**Hong Kong Institute of Vocational Education**

**(Tuen Mun)**

**Department of Information Technology**

**HD in Software Engineering (IT114105)**

**Module Code: ITP4507**

**Contemporary Topics in Software Engineering**

**Assignment Report**

**Date: 13-11-2020**

**Student Name: Cheung King Hung**

**Student ID: 190001623**



|  |
| --- |
| **Class:2A** |



Contents

[1. Assumptions regarding the problem context 1](#_Toc56119621)

[2. Application design with class diagram 2](#_Toc56119622)

[3. Discussion and explanation on each of the design patterns applied to the application. 4](#_Toc56119623)

[Command Pattern: 4](#_Toc56119624)

[Factory Method Pattern: 5](#_Toc56119625)

[Memento Pattern: 6](#_Toc56119626)

[4. User Guide 7](#_Toc56119627)

[5. Test Plan and Test Cases 12](#_Toc56119628)

[6. Well documented Source Code 22](#_Toc56119629)

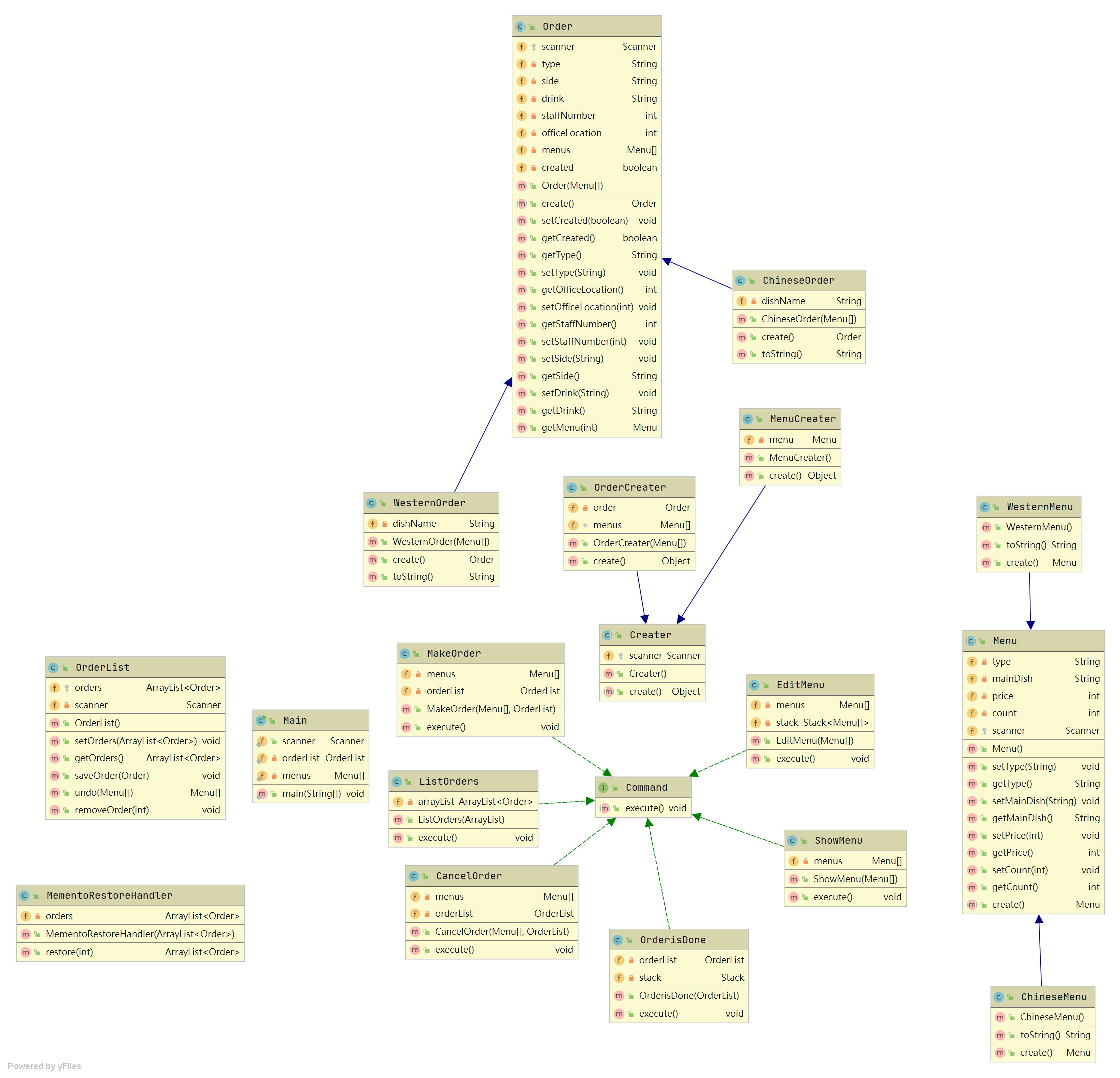
# **Assumptions regarding the problem context**

