1. **Assumptions regarding the problem context**

As a system analyst of the Company, I are required to design and develop TIMS. ToyProduct, Robot and RemoteControlCar must be kept unchanged. And the System has required to provide the following functions.

1. Create a ToyProduct record with zero quantity (Robot or RemoteControlCar or any new kind of ToyProduct in the coming future).

2. Display ToyProduct details (such as productID, name, qty, height/maximum speed) by a given productID (input code=\* to display all records)

3. Purchase ToyProduct and update the qty and the cost of the inventory records

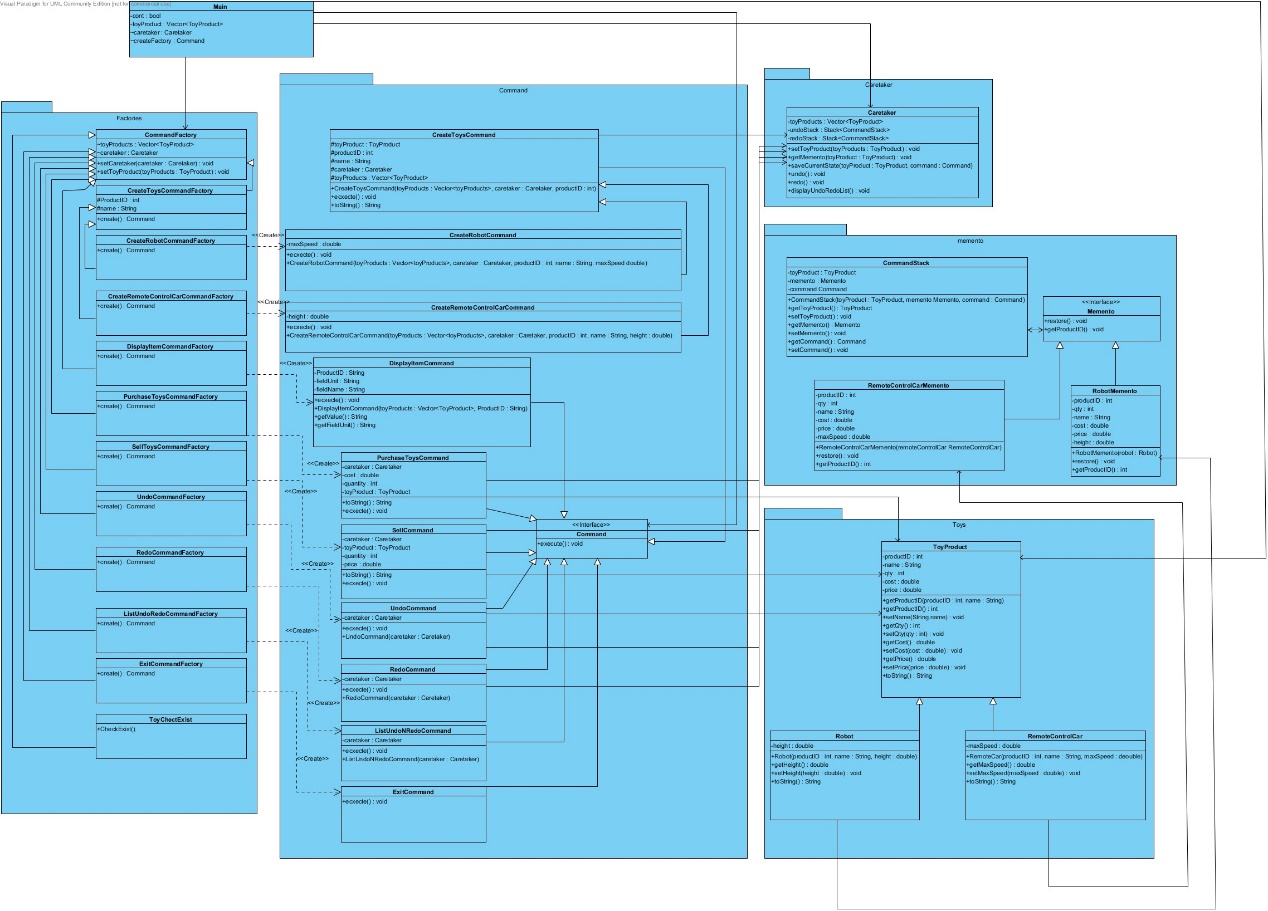
4. Sell ToyProduct and update the qty and the price of the inventory records

5. Undo last command

6. Redo the last undone command

7. Show undo/redo list

1. **Application design with class diagram**



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

In this Program, it has adopted 3 design patterns, which were command pattern, factory Pattern and memento.

Firstly, we have adopted abstract factory Pattern to create the command, the reason that it is necessary move because no one can tell will there be a new command come up, therefore, factory Pattern can close the existing class modification by separated the Command Factory and Command. In the program, all the Command Factory extends the CommandFactory Class, and CreateRobotCommandFactory and CreateRemoteControlCar class is extends the CreateToysCommandFactory, when it has new Command or product further, the programmer just needs to create a new class and control flow, it is fixable and increase the readability, after user input the required information, the system will create the command class for further action.

Next, it has applied command pattern in the program, it has declares an interface class (Command) to provide providing a simple execute() method which asks the Receiver of the command to carry out an operation, each Command class must implements the Command interface, and each Command class will override the excute() method, and do the actual action in the excute() method.

Finally, the memento design pattern is used due to we need to save the state of the object so that can restore later on. In the program, when user create, purchase, or sell product command, the execute() method will run the caretaker.saveCurrentState(), when saveCurrentState() received the object, it will run the getMemento() to check which Product it is, then it will create the corresponding Memento Class, Next, the command and product will push to undoStack to store it, it will capture the internal state of an object without violating encapsulation, Therefore, when user input the undo command, it will run the caretaker.undo() method, and it will pop the object in the undoStack and call the restore method in that memento, In addition, the interface of Memento has been created, in order to avoid violating "open/closed" principle when it has new toy product.

# User Guide

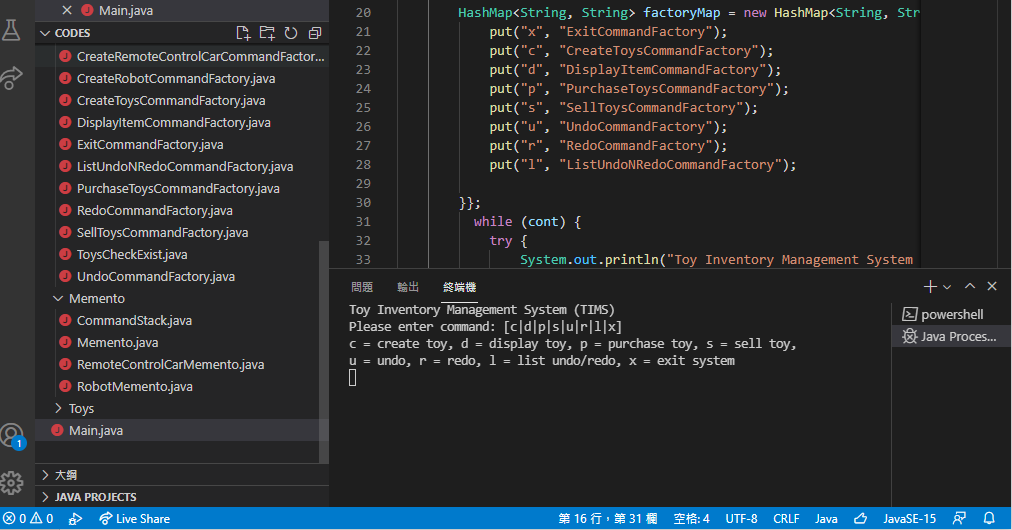
When you enter the system, it will ask user to type the following character to do the action that what user want (The commands below are case insensitive.):

|  |  |
| --- | --- |
| **Input** | **fun** |
| **c** | 1. Create a ToyProduct record with zero quantity (Robot or RemoteControlCar or any new kind of ToyProduct in the coming future). |
| **d** | Display ToyProduct details (such as productID, name, qty, height/maximum speed) by a given productID (input code=\* to display all records) |
| **p** | Purchase ToyProduct and update the qty and the cost of the inventory records |
| **s** | Sell ToyProduct and update the qty and the price of the inventory records |
| **u** | Undo last command |
| **r** | Redo the last undone command |
| **l** | Show undo/redo list |
| **x** | Exit the System |

