Funcionamiento y diseño del programa

El diseño del programa es el siguiente:

1. Un cliente que conoce el buffer, tiene un número de mensajes y también una id.
2. Un buffer que tiene una cola para los mensajes que envían los clientes, otra cola para las respuestas que envían los servidores, un valor correspondiente al tamaño del buffer y dos objetos llamados lleno y vacío usados para la sincronización.
3. Un servidor que únicamente necesita conocer el buffer con el que interactúa.
4. Un mensaje con el id del cliente al que corresponde, un valor de consulta que es el que el cliente envía y la respuesta que el servidor eventualmente le proveerá.

El funcionamiento del programa se da de la siguiente manera:

1. Al ejecutarse, el cliente crea un mensaje con su respectivo contenido.
2. El cliente revisa si es posible encolar el mensaje en el buffer. Si no es posible, se queda en una espera activa donde sigue intentando encolar su mensaje, pero cede (yield()) el procesador luego de cada intento. Este proceso de encolado donde el cliente se comunica con el buffer se sincroniza utilizando un synchronized() sobre la cola de mensajes, de manera que nadie más pueda poner algo ahí mientras el encolado se cumple. Luego de que encola su mensaje, notifica a algún servidor de la existencia del mensaje y duerme hasta que un servidor notifique que existe una respuesta a su mensaje.
3. Mientras tanto, el servidor se está ejecutando al tiempo y verificando si hay mensajes en el buffer.
4. Cuando un mensaje entra al buffer, el servidor lo intenta desencolar de ahí. En el proceso de desencolado, el servidor revisa si el buffer está vacío y si es así, se queda durmiendo hasta que un cliente encole un mensaje y lo despierte. La sincronización en este caso se hace de dos maneras. La primera sobre un objeto de vacío para que no ocurran cambios en esa condición mientras se revisa el estado del buffer, y la segunda sobre este método mientras el servidor remueve el mensaje de la cola para preparar la respuesta, así nadie puede entrar en conflicto al intentar tomar el mismo mensaje al mismo tiempo.
5. Ahora que el servidor ha tomado el mensaje, pone la respuesta debida en este y lo vuelve a encolar en el buffer. El encolado de esta respuesta también se debe sincronizar. Por lo tanto, se revisa que el buffer no esté lleno. Si es así, se pone a dormir hasta que alguien lo despierte al dejar espacio disponible en el buffer. Este proceso de revisión se sincroniza sobre la cola para no hayan modificaciones en ese momento. Luego, si es posible encolar la respuesta, se sincroniza sobre este método para evitar cambios en la cola de mensaje, se pone la respuesta y se notifica al cliente para que despierte.
6. Ahora, el cliente despierto llega al buffer, se sincroniza sobre la cola de mensajes de este para que no hayan cambios y saca su mensaje de ahí y como extra, para comprobar funcionamiento, imprime la respuesta en consola.