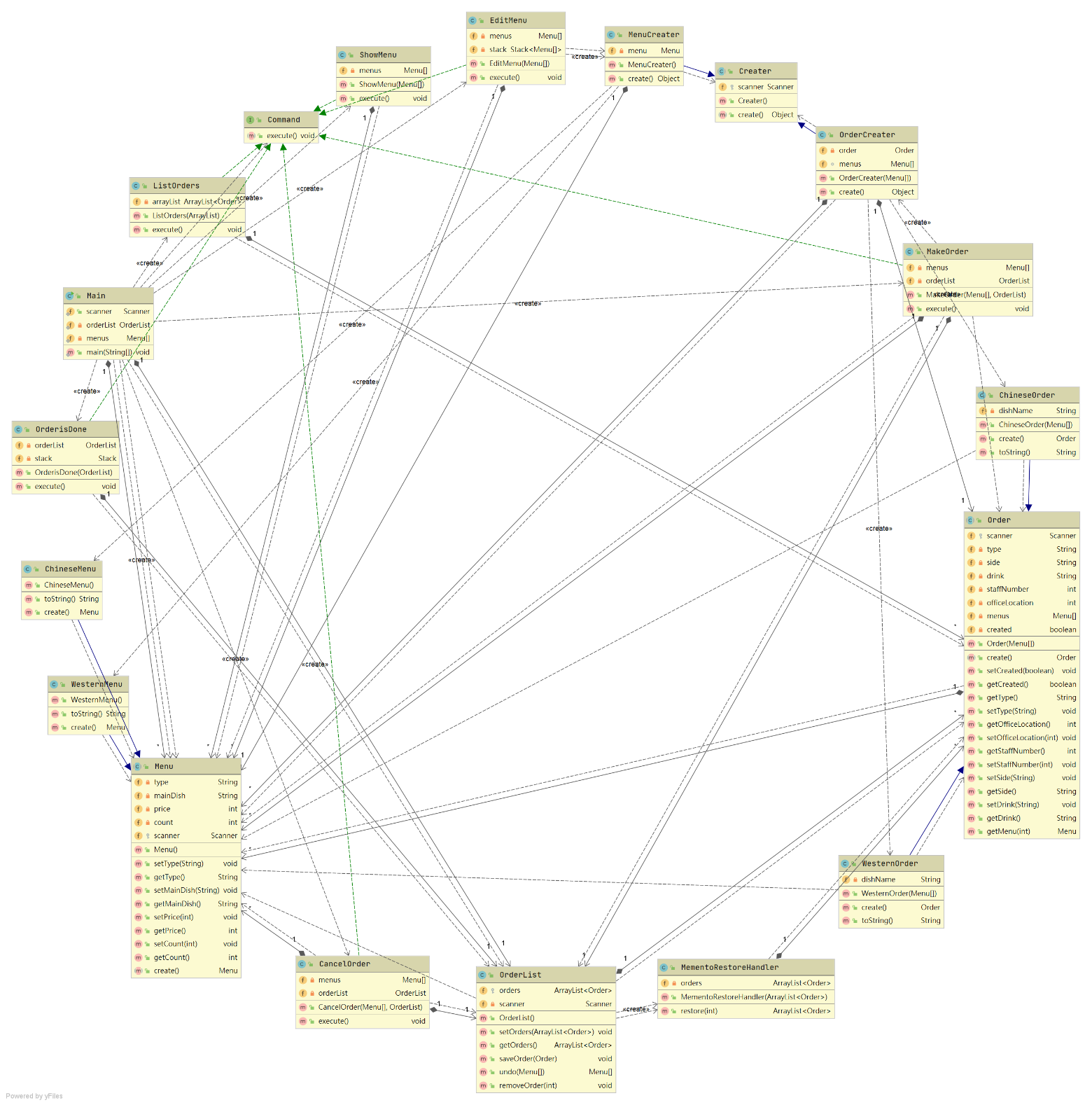
The main dish of various lunch sets changes every day, and the lunch price set may also change according to the main dish.

