

Semaphores

Coordinators

- waitingRoomCoord = new Semaphore(1, true);
- agentCoord = new Semaphore(1, true);

Both of these Semaphores are used by the customers to ensure that access to their respective communication resources are mutually exclusive.

The first, waitingRoomCoord, requires that only one customer can be brought into the Agent Line at a time.

The second, agentCoord, requires that only one customer can be picked from the line at a time. These allow uninterrupted communication.

Info Desk Access

- infoDeskWait = new Semaphore(0, true);
- infoDeskTakeNum = new Semaphore(0, true);
- infoDeskTakeNumWait = new Semaphore(0, true);

These Semaphores are used by the info desk and the customers to coordinate number assignments. Each of them is 0 because they are checkpoints that should only be passed when signaled by another entity.

infoDeskWait has the info desk wait until a customer signals its availability.

infoDeskTakeNum has the customer wait for the Info Desk to acknowledge the customer's first signal and send a number.

infoDeskTakeNumWait has the Info Desk wait for the Customer to take the number before each is released on.

Announcer Control

- announcerWait = new Semaphore(0, true);
- waitToGetInAgentLine = new Semaphore(0, true);
- spaceInLine = new Semaphore(4, true);

The Announcer Semaphores control Customers who have received a number and call them forward by the assigned numbers.

announcerWait has the announcer wait until a customer is in the waiting area with a number before proceeding to fill the line.

waitToGetInAgentLine has the customer wait until the Announcer calls them to join the agent line.

spaceInLine has the Announcer wait to call a number until there is space in the line, allowing for 4 customers to be in the line at a time until they are called by the Agents

Agent Line Semaphores

- `agentWait = new Semaphore(0, true);`
- `agentExamWait = new Semaphore(0, true);`

These Semaphores are for the agents interacting with customers in the agent line. Like the info desk, they should only be passed if signaled to, and so all start with 0. This is also where `spaceInLine` is called to free up space in the line after the agent calls.

`agentWait` has the agents wait until a customer is at the front of the line, then one of them takes that customer by calling them.

`agentExamWait` has the Customer wait until an agent calls them forward from the line.

Agent Exam Semaphores

- `custFinishesExam1 = new Semaphore(0, true);`
- `custFinishesExam2 = new Semaphore(0, true);`
- `waitForLicense1 = new Semaphore(0, true);`
- `waitForLicense2 = new Semaphore(0, true);`

This set of Semaphores, one pair for each agent, facilitate the back and forth actions of the customer and their agent.

`custFinishesExam1` and `custFinishesExam2` have the agent wait until the customer finishes the exam.

`waitForLicense1` and `waitForLicense2` have the customer wait until the agent hands them their license.