packages that passengers can purchase as part of their booking:

- **Beverage** package ($700)
- **Klondike** trail guided horseback tour ($50)
- **Helicopter** glacier tour ($200)
- Rock **climbing** lessons ($60)
- Admission to a **lumberjack** competition ($15)
- **Scientology** enlightenment tour ($500)

Room taxes are computed at 16% (per passenger), while add-on packages are taxed at 6%. A 15% per-person gratuity is added to the before-taxes total.

Write a GUI-based Java program, that uses the Decorator Pattern, to compute and display the charges based on user selections. The window should display the room options as radio buttons and the add-on packages as check boxes. Use simple labels to display the total taxes, total gratuity and overall total. You can use a "Compute" button if you wish to begin the computation process. You can assume the entire window relates to one passenger (not a room). You don't ned to deal with any other pasenger or payment information.

Include a UML class diagram showing your design (See submission instructions at the end of this assignment).

Problem 2[1] - Observer Pattern

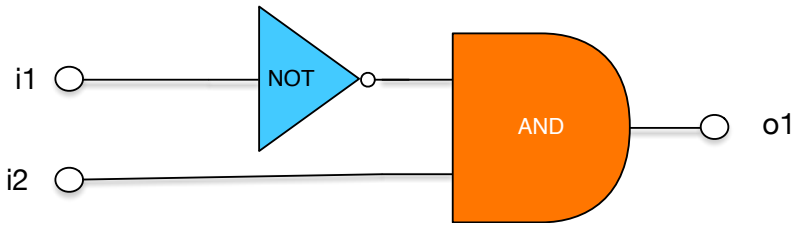
Suppose we are writing a simulator for digital circuits (assuming you have all taken CS 161 or equivalent). Refer to the set of partially-completed classes given at the end of this discussion.

The Gate class represents a logic gate. A gate has an output whose value is stored in state and returned by getState(). Values are boolean (false for 0, true for 1). Gate has three subclasses:

- AndGate simulates an "AND" logic gate with two inputs, each of which is connected to another gate's output. References to these two gates should be stored in input1, input2.
- NotGate simulates a "NOT" logic gate. It has one input, a reference to the Gate whose output is connected to the input. This should be stored in input.
- CircuitInput is a subclass of Gate but doesn't really represent a gate. It is used to represent inputs to the simulated digital circuit. The value of the circuit input is changed with the method setValue().

There is also a Probe class for measuring the value anywhere in the circuit. The gate whose output the probe is connected to should be stored in input. Probe is used to show a components output on the screen.

As an example of the usage, assume that we want to simulate the following circuit:



Also assume that the initial value of i1 and i2 is 0 and 1 respectively, and that the value of o1 should be displayed on the screen. This circuit can be represented by the following lines:

[1] (adapted from an assignment given at the University of Gothenberg, Sweden - author unknown)

```
CircuitInput i1 = new CircuitInput(false);
CircuitInput i2 = new CircuitInput(true);
Gate gate1 = new NotGate(i1);
Gate gate2 = new AndGate(gate1, i2);
Probe o1 = new Probe(gate2);
```

When one of the circuit input changes value, the change should be propagated through the circuit so that the correct value for each Probe is shown on the screen. To do this, at each update of a Gate's value, notify everything that is connected to its output.

When you solve the assignment, you can disregard the problems that arise when there are loops in the circuit. If each Gate only can be connected at creation and not reconnected later, as in the code above, then it is not possible to construct a circuit with loops. Also, do not pay any attention to the fact that the value of a Gate or Probe may be updated more than once for each update of a circuit input.

Your assignment is to modify and implement the classes along the design pattern Observer. Your assignment should include:

a) Class Diagram

Apply the design pattern for the given problem. The classes ConcreteSubject and ConcreteObserver in the diagram should be replaced with appropriate problem-specific classes. Draw a class diagram containing the classes Subject, Observer, Gate, CircuitInput, AndGate, NotGate, and Probe. The diagram should contain classes (with atributes and methods shown) and associations.

b) Code

Implement the classes Gate, CircuitInput, AndGate, NotGate and Probe. Don't forget to implement the update() method for the classes chosen to be Observers. You can make Gate inherit or implement some class or interface, but apart from this it should not be changed.

```
public class Gate
{
   protected boolean state;
   public boolean getState() {return state;}
}

public class CircuitInput extends Gate
{
   public CircuitInput(boolean initstate) { ... }
   public void setValue(boolean newstate) { ... }
}

public class AndGate extends Gate
{
   private Gate input1, input2;
   public AndGate(Gate inp1, Gate inp2) { ... }
}

public class NotGate extends Gate
{
   private Gate input;
   public NotGate(Gate inp) { ... }
}

public class Probe
{
   private Gate input;
   private boolean state;
   public Probe(Gate inp) { ... }
   public void show()
   {
      System.out.println(state ? "1" : “0”);
   }
}
```

Submission Guidelines:

Submit your assignment as a single zip file through the moodle link provided. Use the following naming format for your zip file:

lastname-as1.zip (e.g. howard-as1.zip)

The zip file should contain two folders, “Problem1” and “Problem2”, corresponding to the two questions above. Each folder should include all .java source files in a BlueJ project folder and PDF file containing the UML diagram. Please do not use packages in your code to create a hierarchical folder structure. Please DO use a UML editing tool to draw the UML diagrams (which will be turned into PDF files). Some examples are Tiny UML, UMLet, ArgoUML, Omnigraffle, etc.