Lab Report for Software Engineering course Lab 4: Starbubucks coffee online retailing system v3.0

Wang, Chen	Liu, Jiaxing	Huang, Jiani	Tang, Xinyue
16307110064	17302010049	17302010063	16307110476

School of Software
Fudan University

April 15, 2019

Contents

Chapter 1

Overview of this lab

1.1 The Objectives of the Project

1.2 Specifications of the Lab

1.3 The division of work in the team

Chapter 2

Improved skills on team collaboration

2.1 Consistent git commit message styles

2.1.1 Commit message requirements

The following are the requirements for the commit message in our team, this version of specifications are revised according to the commit message style recommendation from website *Commit message guidelines · GitHub*[1], *How to Write a Git Commit Message*[2], *How To Write a Good Commit Message*[3] and the git commit message recommendation from one of the most authoritive open source project **GNOME** *Guidelines for Commit Messages*[4].

1. Only ASCII characters are allowed in the entire commit message
2. All commit messages must start with one of the types identified in the following table, all words are lowercase
3. It is best to have an associated work item, associated with the work item, followed by type, space # number space followed by content, such as fix #123 content
4. The total number of characters recommended in the subject *(note that it is the number of chars instead of the number of words)* is less than 50, and the maximum number is not more than 74 characters (including the previous type and item number, etc.)
5. There is no need to add a period at the end of the head
6. After the type in the subject line, the first letter of the first word after the task number (if any) is capitalized and that indicates the beginning of a sentence

[1]`https://gist.github.com/robertpainsi/b632364184e70900af4ab688decf6f53`
[2]`https://chris.beams.io/posts/git-commit/`
[3]`https://api.coala.io/en/latest/Developers/Writing_Good_Commits.html`
[4]`https://wiki.gnome.org/Git/CommitMessages`

7. Use the imperative tone in the subject sentence (although it is you who have actually done the work)
8. The tense of the subject is the general present tense
9. It is recommended that for commits involving complex modifications, body should be added in addition to the subject for further explanation. The method is as follows: break a new line and then write is the body, [do not follow the head without line break]
10. For the body part of the commit message, there is no requirement other than writing Chinese, and it is relatively free. It is also recommended to write more than one line instead of one line in order to facilitate reading.

2.1.2 Types allowed in the subject of the commit message

Table 2.1: Commit message types

Type	**Description**
feat	A new feature
fix	A bug fix
wip	While working on a fix/feature
docs	Documentation only changes
style	Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)
refactor	A code change that neither fixes a bug or adds a feature
test	Adding missing tests
chore	Changes to the build process or auxiliary tools and libraries such as documentation generation

Chapter 3

Structure of the project

Chapter 4

Implementation of the features

4.1 Entities implementation

The implementation of entity package can be divided into two packages: the package of drinkEntity and the one of ingredients.

To our attention, I apply this.getClass.getName() to simplify the constructor by avoiding inputting string every time.

4.1.1 drinkEntity

At present we have four different kinds of drinks: Cappuccino,Espresso,GreenTea and RedTea. But considering the possibilities of adding other kinds of coffee and tea and the rationality of logic, the abstract classes of Coffee and Tea are applied to construct the whole inheritance relationship.That is, Both Coffee and Tea will inherit the OrderItem class, and all specific kinds of coffee and tea will inherit the abstract classes of Coffee and Tea.

We should pay attention to the constructor method of the different concrete classes. Since the initialization only need the different basic price of drinks, we only need to call the setPrice() method in constructor method. And all the generic method of these classes will be implemented in the parent class, the OrderItem class, such as cost() and size2price() methods. Thanks to the inheritance tree, in the size2price() method, we can use the instanceof keyword to simplify the judge of cup-size.

4.1.2 ingredientEntity

It is evident that the four concrete ingredients inherit the Ingredient class in dto package: chocolate, cream, milk and sugar. The constructor like the above drinkEntity's only need to set the price.

4.1.3 OrderItem

I add two methods: size2price and cost in this class in order to make all its subclasses use this method.

4.2 Order services implementation

Chapter 5

Testing of the features

5.1 Testing methods adopted

5.2 Testing results

Bibliography

[1] Wikipedia contributors. (2019, March 22). JUnit. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:53, April 1, 2019, from `https://en.wikipedia.org/w/index.php?title=JUnit&oldid=888928403`