1. **Test Plan and Test Cases**

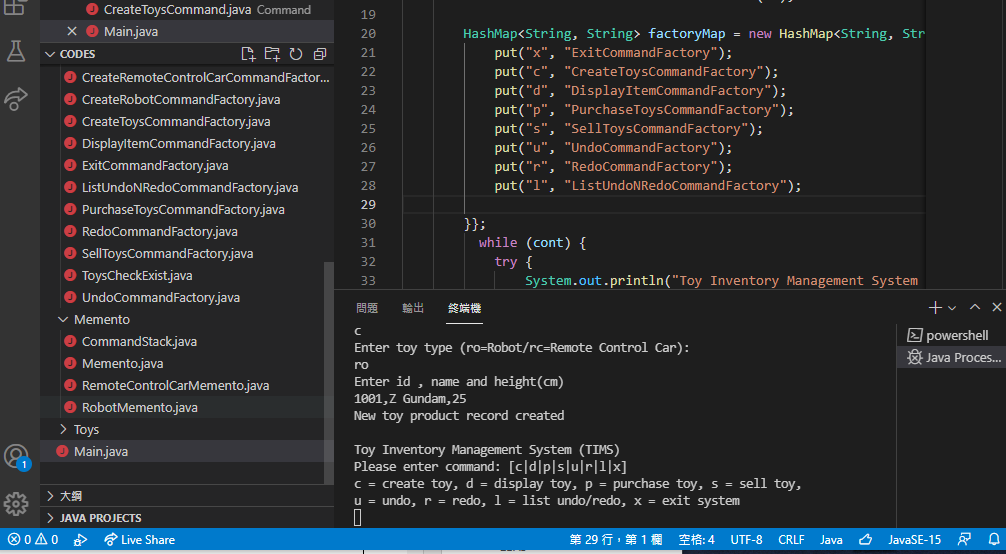
**Test Plan：**

**In the following test Cases**, **it will divide to 2-part, first part will be following the “*Sample User Inputs for a Test Run*” to run the program. After that, it will input incorrect to the program to test what will happen when error occurred. All the Test is test on Visual Studio Code.**

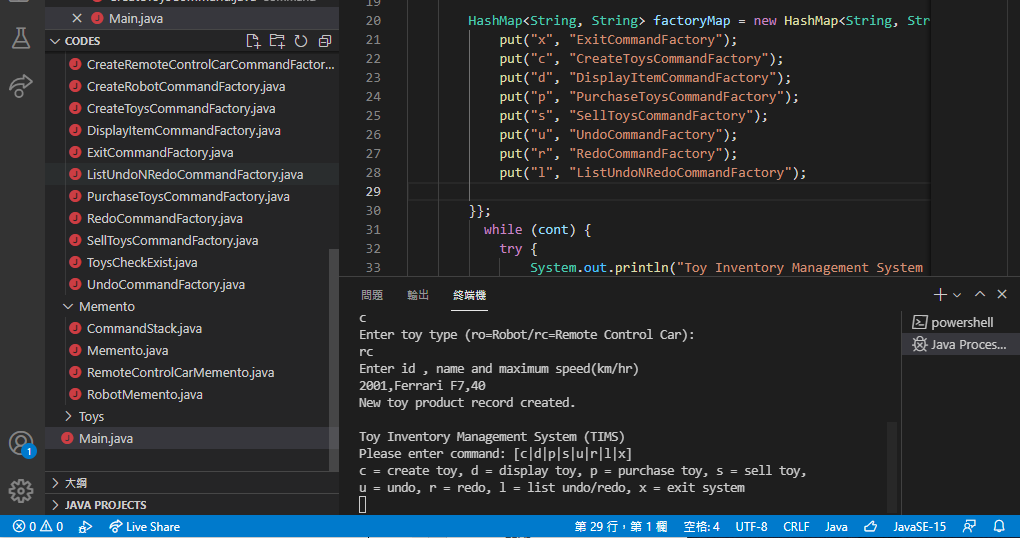


**Test Cases（The Input is following the *Sample User Inputs for a Test Run*）：**

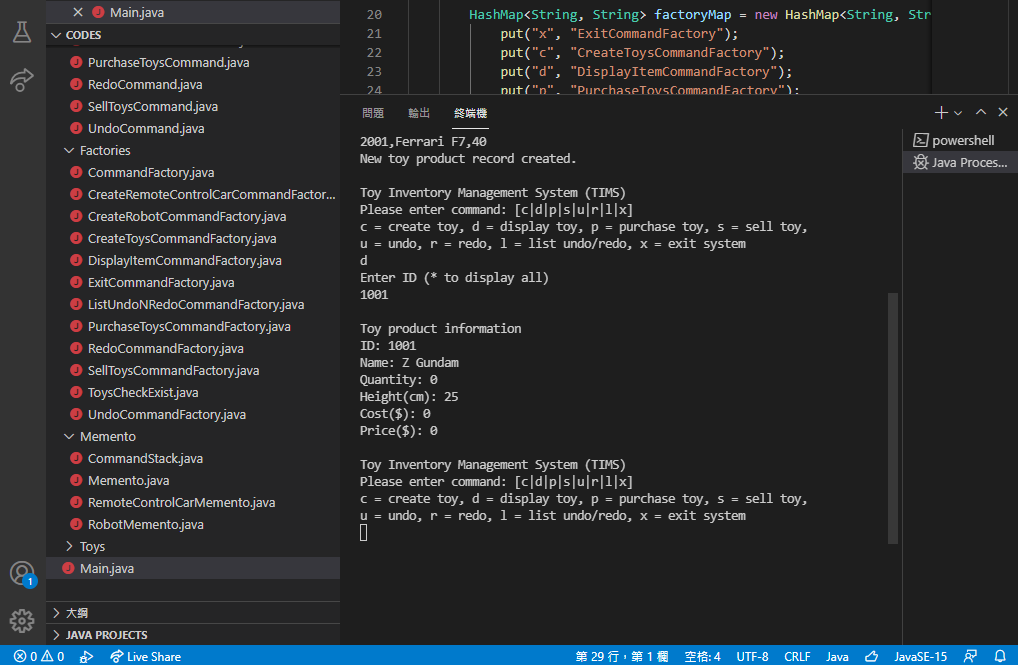
|  |  |
| --- | --- |
| **Test Case ID** | **TC-001** |
| **Test Case Description** | **Test create toy** |
| **Steps** | 1. **Input c** 2. **Input “ro”** 3. **Input 1001, Z Gundam,25** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The Toy Product Record Created** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |

****

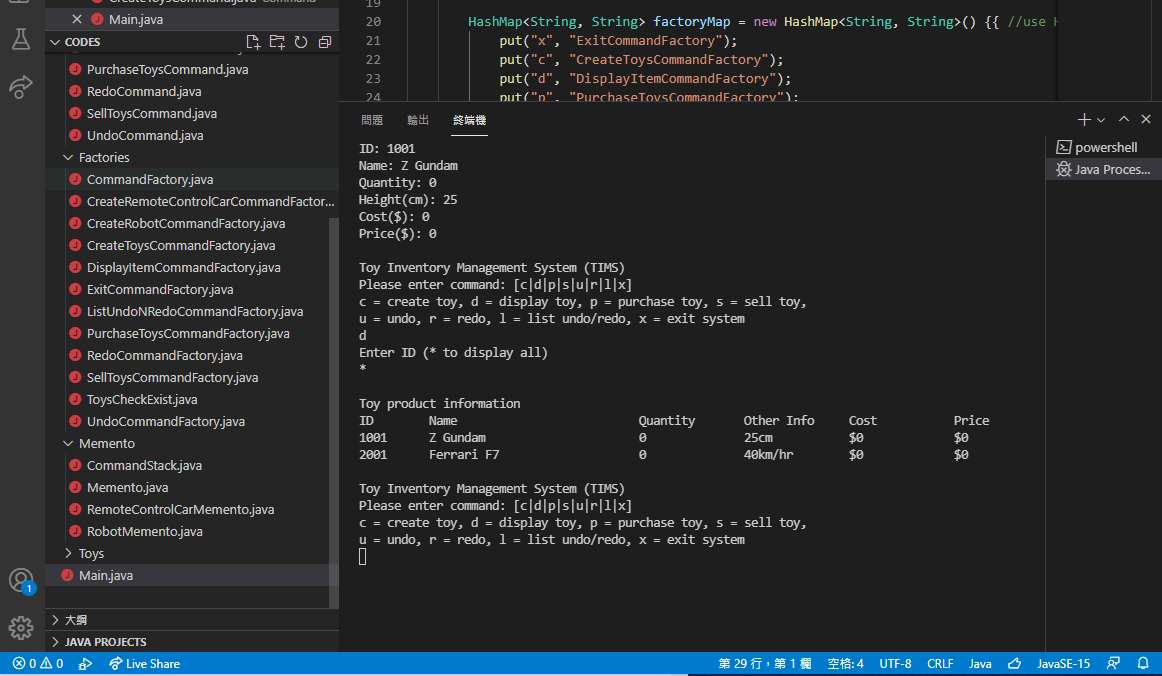
|  |  |
| --- | --- |
| **Test Case ID** | **TC-002** |
| **Test Case Description** | **Test create toy** |
| **Steps** | 1. **input c** 2. **input “rc”** 3. **input 2001, Ferrari F7,40** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The Toy Product Record Created** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |

