Design Question 7

Xuanqi Wei 1009353209

October 29, 2023

1 Week 6 & 7 Design Question

The design I feel is more consistnet with the SOLID principles is the Design Number Two. My design question will include three parts as follows:

1. A discussion of Design Number Two with respect to the SOLID principles.
2. A discussino of how well Design Number Two satisfies the specification.
3. Based on the discussions above, I will suggest one way might further improve the design to better adhere to the SOLID principles, and one way I might better satisfy the specification.

1.1 A Discussion of Design Number Two with Respect to the SOLID Principles

The SOLID principle, as we learnt from the lecture are a set of five design principles for writing maintainable and understandable software. They are SRP(Single Respoonsibility Principle), OCP(Open/Closed Principle), LSP(Liskov Substitution Principle), ISP(Interface Segregation Principle) and DIP(Dependency Inversion Principle). I will evaulate the Design Number Two based on these principles.

1. SRP(Single Respoonsibility Principle): This principles states that 'A class should have only one reason to change'. For Design Number Two, as the diagram shows, the aim of the Builder class as shown in the UML Design Diagram is to show the topping that a BubbleTea might include, which only if I want to change the functionality they originally have, otherwise they wouldm't change for any other reason. Generally, as the handout stated, the design includes separate classes for handling order, tea flavors, ingredients and topping, which each class has a distinct responsibility.
2. OCP(Open/Closed Principle): This principle statest that 'software entities (class, modules, functions, etc.) should be open for extending but closed for modifiction'. For Design Number Two, as the diagram shows, there are two classes implementing an inteerface called Item, which ensure the functionality of adding other items to the BubbleTea by just writing another class that implements the interface Item, which keeps the getName and getCost. This achieve the need of supporting seasonal toppings without modiying the existing classes by introducing new classes for seasonal topping.

3. LSP(Liskov Substitution Principle): This principle states that 'Subtypes must be substitutable for their base types without altering the correctness of the program'. For Design Number Two, as the diagrams shows, the Ingredient subclass or a Topping subclass doesn't do anything more than what an Item does, which means there's nothing that only in Ingredient or only in Topping, which we can definitely replace object of type Item by Object of type Ingredient or Topping without error.

4. ISP(Interface Segregation Principle): This principle states that 'No code should be forced to depend on methods that it does not use'. For Design Number Two, as the diagram shows, there is only one interface and the functionality of that interface is simple and doesn't require to be broken into smaller sparts. Moreover, as the UML disgram and Sequence Diagram clearly show, none of the classes have methods that they doesn't use.

5. DIP(Dependency Inversion Principle): This principle states that 'High-level modules should not depend on low-level modules. Both should depend on abtractions'. For Design Number Two, I observe that these entityu classes do not depends on lower level ones, for example, Interactor, Contoller, and Presenters, which they satisfy the DIP.

1.2 A Discussino of How Well Design Number Two Satisfies the Specification

As the hoandout stated, firstly, it allows customers to order and personalized their bubble tea beverages. Secondly, bubble tea is a customizable drink that can contain a variety of tea flavors. Thirdly, the customers can select from a menu of preset bubble tea drinks or create a drink of their own choosing. Fourhtly, the company will introduce seasonal toppings. Fifthly, there is a basd price associated with a bubble tea order as well as for each additional topping and ingredient.

1. Allows Customers to Order and Personalized Their Bubble Tea Beverages: For Design Number Two, based on the diagram shows, the BubbleTEACafe firstly creates a TeaFlavour, then uses the Builder to further create a Bubble Tea. Therefore, if needed, the customization can be applied when creating one or more Ingredient or Topping. Though this process, the Buidler then builds the BubbleTea needed and the price is therefore shown by obtaining the price of all Item and the TeaFlavour.

2. Bubble Tea is a Customizable Drink That Can Contain a Variety of Tea Flavors: As shown in the diagram, the BubbleTea class has a teaFlavour variable and a variable called customization - a List containing Item inside, which includes Ingredient and Topping as needed.

3. Customers can Select from a Menu of Preset Bubble Tea Drinks or Create a Drink of Their Own Choosing: As shown in the diagram, the StandardDrinks class has the functionality of make the menu of preset bebble tea drinks, which has greenTeaWithPearls and blackTeaWithMilkAndJelly preset. Moreover, to create a drink of the customer's own choosing, the builder class can create a BubbleTea based on the name, basePrince and teaFlavour as shown.

4. Company Will Introduce Seasonal Toppings: As stated in the first part in the OCP section, it's doable and the variable called name inside the Topping class can indicate if it's a seasonal topping.

5. There is a Basd Price Associated With a Bubble Tea Order As Well As For Each Additional Topping and Ingredient: As shown in the diagram, there is a bariable called basePrice inside the BubbleTea class and the customization variable which includes Ingredient and Topping stated before also has a cost variable, ensuring the purpose in achineved.

1.3 Improvements

I will suggest one way might further improve the design to better adhere to the SOLID principles, and one way I might better satisfy the specification.

1. SOLID Principles: As the SRP(Single Respoonsibility Principle) states, A class should have only one reason to change. I suggest that to create a new class called CalculatePrice which purpose is to calculate the price of the BubbleTea rather than using the calculatePrice method inside the BubbleTea class. The reason of the suggestion is because it can better satisfy the SRP in SOLID because now it only change for the former but prevously, both the change of BubbleTea and the way of calculating the price will cause the change.

2. Specifications: When relating the diagram to the Specification, we easily found that the BubbleTea and the TeaFlavour both have a price by observeing the vairables cost in TeaFlavour and basePrice in BubbleTea. However, in the specification, it only states that there is a base price associated with a bubble tea order as well as for each additional topping and ingredient without specifically mention the cost of tea flavour. To improve this, we can add some addition messages of what they actually are by stating them respectively and specifically, which can make it better satisfy the specification.