CS272 Lab Assignment #7.

Due: Thursday, 4/12 by 11:30pm.

In this assignment you must use linked list implementation for generic queue class presented in Chapter 7 of the textbook. The code for linked list version of generic queue (LinkedQueue< E >) and generic node (Node< E >) classes is provided:

- LinkedQueue.java
- Node.java

You will also need *Transaction* class to represent transactions:

- Transaction.java

0: Read Chapter 7 and Appendix B from the textbook.

1: Add methods *peek()* and *updateFront(E item)* to LinkedQueue< E > class. Method *peek()* should return the item from the front of the queue without removing it from the queue. Method *updateFront(E item)* should replace the element at the front of the queue with the given item. Headers of the methods are the following:

```
public E peek( )
public void updateFront(E item)
```

2: When a share of common stock of some company is sold, the **capital gain** (or, sometimes, loss) is the difference between the share's selling price and the price originally paid to buy it. The rule is easy to understand for a single share, but if we sell multiple shares of stock bought over a long period of time, then we must identify the shares actually being sold. A standard accounting principle for identifying which shares of a stock were sold in such a case is to use a FIFO protocol - the shares sold are the ones that have been held the longest (indeed, this is the default method built into several finance software packages). For example, suppose we buy 100 shares at $20 each on day 1, 20 shares at $24 on day 2, 200 shares at $36 on day 3, and then sell 150 shares on day 4 at $30 each. Then, applying the FIFO protocol means that of the 150 shares sold, 100 were bought on day 1, 20 were bought on day 2, and 30 were bought on day 3. The capital gain in this case would therefore be 100*(30-20)+20*(30-24)+30*(30-36), or 100*10+20*6+30*(-6), or $940.

Write a Java program that

- prompts the user for the name of the file containing a sequence of transactions,
- reads from the file transactions of the form "buy x share(s) at $y each" or "sell x share(s) at $y each", assuming that the transactions occur on consecutive days and the values x and y are integers,
- given this input sequence of transactions, the output should be
  - the total capital gain (or loss) for the entire sequence, using FIFO protocol to identify shares, and
  - the number of shares left (if any).

- If at any point there is an attempt to sell more shares than were bought up until that moment, your program must output a message saying that and finish execution.

You may assume that transactions in a file are always in the specified format, as in the following sample transactions files: transactions.txt, transactions1.txt.

Sample run of the program may look like the following:

```
Please enter transactions file name: transactions.txt
The following transactions were read from the file:
    Buy 100 at $20
    Buy 20 at $24
    Buy 200 at $36
    Sell 150 at $30
    Buy 50 at $28
    Sell 100 at $32

Processing transaction: Buy 100 at $20
Processing transaction: Buy 20 at $24
Processing transaction: Buy 200 at $36
Processing transaction: Sell 150 at $30
Processing transaction: Buy 50 at $28
Processing transaction: Sell 100 at $32

Capital gain is $540.
There are 120 shares left.
```

Another sample run of the program may look like the following:

```
Please enter transactions file name: transactions1.txt
The following transactions were read from the file:
    Buy 10 at $10
    Buy 20 at $5
    Sell 50 at $8
    Buy 50 at $20

Processing transaction: Buy 10 at $10
Processing transaction: Buy 20 at $5
Processing transaction: Sell 50 at $8
Error: attempt to sell non-existing shares!
```

Implementation details:

- You must use class *Transaction* to represent a transaction. The code for *Transaction* class is provided: Transaction.java.
- You must use two queues in your program, one for all transactions (“all” queue) and the other one for "buy" transactions (“buy” queue).

  First, all transactions from the input file should be read and placed in the “all” queue (the one for all transactions). Then, you should remove transactions from the “all” queue, one at a time, and process each transaction in the following manner. If it is a buy transaction you place it in a "buy" queue. If it is a sell transaction you use it and transactions from the "buy" queue to compute the change in capital gain (or loss). If there is an attempt to sell more

shares than are available, then your program should output a message like "Error: attempt to sell non-existing shares" and finish execution.

- In this assignment you need to read data from a file. One way to do it is to use the Scanner class. Scanner class is described in Appendix B of the textbook. You may take a look at an example of opening a text file for reading with Scanner .
- Conversion from String to integer may be done as in the following example:

```
String str = "25";
int i = Integer.parseInt(str);
```

Specifications for all the methods which you implement should be included as comments in your code. Also, please use inline comments, meaningful variable names, indentation, formatting and whitespace throughout your program to improve its readability.

What to submit:

- Submit your source code (submit **all** *.java files you used including those which you did not modify) electronically on Canvas.