Metropolitan University of Tirana
Data Structures
Queues in Java

8.1 Grocery Store Simulation using a Queue

You have been hired by a grocery store to optimize their checkout process. To do this, you need to simulate the checkout process using a queue data structure.

Task

Your task is to write a program that simulates the checkout process at a grocery store using a queue. The simulation should model the following scenario:

- Customers arrive at the store and join a single queue.
- Each customer has a name, the number of items they are buying, and the amount of time it takes to scan each item.
- The checkout process is modeled as follows:
  - The first customer in the queue is processed.
  - Each item is scanned, taking the specified amount of time for each item.
  - Once all items have been scanned, the customer leaves the checkout area and the next customer is processed.
- The simulation should end once all customers have been processed.

Requirements

- Use a Queue data structure to model the line of customers waiting to checkout.
- Use an object-oriented design for your solution. Create a Customer class to represent each customer in the queue.
- Use random numbers to generate the number of customers, the number of items each customer has, and the amount of time it takes to scan each item.
- Print out the following statistics at the end of the simulation:
- The total number of customers processed.
- The average amount of time each customer spent in the checkout area.
- Test your program with different parameters (e.g., different number of customers, different item scanning times, etc.).

8.2 Bank Waiting Line Simulation using a Queue

The goal of this task is to simulate the operation of a bank and measure the waiting time for customers in the queue. The simulation should be randomized to account for the variability in customer arrivals and transaction times. The waiting time for each customer should be calculated based on the difference between their arrival time and the time they are served. Finally, the statistics should be calculated and printed out to provide insights into the bank's performance.

1. Create a class `Customer` that has the following attributes:
   - `id`: a unique identifier for each customer.
   - `arrivalTime`: the time the customer arrived at the bank.
   - `transactionTime`: the time the customer will take to complete their transaction.
2. Create a class `Bank` that has the following attributes:
   - `queue`: a queue of Customer objects waiting in line.
   - `totalCustomers`: the total number of customers that will visit the bank.
   - `currentTime`: the current time of the simulation.
   - `maxTransactionTime`: the maximum amount of time a customer can take to complete their transaction.
   - `maxArrivalInterval`: the maximum time between customer arrivals.
3. Create a method `simulate()` in the Bank class that simulates the bank's operation. The method should perform the following steps:
   - Initialize a list of customers with random arrival times and transaction times.
   - Add customers to the queue as they arrive, based on their arrival times.
   - Process customers in the queue one at a time, based on their transaction times.
   - Calculate the waiting time for each customer.
   - Calculate the average waiting time for all customers.
4. Create a method `printStatistics()` in the `Bank` class that prints out the following statistics:
   - Total number of customers served.
   - Average waiting time per customer.
   - Maximum waiting time for any customer.
   - Number of customers that had to wait more than a specified time threshold (e.g., 5 minutes).
5. In the `main` method, create a `Bank` object, call the `simulate()` method to simulate the bank's operation, and then call the `printStatistics()` method to print out the statistics.