Financial Computing H/W #1

2012, Fall

(Prof. Eran Fishler)

(Grader : Ankit Mittal / Shinan Zhang)

Name : **Jinil Jang**

Univ ID : **N10917970**

1 . Develop Environment

- Program Language : Java (JDK 1.7.0)
- Program IDE : IntelliJ IDEA 11.0.2

2. Purpose of each Class

- Trade class : An object of this class is represent one trade data.
- TradeIterator class : It can access actual trade database and works as iterator of trade data.
- TradeChecker class : This class helps checking whether a trade data is validate or not with various ways.
- TradeProcessor class : Main class. This program starts in here.

3. How to run the program

I. Java TradeProcessor <file location>
   (Example) java TradeProcessor ./data/tradeData.txt

II. If you use intelliJ IDEA, you can run this program like this :

① Press ‘F9’

② In the ‘Debug’ window, click upper one : ‘0. Edit configurations…’

③ In the ‘Configuration’ tab, you can find ‘Program arguments : “ text box.
Write the file location in here (example : ./data/tradeData.txt)

4. Basic method or implement of H/W#1 description.

Requirement		Location
1. Trade class	Each fields Getter methods Setter methods	Trade.java
2. Iterator class	hasNext() next() use BufferedReader class Read one line at a time Use split method	TradeIterator.java

5. Special things in my program.

➢ **Validation check of trade data**

① Blank Line check in trade database

- If there are blank line in the trade database, display this message.

<< ERROR : The trade inform is blank. >>

After that, the process is keep going.

② Check the number of the trade fields

- If the number is less than 4 (trade data necessarily requires 4 fields), display this error message.

<< ERROR : The trade fields are not enough. >>

If the number is more than 4, display this error message.

<< ERROR : The trade fields are too much. >>

After that, the process is keep going.

③ Check one of fields is not empty

- If one of fields is empty, display error message. In this case, it displays all fields name that data is empty.

<< ERROR : time symbol quantity is(are) empty. >>

After that, the process is keep going.

④ Check each fields' type

- I defined the fields type like this :

  time (int) | symbol (String) | quantity (int) | price (double)

  If each fields' data is not corresponding with this, display error message.

  << ERROR : time (int) type is wrong. >>

  After that, the process is keep going.

★ In this validation check of trade data, I concentrate in this design.

1. The trade processor should keep running, even if this program find some trade data is wrong. There are much important data in the trade database. Therefore, if some trade data is wrong, skip this data and then keep processing other data. (Interruption of this program can cause much loss)
2. If a trade data is wrong, display error message and don't use this data. This is because this wrong data can cause confusion to users.

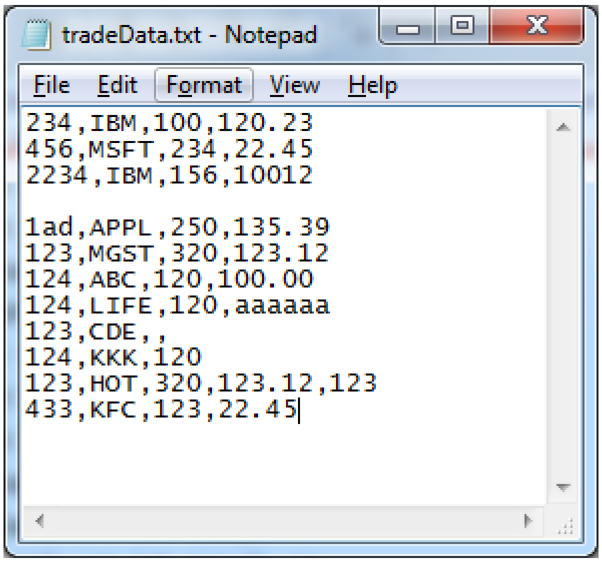
➢ **Façade Design Pattern**

- I use this design pattern in TradeChecker class.
  This class defined various methods to check the validation of trade data. But if program calls all of these methods in TradeIterator class to check trade data, it can be much complex, and it is difficult to control. That is, it is not good at OOP concept. Therefore, program just calls the 'tradeCheckFacade()' method in the TradeChecker class, and this method control all of checking method.

➢ **Object Oriented Programming concept**

- This program divides 4 classes (source code), and the standard of division is their purpose ; Trade data, Trade Iterator, Trade data check, and Trade Process. This is beneficial in the future, because we can think each class as object (It is much easier), and it is easy to modify or improve more methods. That is, this program is based on the Object Oriented Programming concept.