****

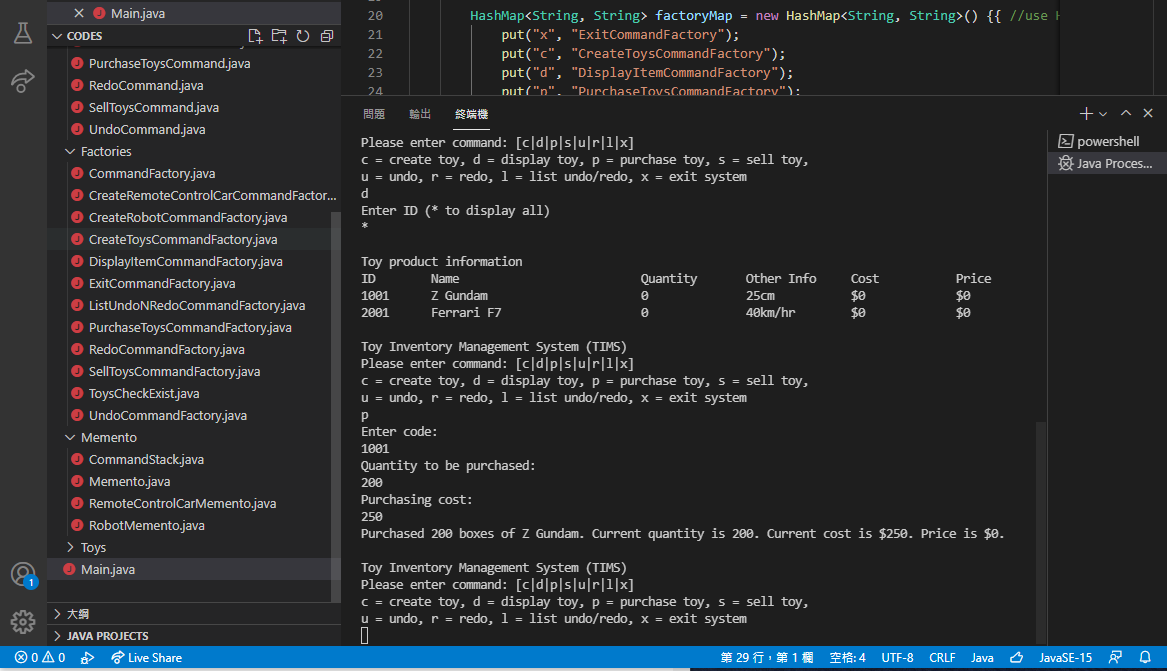
|  |  |
| --- | --- |
| **Test Case ID** | **TC-003** |
| **Test Case Description** | **Display the toy individually** |
| **Steps** | 1. **input d** 2. **input 1001** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **Show 1001 Product detail** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



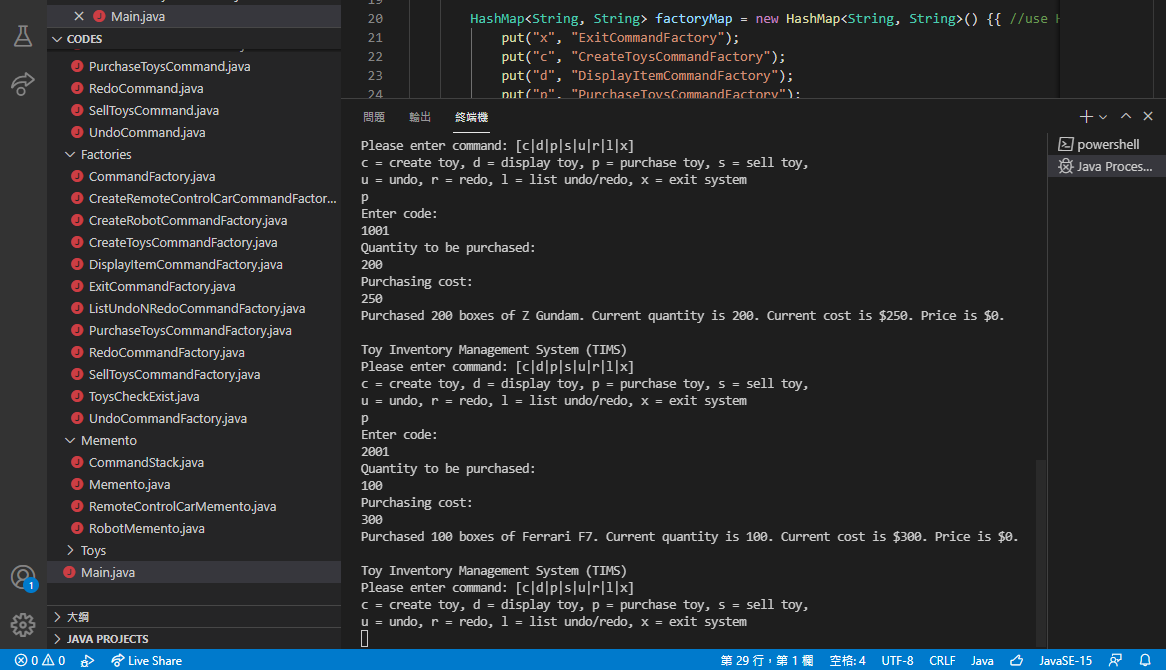
|  |  |
| --- | --- |
| **Test Case ID** | **TC-004** |
| **Test Case Description** | **Display the all the product** |
| **Steps** | 1. **input d** 2. **input \*** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **Show all the detail** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



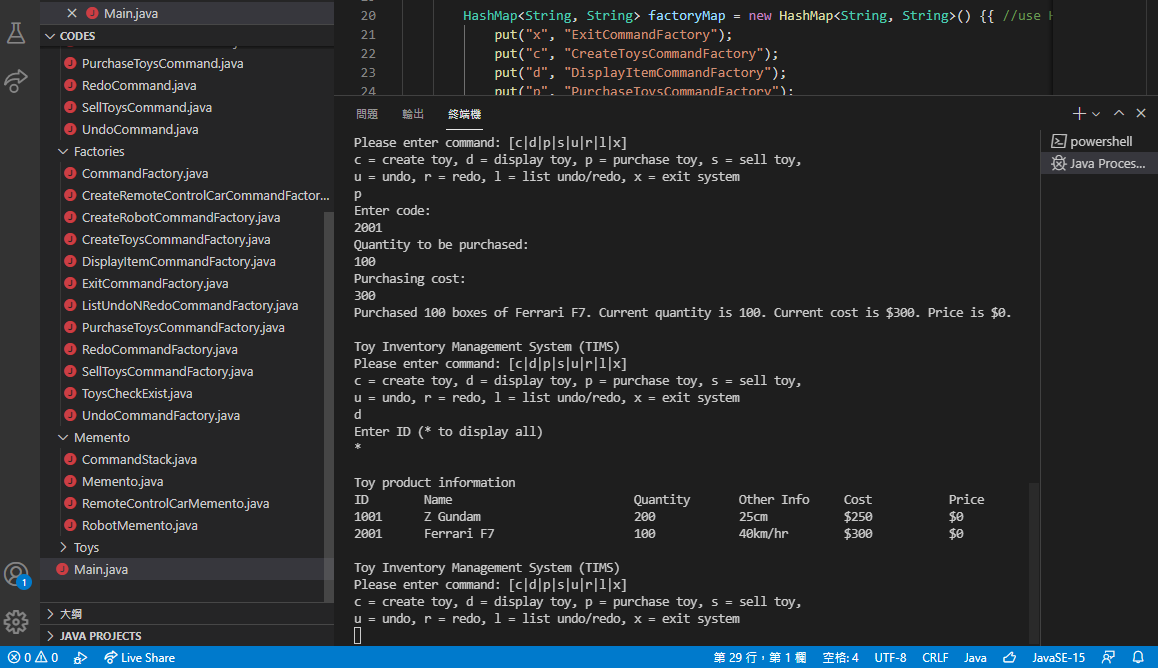
|  |  |
| --- | --- |
| **Test Case ID** | **TC-005** |
| **Test Case Description** | **Test purchase toys** |
| **Steps** | 1. **input p** 2. **input 1001** 3. **input 200** 4. **input 250** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The cost should updated.** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



