CSE1322L Assignment 2 - Fall 2023

Introduction:

In places where buses are a common mode of transportation, bus terminals may employ a dispatcher. As the name implies, a dispatcher's job is to dispatch buses from their terminal. This position may be deemed necessary because, as traffic conditions change throughout the day due to natural jams or accidents, the concentration of buses throughout their service area may become skewed.

To maintain a steady flow of buses and ensure that no passenger waits too long in any given bus stop at any given time, dispatchers monitor the inflow of buses to their terminal and only allows them to leave when appropriate. We will code a simple program which allows a dispatcher to keep track of what buses are in their terminal, as well as their passengers.

This assignment will make use of a data structure which in Java is called "Array List" and in C# is called "List". Going forward, both will be simply referred to as a "list", for simplicity.

Requirements

The features described below must be in your program.
- A total of four classes: the driver, Person, Bus, and Dispatcher
- The Person class must have 3 fields:
    - A string name, which is private.
    - An integer ticketNumber, which is private.
    - A static integer nextTicketNumber, which is also private.
- It must have 2 constructors:
    - A default constructor which assigns "NO NAME" to name, assigns nextTicketNumber to ticketNumber, and updates nextTicketNumber by one.
    - An overloaded constructor which takes in a string. This string is used to set the object's name, with ticketNumber and nextTicketNumber being used and updated as in the default constructor.
    - An override of the toString() method, which returns the objects information in the following format:
        - [ticketNumber]\t[name]
        - E.g.: 1234   Dmitri
- The Bus class must have 3 fields:
    - A list of Persons called passengers
    - A static integer called nextId, which is private
    - An integer busId, which is private

- It must have a single constructor, which initializes passengers, assigns nextId to busId, and updates nextId by one.
- A getter for busId called getId()
- It also features the following methods:
  - addPerson(Person person): A method which takes in a Person as a parameter and returns nothing. It adds the person in the parameter to passengers.
  - removePerson(Person person): A method which takes in a Person as a parameter and returns a Boolean value. If the Person in the parameter was successfully removed from passengers, removePerson() must return true. Otherwise, it must return false.
  - findPerson(string name): A method which takes in a string and returns a Person. It must traverse through passengers looking for the Person whose name matches the one in the parameters. If the Person is found, it must be returned. If the Person isn't found, findPerson() must return null.
  - transferPerson(Bus bus1, Bus bus2, Person person): A static method which returns a Boolean. It takes in two buses and a Person as its parameters. If it finds the Person in the parameters in bus1, it must remove that person from bus1, add them to bus2, and then return true. Otherwise, it must return false.
  - getPassengers(): A method which takes in no parameters and returns a string. It must traverse passengers and return the toString() method of each passenger in a different line.
  - An override of the toString() method, which simply returns the busId.
- The Dispatcher has a single field: a list of buses called busQueue.
- It has no constructors and has the following methods:
  - addBus(Bus bus): A method which takes a single Bus as a parameter and returns an integer. It must add said Bus to the end of the list and then return the position on the list where the Bus was added (starting at 0).
  - addBus(Bus bus, int position): A method which takes a Bus and an integer as parameters and returns an integer. It tries to insert the Bus in the parameters into the position in the parameters. If it fails (e.g.: tries to insert the bus in position 50 when there are only 3 buses in busQueue), it must add the Bus to the current end of busQueue. It must then return the position on the list where the bus was added (starting at 0).
  - findBus(int id): A method which takes in an integer and returns a Bus. If there is a Bus in busQueue whose id matches the one in the parameters, the method must return that Bus. Otherwise, it must return null.
  - removeBus(int id): A method which takes in an integer and returns a Bus. If there is a Bus in busQueue whose id matches the one in the parameters, the method must remove that bus from busQueue and return that Bus. Otherwise, it must return null.
  - dispatchBus(): A method which takes in no parameters and returns a Bus. If busQueue is empty, it must return null. Otherwise, it must remove the first Bus from busQueue (the one at position 0) and return it.