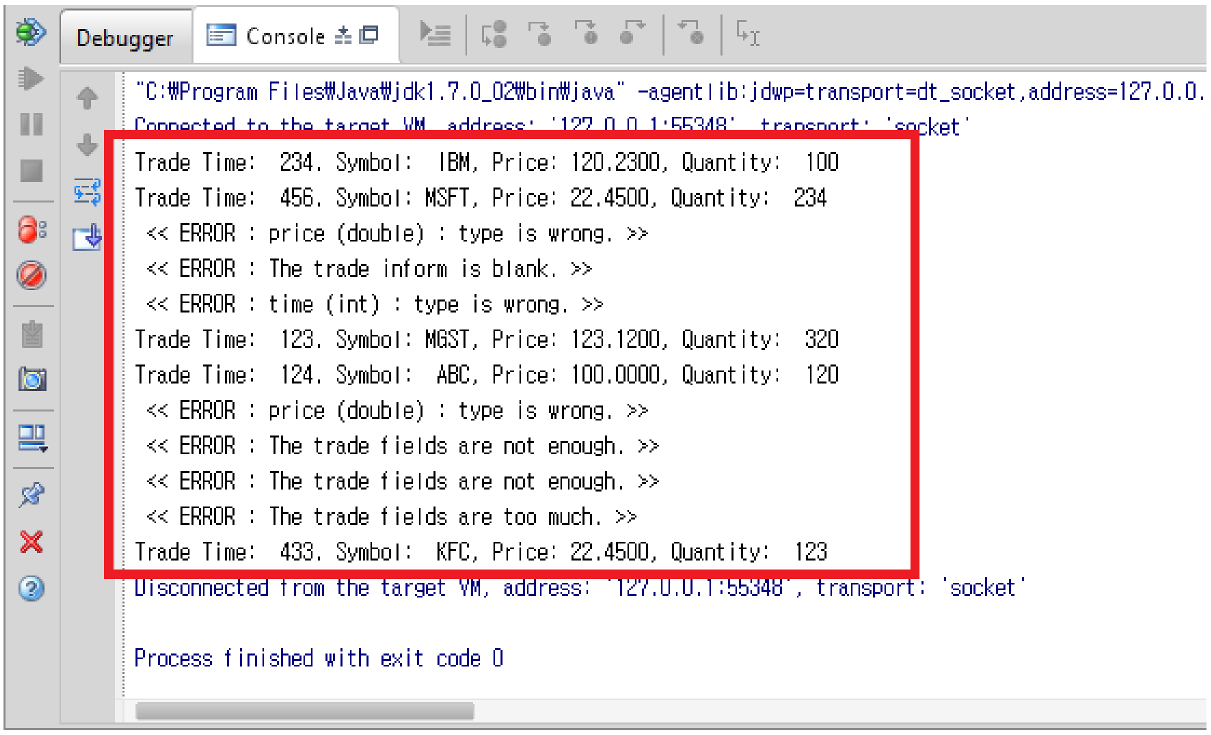
6. Insert data



```
234,IBM,100,120.23
456,MSFT,234,22.45
2234,IBM,156,10012

1ad,APPL,250,135.39
123,MGST,320,123.12
124,ABC,120,100.00
124,LIFE,120,aaaaaa
123,CDE,,
124,KKK,120
123,HOT,320,123.12,123
433,KFC,123,22.45
```

7. Result



```
"C:₩Program Files₩Java₩jdk1.7.0_02₩bin₩java" -agentlib:jdwp=transport=dt_socket,address=127.0.0.
Connected to the target VM, address: '127.0.0.1:55348', transport: 'socket'
Trade Time:  234, Symbol:  IBM, Price: 120.2300, Quantity:  100
Trade Time:  456, Symbol: MSFT, Price: 22.4500, Quantity:  234
 << ERROR : price (double) : type is wrong. >>
 << ERROR : The trade inform is blank. >>
 << ERROR : time (int) : type is wrong. >>
Trade Time:  123, Symbol: MGST, Price: 123.1200, Quantity:  320
Trade Time:  124, Symbol:  ABC, Price: 100.0000, Quantity:  120
 << ERROR : price (double) : type is wrong. >>
 << ERROR : The trade fields are not enough. >>
 << ERROR : The trade fields are not enough. >>
 << ERROR : The trade fields are too much. >>
Trade Time:  433, Symbol:  KFC, Price: 22.4500, Quantity:  123
Disconnected from the target VM, address: '127.0.0.1:55348', transport: 'socket'

Process finished with exit code 0
```