And the company may consider adding more lunch sets.

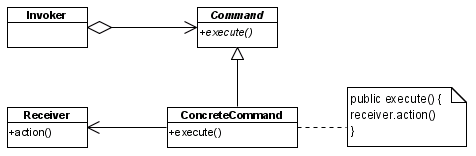
Therefore, the design should follow the open-closed principle provides flexibility and scalability for future improvements.

# **Application design with class diagram**





# **Discussion and explanation on each of the design patterns applied to the application.**

Command Pattern: 

Advantage of Command Pattern:

It decouples the classes that invoke the operation from the object that knows how to execute the operation

Extensions to add a new command is easy and can be done without changing the existing code

You can also define a rollback system with the Command pattern, for example, in the Wizard example, we could write a rollback method

Command Pattern in my project:

The Main Class is the Invoker. The Invoker holds a command and can get the Command

Command Class is the interface. The command is an object that encapsulates a request to the receiver.

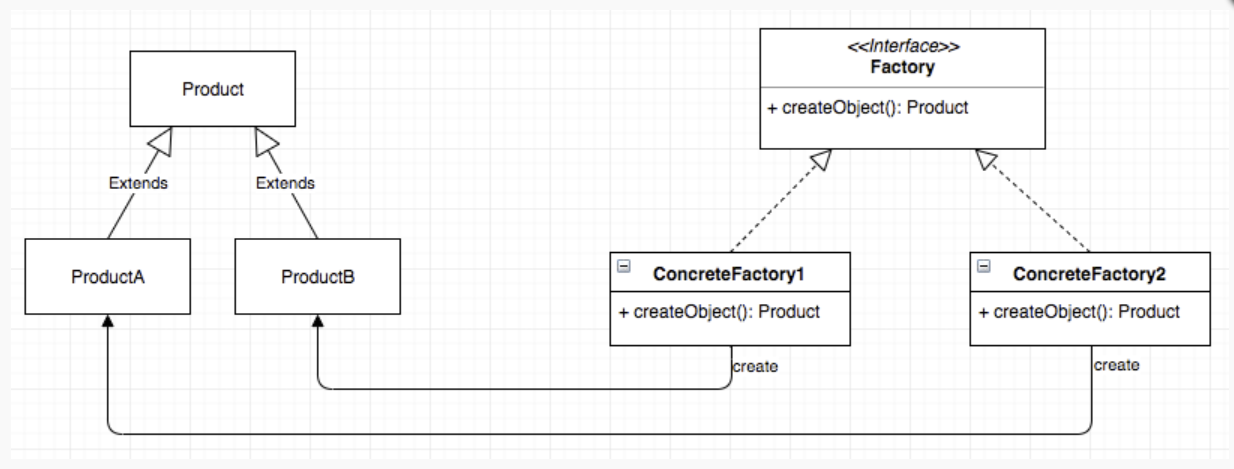
ListOrders Class is the Concrete Command. The ConcreteCommand defines a binding between the action and the receiver.