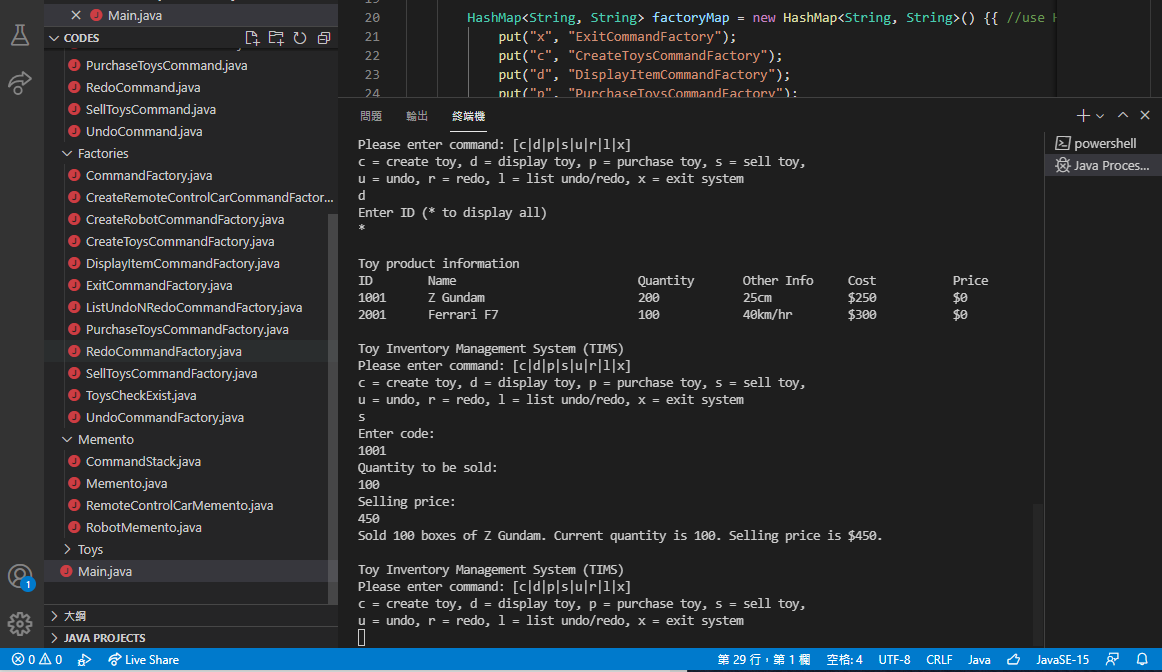
|  |  |
| --- | --- |
| **Test Case ID** | **TC-006** |
| **Test Case Description** | **Test purchase toys again** |
| **Steps** | 1. **input p** 2. **input 2001** 3. **input 100** 4. **input 300** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The cost should updated.** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



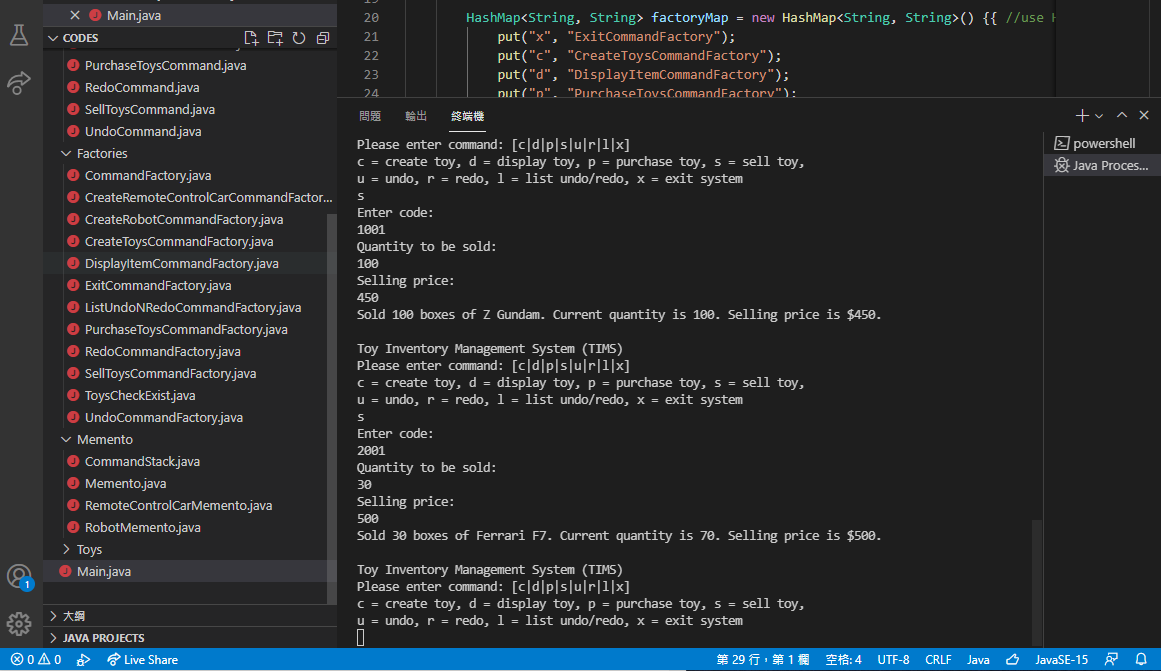
|  |  |
| --- | --- |
| **Test Case ID** | **TC-007** |
| **Test Case Description** | **Check is it updated the cost** |
| **Steps** | 1. **input d** 2. **input \*** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **it is updated** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



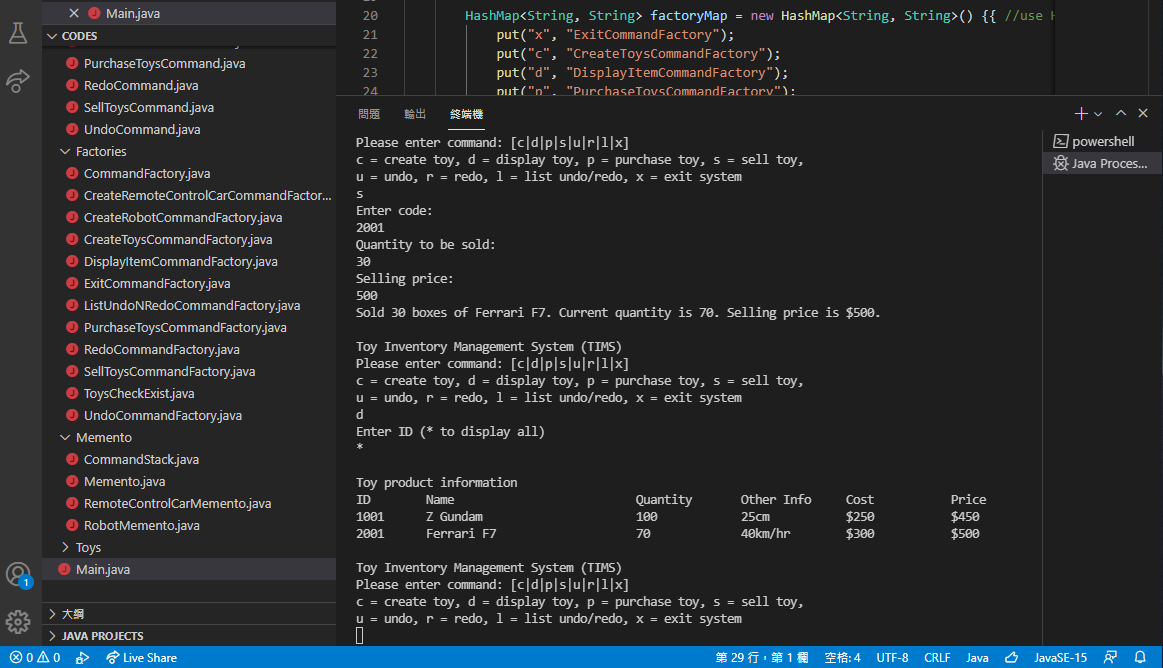
|  |  |
| --- | --- |
| **Test Case ID** | **TC-008** |
| **Test Case Description** | **Test sell product** |
| **Steps** | 1. **input 1001** 2. **input 100** 3. **input450** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The Price should be updated** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



