CSE 1322L – Assignment 7 (Fall 2025)

Introduction

In this assignment, we will create a program to help a fictional corporation keep track of its charity warehouse. Every year, this corporation gives away part of its farms' produce as a gesture of goodwill to the local community. This donation is then used as a write-off in its taxes.

Specifically, the corporation donates boxes or tomatoes, boxes of lettuce, and boxes of carrots at a big event it hosts at one of its farms. The program must keep track of the number of boxes of each produce that were given away as well as the current market price for that box, as the total amount given away (in dollars) will be used in the tax write-off.

Requirements

The features described below must be in your program:

- Produce class
  - Has 2 fields:
    - a String named "produceName"
    - a double named "price"
  - **Produce(String, double)**: Uses the arguments to initialize the object's fields
  - Both fields have getters
- Warehouse class
  - Note: **this class behaves like a queue**, with "storage" being where the queue's elements are stored
  - Has 5 fields:
    - An arraylist of Produce called "storage"
    - A double called "taxSavings"
    - 3 public integers called "tomatoesSold", "lettuceSold", and "carrotsSold"
  - **Warehouse()**: Initializes the arraylist. All other fields are initialized at 0
  - **void storeProduce(Produce)**: this is the Warehouse's enqueue method
  - **Produce getProduce():** This is the Warehouse's dequeue method. If the arraylist is empty, return null instead
  - **void addToTaxSavings(double):** Increments "taxSavings" using the argument
  - "taxSavings" has a getter
- FarmHand class
  - **This class must be able to run in its own thread**

- Has 3 fields:
  - A Warehouse named “warehouse”
  - A Produce named “produce”
  - An integer named “produceAmount”
- **FarmHand(Warehouse, Produce, int)**: Initializes all of the object’s fields with the provided arguments
- **void run()**: This overload does the following:
  - Enqueues “produce” into “warehouse”
  - The step above is repeated “produceAmount” times
  - Once the loop above terminates, prints “Worker storing {produceName} us done.”

- CharityWorker class
  - **This class must be able to run in its own thread**
  - Has 3 fields:
    - A static integer initialized at 0 called “nextId”
    - An integer called “workerId”
    - A Warehouse called “warehouse”
  - **CharityWorker(Warehouse)**: Assigns “nextId” to “workerId”, then increments “nextId” by 1. Assigns the argument to the appropriate field.
  - **void run():** This overload does the following:
    - Dequeues an element from “warehouse”
    - If the element is null, prints “Worker {workerId} is done working” and then terminates the method
    - Otherwise:
      - Increment the “warehouse” “taxSaving” field using the dequeued object’s “price”
      - Increment the appropriate “warehouse” public field by 1, depending on the dequeued object’s “produceName”

      For example: suppose a Warehouse currently has 52.0 for its “taxSavings”, 3 for “tomatoesSold”, 4 for “lettuceSold”, and 5 for “carrotsSold”. If a CharityWorker dequeues from the Warehouse a Produce whose produceName is “carrots” and its “price” is “3.5”, then the Warehouse’s “carrotsSold” will become 6 and its “taxSavings” will become 55.5

  - “workerId” has a getter
- Driver class
  - Create a Warehouse object
  - Prompt the user for how many boxes of tomatoes they would like to store, as well of the price of each box. Create a Produce object with “tomatoes” as its “produceName” and the price above as its “price”. Create a FarmHand with the information above.
  - Do the above for lettuce and carrots as well.

- Once all 3 FarmHand objects have been created, start their threads. **Do not start their threads before all 3 objects have been created**
- Wait for all 3 threads to terminate
- Prompt the user for the number of charity workers as the event, and then create that many CharityWorker objects
- Once all CharityWorker objects have been created, start their threads. **Do not start their threads before all objects have been created**
- Wait for all CharityWorker threads to terminate
- Once all threads have terminated, print the values currently stored in the Warehouse object's integer fields; all 3 of them
- Lastly, print the value stored in the Warehouse object's "taxSavings" field

Deliverables

- Assignment7.java (driver)
- Produce.java
- Warehouse.java
- FarmHand.java
- CharityWorker.java

