Assignment 3: Decorator Pattern: Code from HFDP text: decorator.zip download

The winter holidays are over but the Christmas tree symbology still lives in our hearts. Being the organized individual you are, you have a plan for next year's holiday tree. Implement a software system that allows you to calculate the price of any tree plus any combination of decorations. The system must be easily extendable in the sense that whenever new decorations are added in the store you will have to at most add one class.

Here are two tables representing costs of trees and decorations, respectively. Use these data tables to obtain the output above.

Name	Cost
Fraser Fir	35
Douglas Fir	30
Balsam Fir	25
Colorado Blue Spruce	50

Name	Cost
Star	4
Ruffles	1
Balls Red	1
Balls Silver	3
Balls Blue	2
Ribbons	2
Lights	5
LEDs	10

A very important requirement is that a tree can only have one star. When a user wants to decorate a tree containing a star with a new star, you must print a warning that the tree already has a star and not add the price of a star to tree. Users must be able to continue decorating their tree if they attempt to add another star to it. The following code snippet (which you are NOT required to use on your assignment) illustrates the problem:

```
public class TreeTester {

    public static void main(String[] args) {
        HolidayComponent mySpruce = new BlueSpruce();
        mySpruce = StarDecorator.addStar(mySpruce);

        mySpruce = new RufflesDecorator(mySpruce);
        mySpruce = StarDecorator.addStar(mySpruce); // This should
fail to add a star
        mySpruce = new LEDsDecorator(mySpruce);
        mySpruce = new BallsRedDecorator(mySpruce);
        mySpruce = new LightsDecorator(mySpruce);
        mySpruce = StarDecorator.addStar(mySpruce); // This should
fail to add a star


        // making a second tree
        HolidayComponent myDouglas = new DouglasFir();
        myDouglas = new LEDsDecorator(myDouglas);......finish for all
trees
```

Output: should look something like this test for only 1 Star!

```
This tree already has a star!
Tree1: Fraser Fir decorated with, star, ruffles, ruffles, lights
costs: $46
Tree2: Colorado Blue Spruce decorated with, ruffles, Red Balls, Silver
Balls, ruffles, LEDs costs: $66
Tree3: Balsam Fir decorated with, ribbons, star, Blue Balls, ruffles,
LEDs costs: $44
Tree4: Douglas Fir decorated with, ruffles, star, LEDs, Red Balls
costs: $46
```

You should design your solution such that adding an additional star is not allowed. Furthermore, recognize that you need to control Star construction. Classes outside of Star should not be able to directly create Stars. While stars are decorations, they are special. Consider how you might avoid the problem in the sample above and implement an appropriate solution. An important consideration is what class should be responsible for enforcing a single star on a tree. Try and minimize the effect dealing with a star has on class relations in your solution. REMINDER: YOU ARE NOT REQUIRED TO USE THE SPECIFIC CODE ABOVE ON YOUR ASSIGNMENT -- IT IS THERE TO DEMONSTRATE A PROBLEM (WITH THE STAR CONSTRUCTOR BEING PUBLIC).

Submit a zip file to Canvas with precisely the following:

Submit a .pdf of your UML for your solution
Submit a folder that has your source code or package for your solution.
Name the source file or main file you use to test your solution TreeTester. This file should NOT be declared as part of any package. All other source files in the solution should be declared in a package called holiday_decorations. Thus TreeTester.java (which will contain your main method and you will use to decorate some trees and add some stars) should NOT be part of the holiday_decorations package. The folder holiday_decorations (which represents the holiday_decorations package) will be at the same 'level' as TreeTester.java. Inside the holiday_decorations folder will be all the classes/interfaces to represent your trees and decorations (i.e. Star.java, Ribbons.java, HolidayDecoration.java, etc.)

Using Java, TreeTester should be in the directory created by unzipping the submission.

Create an output file (notepad.text or generated from your code to an output file)