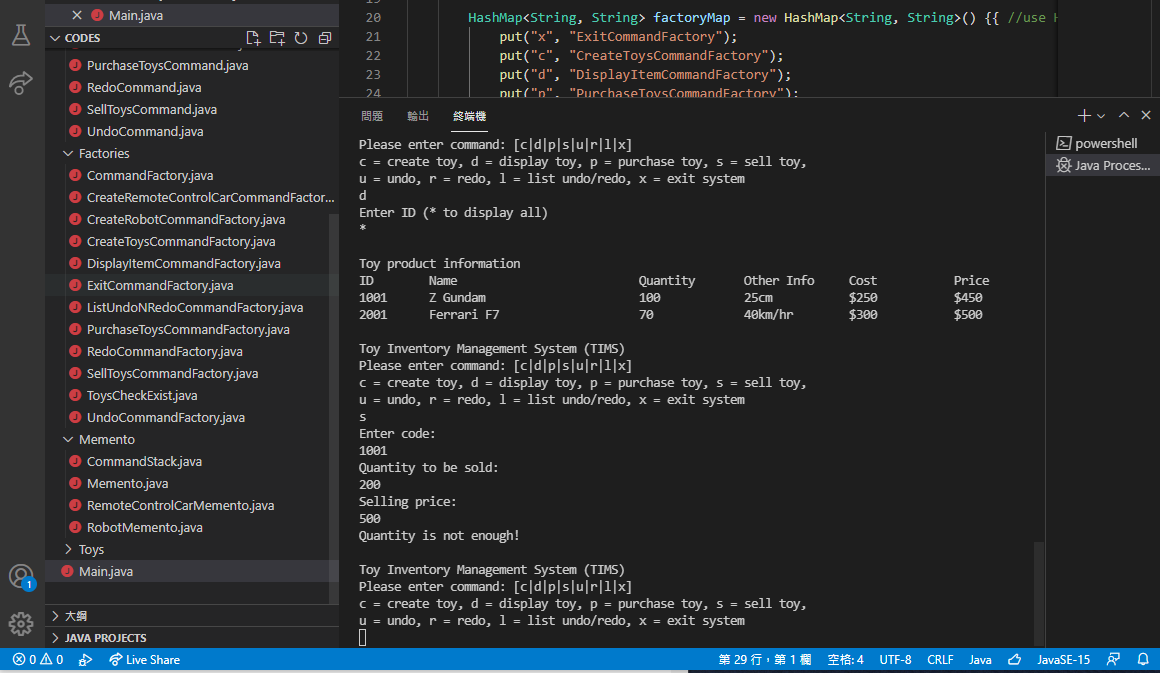
|  |  |
| --- | --- |
| **Test Case ID** | **TC-009** |
| **Test Case Description** | **Test sell product again** |
| **Steps** | 1. **input 2001** 2. **input 30** 3. **input 500** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The Price should be updated** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



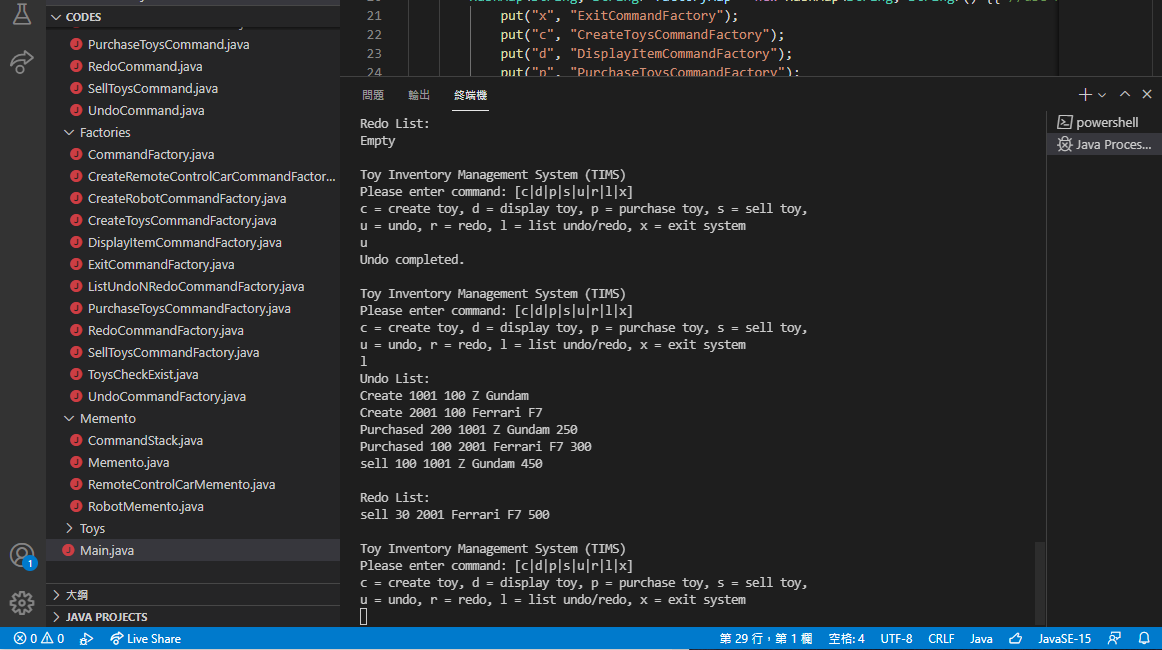
|  |  |
| --- | --- |
| **Test Case ID** | **TC-010** |
| **Test Case Description** | **Check is it updated** |
| **Steps** | 1. **input d** 2. **input \*** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The price undated** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



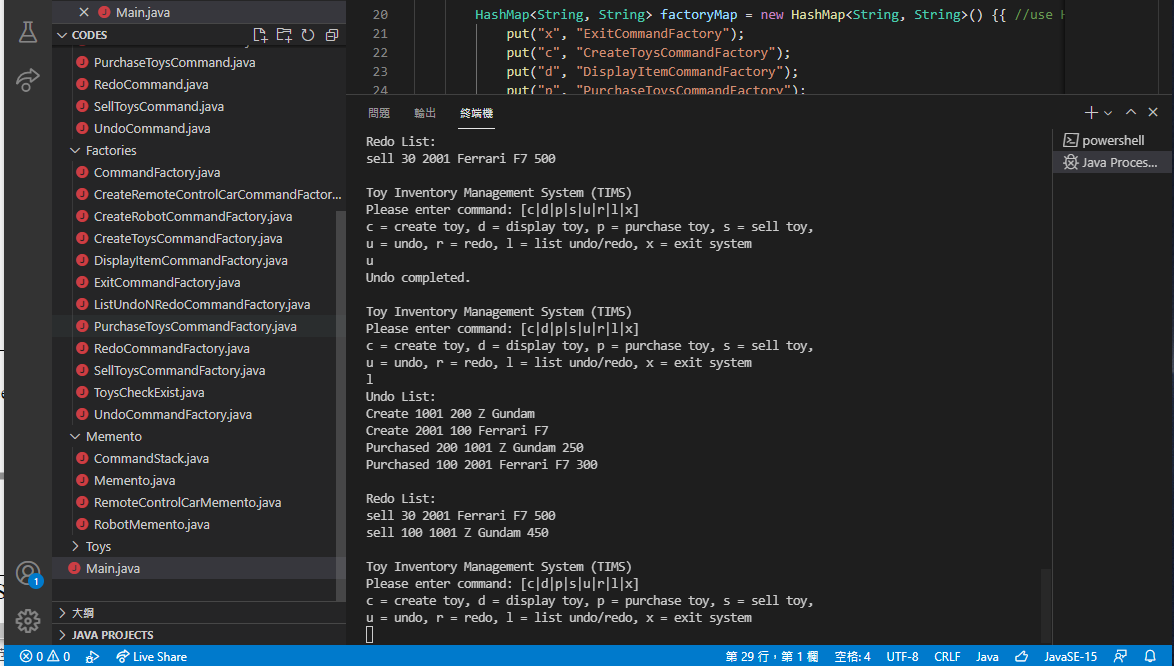
|  |  |
| --- | --- |
| **Test Case ID** | **TC-011** |
| **Test Case Description** | **Test if Quantity is not enough** |
| **Steps** | 1. **input 1001** 2. **input 200** 3. **input 500** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **It will come up the error message** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