OrderList Class is the Receiver. The receiver is the component that is acted upon by each request.

In case of call function:

After the user enters the command, the main class will according to the command create a new object which is handling the function and it is implemented the Command interface.

## Factory Method Pattern:



Advantage of Factory Design Pattern:

The factory method pattern allows subclasses to choose the type of object to be created.

It eliminates the need to bind application-specific classes to the code, thereby promoting loose coupling. This means that the code only interacts with the resulting interface or abstract class, so it will be used with any class that implements that interface or extends the abstract class.

Factory Method Pattern in my project:

Main Class is the client. Only through the factory, the client can create the objects.

Creater Class is the Factory Class, and it is also abstract class. It declares an interface for operations that create abstract product objects

MenuCreater and OrderCreater are the ConcreateFactory Class, they extend the Creater Class.

ConcreteFactory implements the operations to create concrete product objects

Menu Class is the Product class, it is abstract.

Product class declares an interface for a type of product object

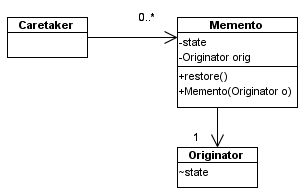
ChineseMenu and WesternMenu are the ProductA/B Class, they extend the Product Class.

ConcreteProduct defines a product object to be created by the corresponding concrete factory and implements the AbstractProduct interface.

In case of create Chinese Order:

The Command Interface ask the Creater interface and the Order(as Concrete Factory) will return ChineseOrder which is extend of Order (as Product)

## Memento Pattern:

****

Advantage of Memento Pattern:

Provides a way of recording the internal state of an object in a separate object without violating law of design pattern,

Simplifies the Originator by giving responsibility of Memento storage among the Caretakers.

Memento Pattern in my project:

The OrderList is the Caretaker. The Caretaker manages the timing of the saving of the state, saves the Memento and, if needed, uses Memento to restore the state of the Originator.

The ArrayList <Order>Class is the Memento. The Memento is another object that saves the state of the Originator. I decided that all Order Class store in Arraylist of OrderList class.

The Order Class is the Originator. The Originator is the object whose state we want to save.

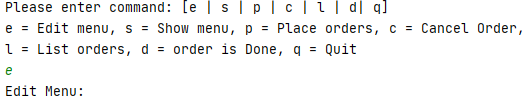
The Order class store all details of order.

In case of Cancel Order:

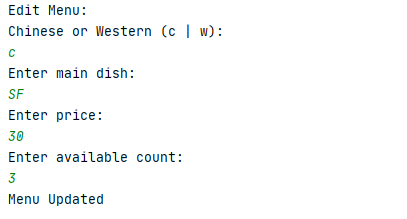
The CancelClass execute method will call OrderList Class(as Caretaker) restore method, it will ask the user input staff number, the method remove the Order by staff Id. All Orders store in ArrayList<Order>(as Memento, Order class (as Originator)store all details of order) in OrderList Class

# **User Guide**

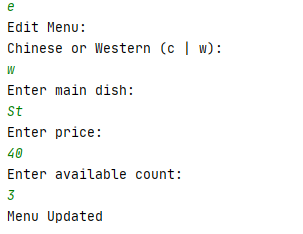
When the program start, please input “e” first, to set up the menu first.

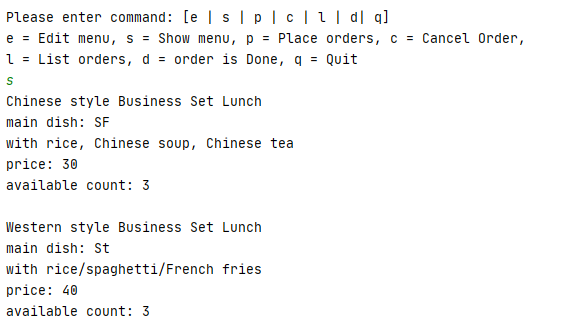


Then, input type of menu, main dish name, price and count.