- An override of the toString() method, which It must traverse busQueue and return the toString() method of each Bus in a different line.
- The Driver must create a Dispatcher object and, inside a loop, must have a menu with the following options:
    1. Add bus: Creates a Bus and passes it to the Dispatcher's addBus(). It then prints "Bus [ID] added to position [POSITION]", where ID is the Bus id and POSITION is where in the Dispatcher's list the Bus was added.
    2. Add Person to bus: It asks for a Bus id and passes it to the Dispatcher's findBus(). If no Bus is found, it must print "No bus with id [ID]". If a Bus is found, it must then ask for the Person's name and call the Buses' addPerson() method. It must then print out "[NAME] has been added to bus [ID]".
    3. Remove bus: It must ask for a Bus id and pass it to the Dispatcher's findBus(). If no bus is found, it must print out "No bus with id [ID]". Otherwise, it must call the Dispatcher's removeBus() and print "Bus [ID] removed".
    4. Remove person: It must ask for a Bus id and pass it to the Dispatcher's findBus(). If no bus is found, it must print out "No bus with id [ID]". Otherwise, it must ask for a person's name and pass it to the Buses' findPerson() and removePerson() to try to remove the Person. If a Person is removed print "[NAME] has been removed from bus [ID]". Otherwise, print "No such person found in bus [ID]".
    5. List passengers: It must ask for a Bus id and pass it to the Dispatcher's findBus(). If no bus is found, it must print out "No bus with id [ID]". Otherwise, it prints "Bus [ID]" in one line and, in the next line, a call of the Buses' getPassengers().
    6. List buses: prints out "BUS QUEUE", followed by the Dispatcher's toString().
    7. Requeue bus: It must ask for a Bus id and pass it to the Dispatcher's findBus(). If no bus is found, it must print out "No bus with id [ID]". Otherwise, it must ask for a new position for the bus in busQueue. It calls the Dispatcher's addBus(), passing both the new position and the Bus. It must then print out "Bus [ID] added to position [POSITION]".
    8. Transfer person: This option must ask for a Person's name, and two Bus ids. It must then use the Dispatcher's findBus() and the first Bus findPerson() method to determine if the buses exist and if the Person exist, respectively. If one of them does not exist, print an appropriate message ("No bus with id [ID]" for the buses and "No person named [NAME]" for the Person). It must then call transferPerson(), printing out "Person transferred successfully" if it succeeds and "Person transfer failed" if it didn't.
    9. Calls the Dispatcher's dispatchBus(). If busQueue is empty, it must print "Bus queue is empty". Otherwise, it prints "Bus [ID] has been dispatched", with ID being the ID of the Bus that was dispatched.
    10. Terminates the program.

Considerations

- This assignment may seem intimidating, but that's just because of the number of things you have to do; the assignment itself isn't very hard, so don't be discouraged.
- Remember that you will get partial credit for partial work. Try to deliver as much of the assignment as you can.
- You may add helper methods in the Driver, but not in any of the other 3 classes: **The rubric will assume that you call the appropriate methods in Bus, Person, and Dispatcher, and you may lose points if you fail to call those methods appropriately**.

Example: [User input in red]

Bus Dispatching System

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
1

Bus 0 added to position 0

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
1

Bus 1 added to position 1

1. Add bus

2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
1

Bus 2 added to position 2

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
3

Enter bus id: 1
Bus 1 removed

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
6

BUS QUEUE
0
2

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
2

Enter bus id: 0
Enter person's name: Alice
Alice has been added to bus 0

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
2

Enter bus id: 0
Enter person's name: Bob
Bob has been added to bus 0

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
2

Enter bus id: 0
Enter person's name: Charlie
Charlie has been added to bus 0

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
2

Enter bus id: 2
Enter person's name: David
David has been added to bus 2

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
2

Enter bus id: 1
No bus with id 1

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person

9. Dispatch bus
0. Exit
2

Enter bus id: 2
Enter person's name: Eve
Eve has been added to bus 2

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
5

Enter bus id: 0
Bus 0
0   Alice
1   Bob
2   Charlie

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
4

Enter bus id: 0
Enter person's name: Bob
Bob has been removed from bus 0

1. Add bus

2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
5

Enter bus id: 0
Bus 0
0   Alice
2   Charlie

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
7

Enter bus id: 0
Enter new bus position: 1000
Bus 0 added to position 1

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
6

BUS QUEUE
2
0

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
8

Enter person's name: Alice
Enter id of bus 1: 0
Enter id of bus 2: 2
Person transferred successfully.

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
5

Enter bus id: 0
Bus 0
2   Charlie

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person

5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
5

Enter bus id: 2
Bus 2
3   David
4   Eve
0   Alice

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
9

Bus 2 has been dispatched

1. Add bus
2. Add Person to bus
3. Remove bus
4. Remove person
5. List passengers
6. List busses
7. Requeue bus
8. Transfer person
9. Dispatch bus
0. Exit
0

Shutting down.

Submitting your answer:

Please follow the posted submission guidelines here:
https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:
https://ccse.kennesaw.edu/fye/courseschedules.php