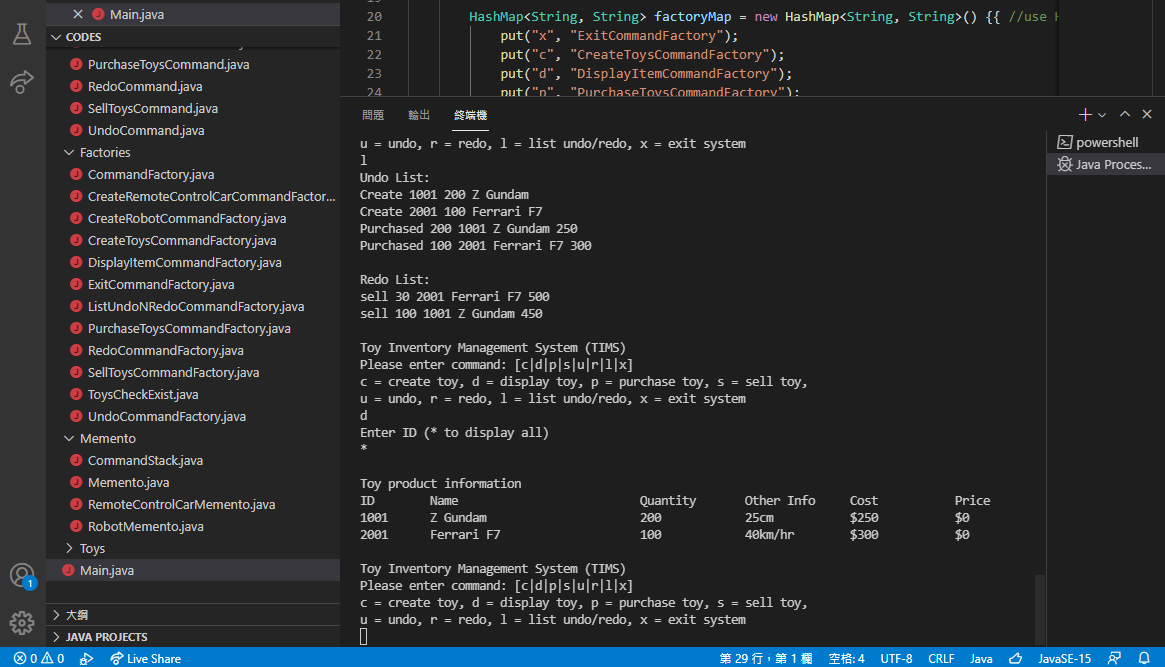
|  |  |
| --- | --- |
| **Test Case ID** | **TC-012** |
| **Test Case Description** | **Test undo function** |
| **Steps** | 1. **input L** 2. **input U** 3. **input L** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **It will show the list** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



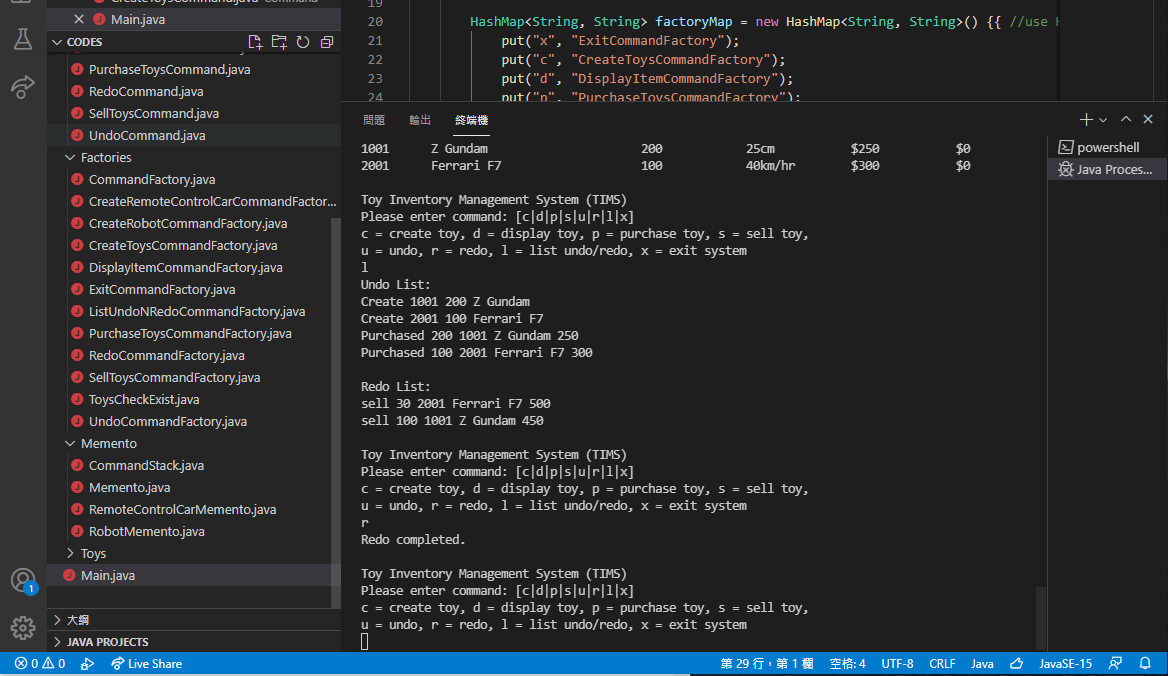
|  |  |
| --- | --- |
| **Test Case ID** | **TC-013** |
| **Test Case Description** | **Test undo again** |
| **Steps** | 1. **input u** 2. **input l** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **Can see it has undo** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



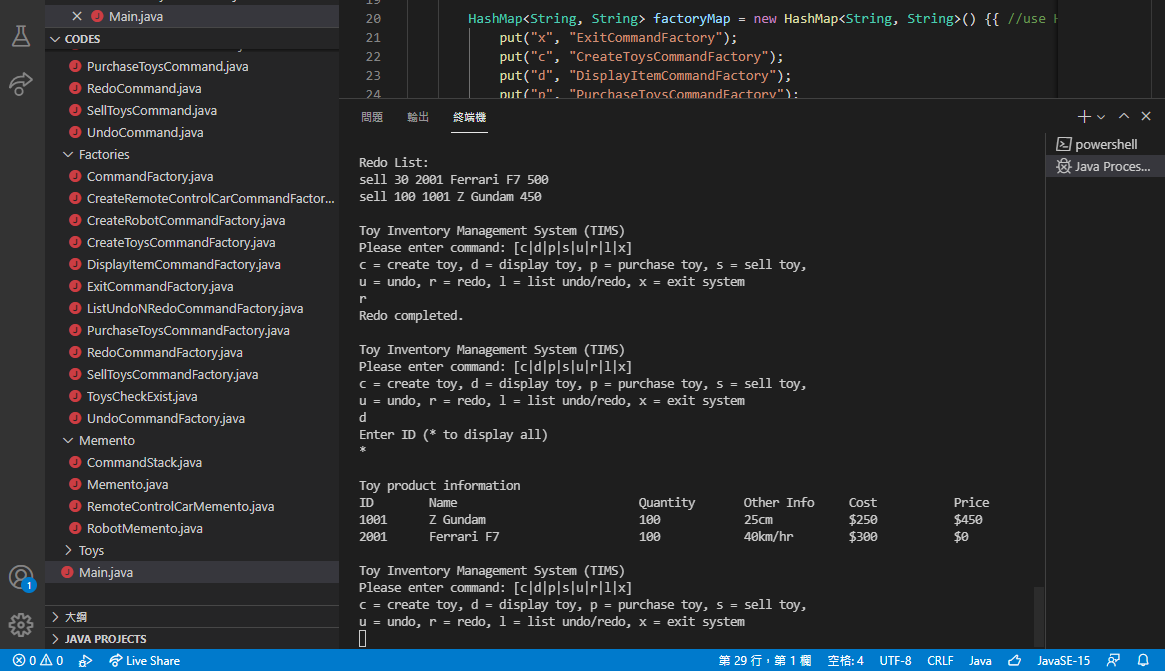
|  |  |
| --- | --- |
| **Test Case ID** | **TC-014** |
| **Test Case Description** | **Check has the Undo processed** |
| **Steps** | 1. **input d** 2. **input \*** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The Price should undo** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