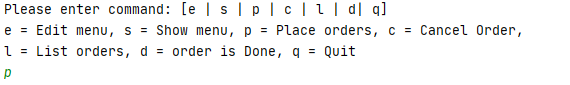


Don’t forget set up other type of menu





For the place order function, input “p”.



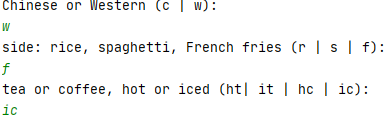
Then, choose Chinese or western



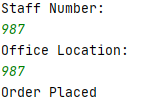
If you choose Chinese, select iced tea or hot tea



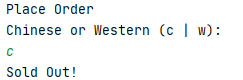
If you choose Western, choose side and drink



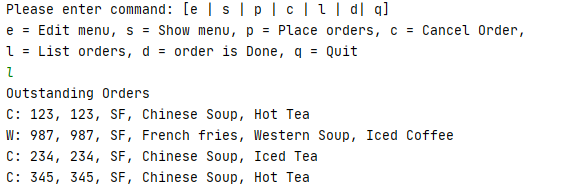
Then, input staff number and office location



If the set lunch is sold out, you can not place it.



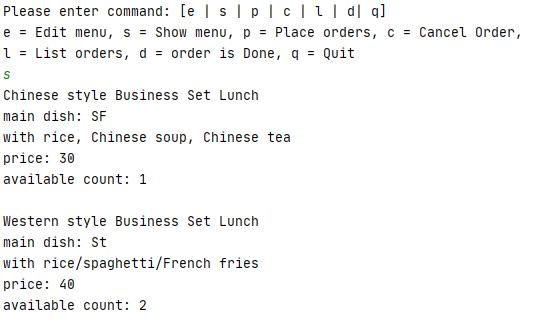
Enter command “l” to list orders



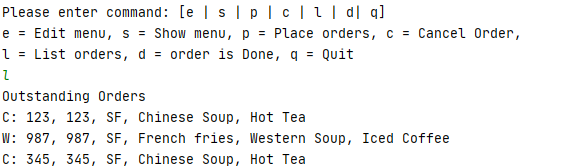
If you want to cancel order, enter command “c”, then enter the staff number:



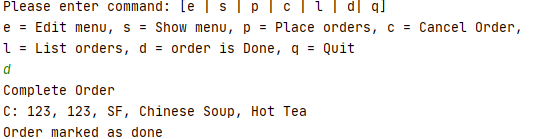
The staff can place order again



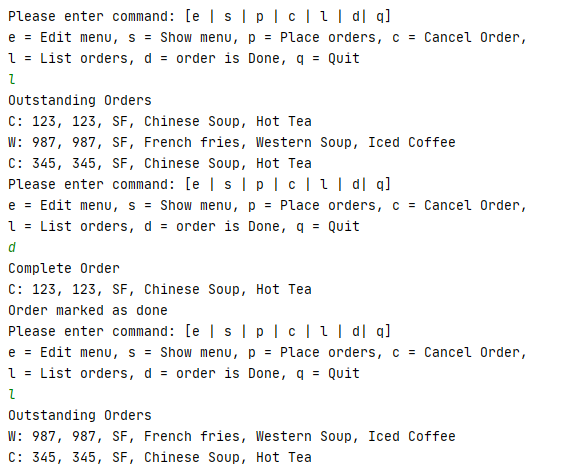
You can see the order has been removed by using list orders



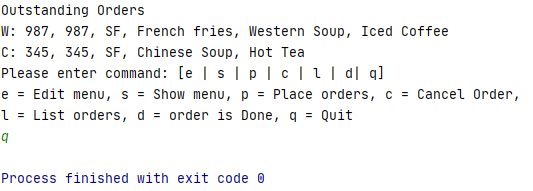
For order is Done function, enter ”d”



The first order will be removed.



