CS350 Software Design
SE310 Software Architecture I
Lab 5: StarBuzz Coffee Shop

Goal:

Practice Decorator Pattern and Strategy Pattern

Note: in this lab, instead of providing detailed instructions, I provide a requirement spec as you will see from a real project.

Requirement Specification

1. The StarBuzz Coffee shop offers both Coffee and Tea Beverages.
    a. The basic coffee beverages include: HouseBlend, Espresso, Decaf
    b. Other coffee beverages can be produced by adding the following ingredients to basic coffee beverages: Chocolate, Milk, Whip Cream.
    c. The basic tea beverages include: green tea, red tea, and white tea
    d. Other tea beverage can be produced by adding the following ingredients: Jasmine, Ginger , Milk
2. The name of a beverage implies the default ingredients within the beverage. Below are the names and ingredients listed in the StarBuzz Coffee Menu:
    a. **Espresso**: espresso coffee
    b. **Decaf**: decaf coffee
    c. **HouseBlend**: houseblend coffee
    d. **Mocha**: espresso + 1 serving of Chocolate
    e. **Latte**: espresso + 1 serving of Milk
    f. **Cappuccino**: espresso + 1 serving of whip cream
    g. **Decaf Mocha**: decaf coffee + 1 serving of Chocolate
    h. **Decaf Latte**: decaf coffee + 1 serving of Milk
    i. **Decaf Cappuccino**: decaf coffee + 1 serving of Whip Cream
    j. **Green Tea**: green tea
    k. **Red Tea**: red tea
    l. **White Tea**: white tea
    m. **Flower Tea**: green tea + Jasmine
    n. **Ginger Tea**: green tea + ginger
    o. **Tea Latte**: red tea + Milk

    The user can also order any combination of tea and tea ingredients, coffee and coffee ingredients.
3. Each beverage has three sizes: small, medium, and large. These are the sizes for basic beverages.
4. The software should take the order from the user in the following format:

    <beverage name> <size> [<ingredient 1, ingredient 2, ingredient 3>]

e.g. a small mocha with two serves of milk

Mocha small milk milk

or:

houseblend large whipcream

5. The software should calculate the cost of each order according to the size and the type of the beverage:
    a. The costs of basic beverages are:
        i. Espresso , Green Tea, White Tea: $1
        ii. Houseblend and Red Tea: $0.8
        iii. Decaf: $0.5
    b. The costs of ingredients
        i. Chocolate, Milk, Whip Cream: $0.3
        ii. Jasmine: $0.5
        iii. Ginger: $0.6
    c. The cost of sizes:
        i. Coffee beverages:
            1. Large: $1
            2. Medium: $0.7
            3. Small: $0.4
        ii. Tea beverages:
            1. Large: $0.7
            2. Medium: $0.5
            3. Small: $0.2

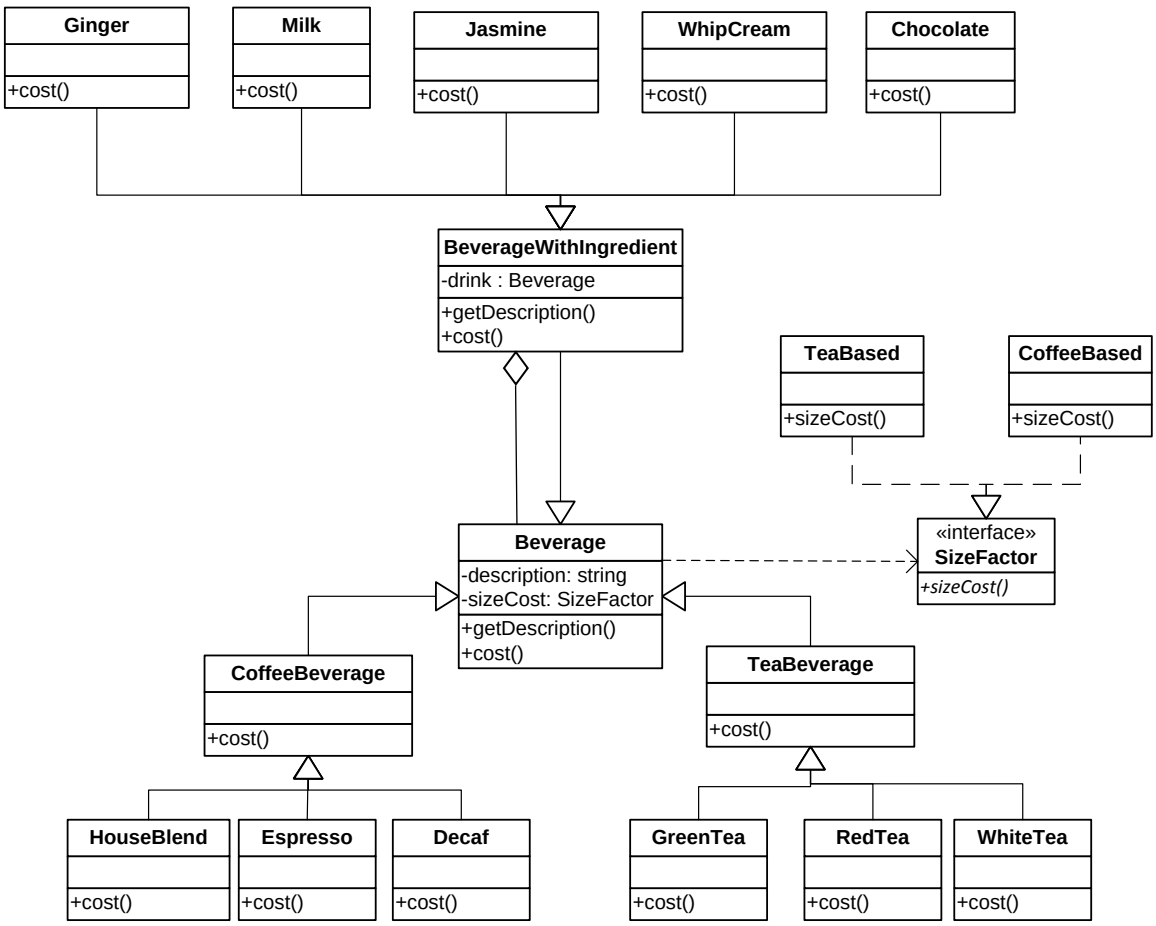
    The total cost of a beverage will be the cost of the basic beverage + the cost of ingredients + the cost of size.
6. The output format of the program should be:

    “The total cost of your order is: “ <cost>

Design Instructions:

1. Using decorator pattern combined with strategy pattern as we talked in class.

2. Implement the system according to the UML class diagram below. Note that the diagram mainly shows a structure. You can change the names and signatures of the operations


Ginger
+cost()
Milk
+cost()
Jasmine
+cost()
WhipCream
+cost()
Chocolate
+cost()
BeverageWithIngredient
-drink : Beverage
+getDescription()
+cost()
TeaBased
+sizeCost()
CoffeeBased
+sizeCost()
«interface»
SizeFactor
+sizeCost()
Beverage
-description: string
-sizeCost: SizeFactor
+getDescription()
+cost()
CoffeeBeverage
+cost()
TeaBeverage
+cost()
HouseBlend
+cost()
Espresso
+cost()
Decaf
+cost()
GreenTea
+cost()
RedTea
+cost()
WhiteTea
+cost()