# Cashiers and Customers Simulation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| C | C | C | C | C |  |  |  |
| C | C | C |  |  |  |  |  |
| C | C | C | C | C | C |  |  |
| C | C | C | C |  |  |  |  |

ArrayList of Queues of Customers

Grocery stores typically have many cashiers, each of whom checks out the groceries of each customer in his/her queue. The customers always choose to join the shortest queue. Customers with a few groceries are checked out quickly while customers with many groceries take a longer time. The picture shows four cashiers in the process of checking out the first customer in each of four queues.

Suppose a grocery store wants to gather data about how long it takes to serve each customer, from the minute they enter queue to the minute they check out. The grocery store hires you to program a *simulation* of the situation.

* You need an arrayList of the correct size to represent the cashiers.
* You need a queue of Customers for each cashier in the arrayList.
* You need to know how often, for each minute, that customers join the queue.
* For each new Customer, you need to find the shortest queue. If two queues are equally short, just join one of the queues.
* You need to know how long it takes (a random number) each customer to check out when they reach the cashier, a random integer between 2 and 6 minutes, inclusive. The file will display the minutes each customer spends at the cashier, counting down to 0, when the customer leaves the queue.
* You need to store the time at which the customer entered the queue.
* You need to calculate how long the customer spent getting served (the total wait time), from the minute they entered the queue to the minute they checked out.
* You need to know how long (10 minutes? 60 minutes? 8 hours?) to run the simulation.
* How should we handle the customers in the queues when the time is up?

For each minute write to a file:

* The minute.
* The cashiers with their queues of customers.
* The front of the queue will be counting down the time spent while checking out.

Display these calculations to the screen:

* The number of customers who joined the queues.
* The number of customers who were checked out.
* The average wait time of those who were checked out.
* The longest wait time of those who were checked out.
* The length of the longest queue of customers.

**Customer Class Design Considerations**

Customer class. What private data should it store?

constructor?

instance methods?

**Algorithm Design**

What happens in each minute?

**Output**

The shell for the Cashier simulation does a lot of work for you. Use it!

Try using 4 cashiers, then 5, then 6, for 60 minutes each.

This sample run uses 3 cashiers for 10 minutes.

How many customers were checked out?

Why did 3 change to 2?

|  |  |
| --- | --- |
| outfileCashiersAndQueues to the file:  What does the 3 represent? | Print the statistics to the screen: |
| minute 0:  3 cashiers  4  3    minute 1:  3 5  2  1  minute 2:  2 5  1 6  6  minute 3:  1 5 2  6  5 4  minute 4:  5 2  5 5 4  4 4  minute 5:  4 2 5  4 5 4  3 4 5  minute 6:  3 2 5 4  3 5 4 2  2 4 5  minute 7:  2 2 5 4 6  2 5 4 2  1 4 5 4  minute 8:  1 2 5 4 6  1 5 4 2 3  4 5 4 2  minute 9:  2 5 4 6 4  5 4 2 3  3 5 4 2 3 | Cashiers and Customers Simulation!  How many cashiers? 3 How long, in minutes, should the simulation run? 10 Number of cashiers = 3 Customers joining each minute = 2 Customers who joined queues = 20 Customers who were checked out = 6 Average wait time of those who were checked out = 5.8 Longest wait time of those who were checked out = 9 Longest queue of customers = 5  When the time is up, ignore these customers.  Look at the output. Can you count 6 customers who were checked out?  Look at the output. Can you count 20 customers?  How long will this customer take from joining the queue to checking out? What calculation do you make? |

**Hints for passing the Codepost tests:**

Your outfile must be in the right format! Codepost will process your outfile to see if your calculations are correct.

1. Use the given variables and methods, not your own.
2. Start the outfile from minute 0.
3. Outfile every minute, even if all the queues happen to be empty.
4. First do the processing on the queues, then outfile the queue.
5. Debugging strategy: change the CHANCE\_OF\_CUSTOMER to 1 and run it for 10 minutes. Now the activity recorded in your outfile is so small that you can count and calculate by hand.