Finally, enter “q” to exit program.



# **Test Plan and Test Cases**

Test Plan:

All function should be tested.

1. Edit menu
2. Show menu (display lunch sets with available counts)
3. Make order
4. Cancel order
5. List outstanding orders (orders which have not been completed)
6. Mark order as completed

Test Case

|  |  |
| --- | --- |
| **Test Case ID** | Test-1 |
| **Test Case Description** | Edit Menu and show Menu |
| **Steps** | 1. When the program start, please input “e” first, to set up the menu first. Input the information needed and submit 2. Then, input type of menu, main dish name, price and count. 3. Don’t forget set up other type of menu 4. Finally, Input ‘s’ command can show the details of menu |
| **Priority** | High |
| **Pre-Conditions** | / |
| **Expected Result** | Show menu function show test data successfully |
| **Final Result and**  **Test Data** | Success |
| **Expected Testing Date** | 12/11/2020 |
| **Actual Date** | 12/11/2020 |
| **Owner** | Cheung King Hung |
| **Test Case ID** | Test-2 |
| **Test Case Description** | Place Order and List Order Function |
| **Steps** | 1. For the place order function, input “p”.Then, input type of menu, main dish name, price and count. 2. Then, choose Chinese or western 3. For choose Chinese, select iced tea or hot tea 4. For choose Western, choose side and drink 5. Then, input staff number and office location 6. Repeat step 1-5 Four times 7. If the set lunch is sold out, you can not place it. 8. Enter command “l” to list orders |
| **Priority** | High |
| **Pre-Conditions** | Test Case ID :Test-1 |
| **Expected Result** | List orders function show test data successfully |
| **Final Result and**  **Test data** | Success |
| **Expected Testing Date** | 12/11/2020 |
| **Actual Date** | 12/11/2020 |
| **Owner** | Cheung King Hung |

|  |  |
| --- | --- |
| **Test Case ID** | Test-3 |
| **Test Case Description** | Cancel Order |
| **Steps** | 1. enter command “c”, then enter the staff number 2. Enter ‘s’ command to find out the count of munch set. 3. Enter ‘l’ command to find out the list orders |
| **Priority** | High |
| **Pre-Conditions** | Test Case ID :Test-1, Test-2 |
| **Expected Result** | Chinese lunch set count will be increase  Only 3 Outstanding Orders |
| **Final Result and**  **Test Data** | Success |
| **Expected Testing Date** | 12/11/2020 |
| **Actual Date** | 12/11/2020 |
| **Owner** | Cheung King Hung |

|  |  |
| --- | --- |
| **Test Case ID** | Test-4 |
| **Test Case Description** | Order is Done function |
| **Steps** | 1. enter command “d” 2. The first order will be removed. 3. Enter ‘l’ command to find out the list orders |
| **Priority** | High |
| **Pre-Conditions** | Test Case ID :Test-1, Test-2, Test-3 |
| **Expected Result** | Only 2 Outstanding Orders |
| **Final Result and**  **Test Data** | Success |
| **Expected Testing Date** | 12/11/2020 |
| **Actual Date** | 12/11/2020 |
| **Owner** | Cheung King Hung |

|  |  |
| --- | --- |
| **Test Case ID** | Test-0 |
| **Test Case Description** | Quit program function |
| **Steps** | 1. enter command “q” |
| **Priority** | Low |
| **Pre-Conditions** | / |
| **Expected Result** | The program will be terminated |
| **Final Result and**  **Test Data** | Success |
| **Expected Testing Date** | 12/11/2020 |
| **Actual Date** | 12/11/2020 |
| **Owner** | Cheung King Hung |

# **6. Well documented Source Code**