Considerations

- You will get partial credit for partial work, as long as the rubric permits it.
- Despite what deliverables say, all of your classes can be submitted in a single file.
- You don't have to worry about rounding your floating-point numbers to a specific decimal place or about rounding errors. As long as your calculations are correct, you will get full credit.
- This assignment requires you to implement multithreading. You may either extend the Thread class or implement the Runnable interface.
- There won't be any rubric items checking if any race conditions have been prevented.
- Because you are only asked to create 3 Produce objects, your queue will contain only copies of those 3 objects.
    - This is being done to preserve memory and to save processing time. Otherwise, your computer would be required to create 100000 Item objects just so you could test one of the sample inputs.
- Your output will differ from the samples below due to the nature of concurrency. Specifically, the order in which your FarmHands and CharityWorkers terminate will vary between runs.
    - Still, some of the main()'s print statements must appear at specific locations in your output.
    - Remember that you can make a thread wait for another thread to terminate before continuing its own execution by using join().

Sample Output (user input in red)

```
[GoodFortune Warehouse]

Store how many boxes of tomatoes: 15000
What is the price of tomatoes? $4.88
Store how many boxes of lettuce: 20000
What is the price of lettuce? $9.94
Store how many boxes of carrots: 25000
What is the price of carrots? $2.91

Farm hands are ready to start working.
Press any key to start the work...

Work has started. Waiting for farm hands to finish...
Worker storing tomatoes is done.
Worker storing lettuce is done.
Worker storing carrots is done.
Warehouse is fully loaded.

Assign how many workers at the warehouse? 20
All workers are at their stations.
Press any key to start charity event...

Charity event started...
Worker 6 is done working
Worker 5 is done working
Worker 7 is done working
Worker 10 is done working
Worker 8 is done working
Worker 11 is done working
Worker 9 is done working
Worker 13 is done working
Worker 14 is done working
Worker 1 is done working
Worker 12 is done working
Worker 16 is done working
Worker 15 is done working
Worker 19 is done working
Worker 17 is done working
Worker 18 is done working
Worker 2 is done working
Worker 0 is done working
```

```
Worker 3 is done working
Worker 4 is done working

The event was a success!
A total of 15000 boxes of tomatoes were given away.
A total of 20000 boxes of lettuce were given away.
A total of 25000 boxes of carrots were given away.
A total of $344750.00 was made in tax savings.
```

Sample Output (user input in red)

```
[GoodFortune Warehouse]

Store how many boxes of tomatoes: 35000
What is the price of tomatoes? $5.02
Store how many boxes of lettuce: 45000
What is the price of lettuce? $10.01
Store how many boxes of carrots: 20000
What is the price of carrots? $2.99

Farm hands are ready to start working.
Press any key to start the work...

Work has started. Waiting for farm hands to finish...
Worker storing carrots is done.
Worker storing tomatoes is done.
Worker storing lettuce is done.
Warehouse is fully loaded.

Assign how many workers at the warehouse? 30
All workers are at their stations.
Press any key to start charity event...

Charity event started...
Worker 8 is done working
Worker 27 is done working
Worker 7 is done working
Worker 20 is done working
Worker 15 is done working
Worker 13 is done working
Worker 0 is done working
Worker 6 is done working
Worker 5 is done working
```

```
Worker 2 is done working
Worker 4 is done working
Worker 24 is done working
Worker 23 is done working
Worker 3 is done working
Worker 28 is done working
Worker 29 is done working
Worker 26 is done working
Worker 9 is done working
Worker 25 is done working
Worker 22 is done working
Worker 21 is done working
Worker 19 is done working
Worker 17 is done working
Worker 18 is done working
Worker 12 is done working
Worker 14 is done working
Worker 16 is done working
Worker 1 is done working
Worker 11 is done working
Worker 10 is done working

The event was a success!
A total of 35000 boxes of tomatoes were given away.
A total of 45000 boxes of lettuce were given away.
A total of 20000 boxes of carrots were given away.
A total of $685950.00 was made in tax savings.
```