CSE 2010, HW4

Due Tue Mar 13 at the start of your lab section; Submit Server: class = cse2010, assignment = hw4S*x*Individual
Due Tue Mar 13 at the end of your lab section; Submit Server: class = cse2010, assignment = hw4S*x*GroupHelp
x is 14 or 23—your merged section number.

Bitcoin, or other kinds of cryptocurrency, has been frequently in the news. Besides the technology around Bitcoin, the price of a Bitcoin has been fluctuating in large amounts on different exchanges. How would you design an exchange system to match buyers and sellers of Bitcoins?

The goal of HW4 is to design an exchange system that can efficiently match buyers and sellers of "Fitcoins." The system allows users to enter, cancel, and change buy/sell orders. Each buy/sell order consists of a buyer/seller, a price, a quantity, and a timestamp. A buy order and a sell order are executed if they have the same price. During execution, if the buy (or sell) quantity is larger, the buy (or sell) order is updated with the remaining buy (or sell) quantity; however the timestamp remains the same. If two buy (or sell) orders have the same price, the earlier order has a higher priority. In the unlikely case that the buy order has a higher price than the sell order, the orders are executed at the average of the buy and sell prices. For simplicity, each user can have only one order and the timestamps are unique.

To manage and match buy and sell orders efficiently, use two priority queues: one for the sellers and one for the buyers. For priority queues, use HeapAdaptablePriorityQueue (textbook). You may modify HeapAdaptablePriorityQueue and related classes to increase their general functionality. The program files are on the course website.

Input: The command-line argument for HW4.java is the name of the input file, which has:

- EnterBuyOrder *time buyer price quantity*
- CancelBuyOrder *time buyer*
- ChangeBuyOrder *time buyer price quantity*
- EnterSellOrder *time seller price quantity*
- CancelSellOrder *time seller*
- ChangeSellOrder *time seller price quantity*
- DisplayHighestBuyOrder *time*
- DisplayLowestSellOrder *time*

Time is an integer in HHMMSS format, where HH is 00-23 and MM/SS is 00-59 (leading zeros are optional). Sample input files are on the course website.

Output: Output goes to the standard output (screen), each line corresponds to an action:

- EnterBuyOrder *time buyer price quantity* [ExistingBuyerError]
- CancelBuyOrder *time buyer* [noBuyerError]
- ChangeBuyOrder *time buyer price quantity* [noBuyerError]
- EnterSellOrder *time seller price quantity* [ExistingSellerError]
- CancelSellOrder *time seller* [noSellerError]
- ChangeSellOrder *time seller price quantity* [noSellerError]
- DisplayHighestBuyOrder *time buyer orderTime price quantity*
- DisplayLowestSellOrder *time seller orderTime price quantity*
- ExecuteBuySellOrders *price quality*
  Buyer: *buyer remainingBuyQuantity*
  Seller: *seller remainingSellQuantity*

Sample output is on the course website.

Submission: Submit HW4.java that has the main method, (modified) HeapAdaptablePriorityQueue.java and related classes, and other program files. Submissions for Individual and GroupHelp have the same guidelines as HW1.

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top of the assignment.