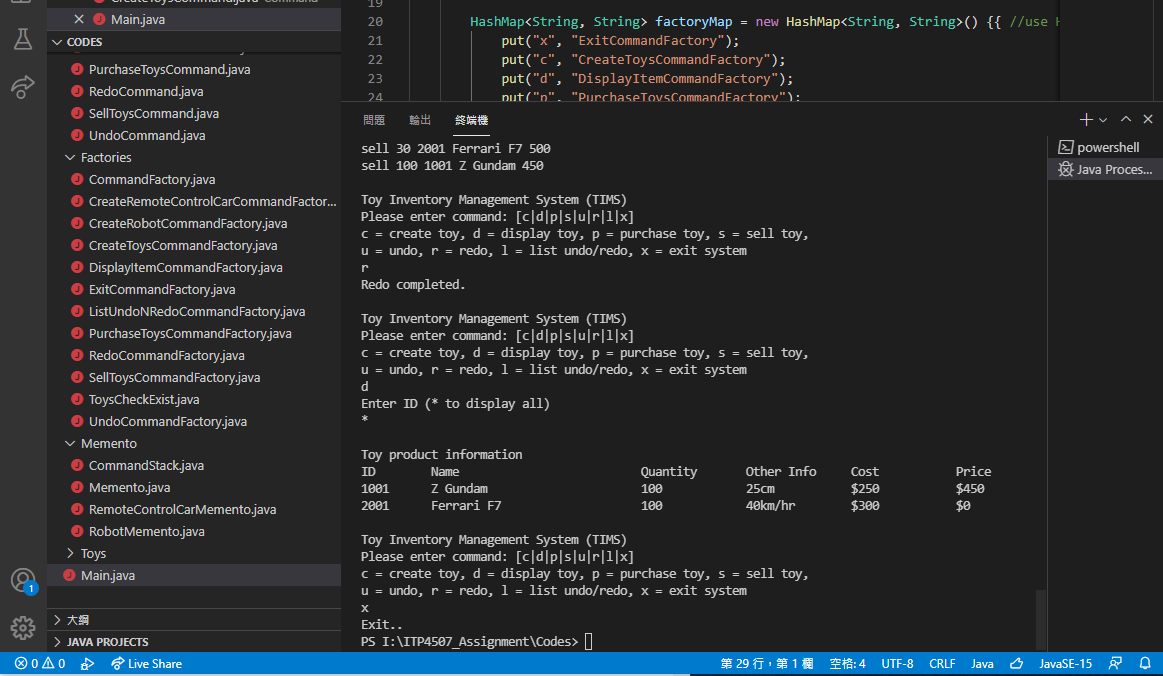
|  |  |
| --- | --- |
| **Test Case ID** | **TC-015** |
| **Test Case Description** | **Test redo function** |
| **Steps** | 1. **input l** 2. **input r** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **It should redo the command** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



|  |  |
| --- | --- |
| **Test Case ID** | **TC-016** |
| **Test Case Description** | **Check is it redo** |
| **Steps** | 1. **input d** 2. **input \*** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **The Price has updated** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |

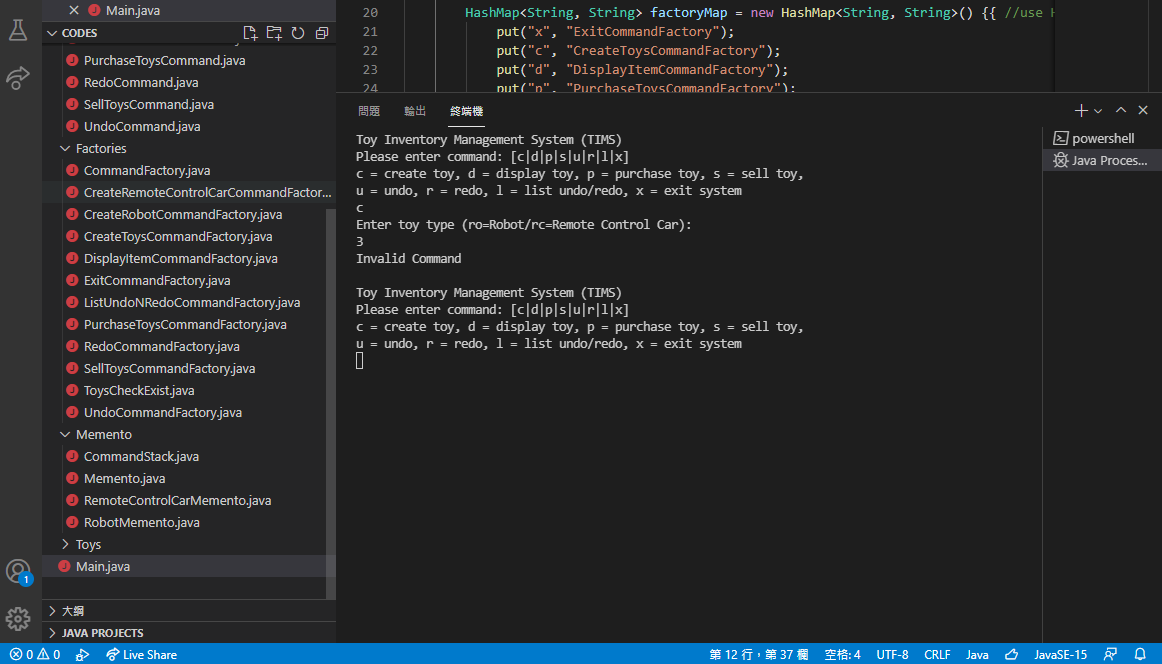


|  |  |
| --- | --- |
| **Test Case ID** | **TC-017** |
| **Test Case Description** | **Exit the System** |
| **Steps** | 1. **input x** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | 1. **Exit the System** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |

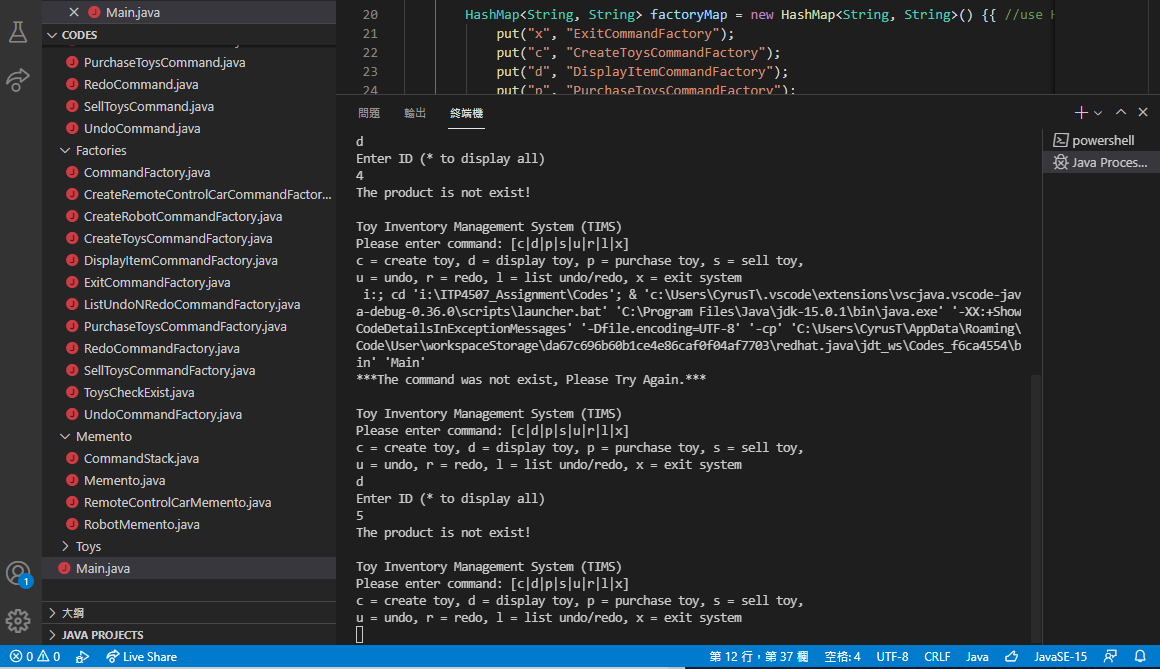


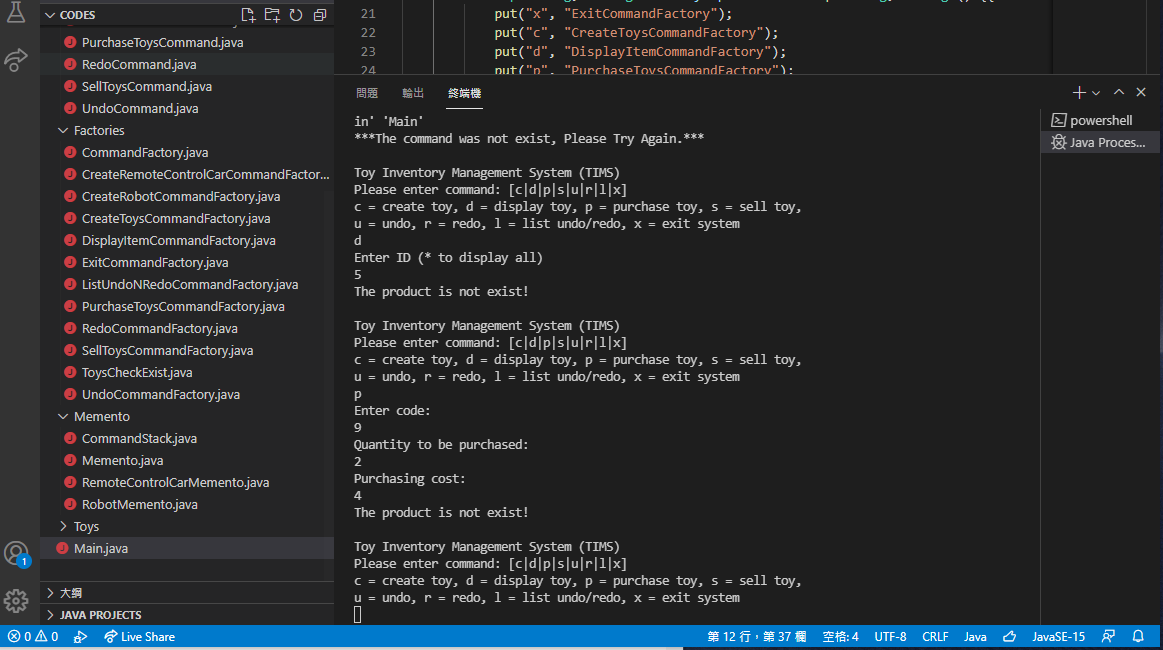
**Test Cases（Check the wrong input）：**

|  |  |
| --- | --- |
| **Test Case ID** | **TE-001** |
| **Test Case Description** | **Test Wrong input have process** |
| **Steps** | 1. **input x** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **Error Message pop out** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |

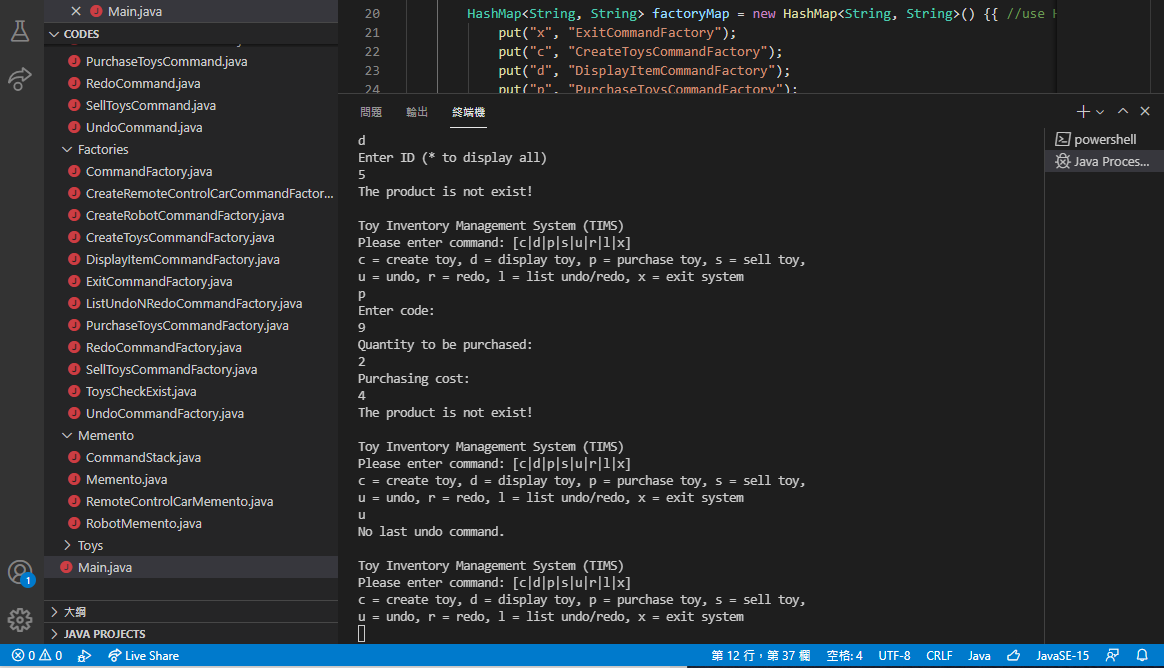


|  |  |
| --- | --- |
| **Test Case ID** | **TE-001** |
| **Test Case Description** | **Input Product ID that is not exist** |
| **Steps** | 1. **input x** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **Error Message pop out** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |

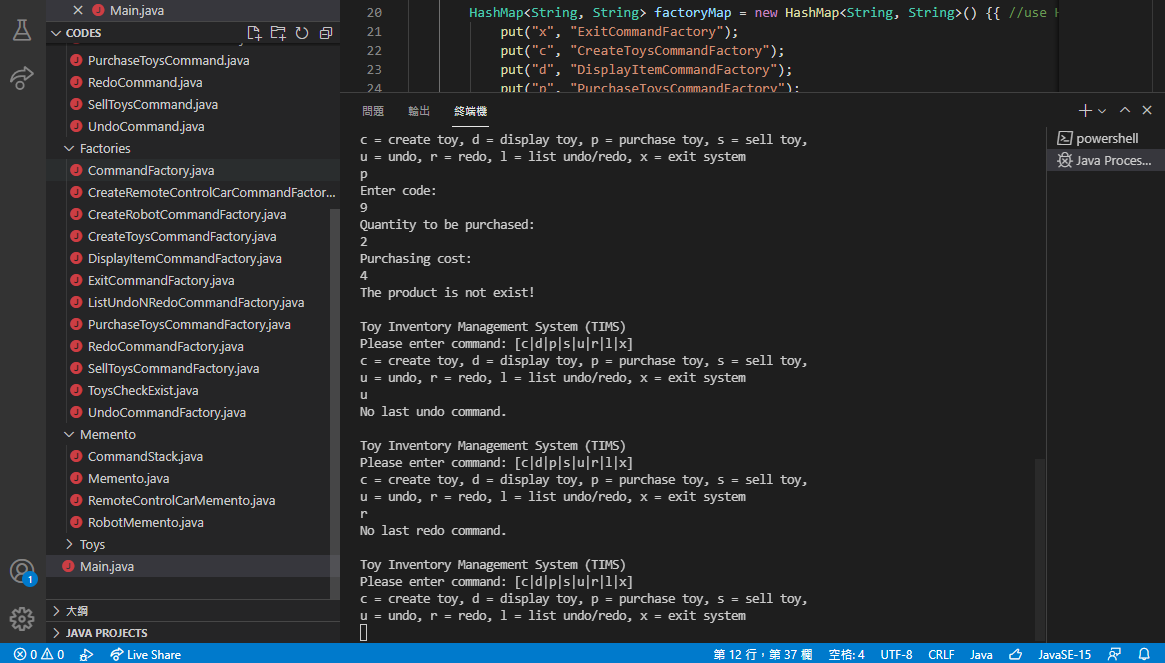




|  |  |
| --- | --- |
| **Test Case ID** | **TE-001** |
| **Test Case Description** | **Test undo when no command has processed** |
| **Steps** | 1. **input u** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **Error Message pop out** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



|  |  |
| --- | --- |
| **Test Case ID** | **TE-001** |
| **Test Case Description** | **Test redo when no command has processed** |
| **Steps** | 1. **input r** |
| **Test Date** | **12/11/2021** |
| **Expected Result** | **Error Message pop out** |
| **Final Result** | **Success, The Program returns the same result** |
| **Tester** | **Tsang Tsz Fung** |
|  |  |



1. **Well documented Source Code**

The Source Code has zipped in the file.