**Mesanjes Class (multiple)**

- Have a Client's query and Server's answer as variables

-

**Cliente Class (multiple)**

- Create & initializes queries (random number) as a Message object, tries to store (send to) it in the buffer; expects answer to every query

- If Buffer is full, the Client uses active waiting.

- A Client must "yield" the processor after every attempt to store a message

- If successfully stored, it must wait for an answer using passive waiting (in the Buffer)

- Receives answer (original number ++) from a server, then print and discard

- When a client has sent all queries it notifys the buffer it is leaving

- Does not directly communicate with servers

- Client communicates to Server through a Buffer

= Constructor that specifies number of queries

= N number of times one thread will query. After all queries answered, the thread ends

= Method that takes in number of clients and number of queries

= Generates a Message object (query), each query is a Messsage object

**Buffer Class**

- Receives queries from client

- Stores queries

- Make queries available to Server for picking and answering....??

- Has a defined storage space (specified in file)

- Needs to know the number of clients in the system, because if there are no Clients the Server threads can end

- However it doesn't know the number of messages

**Server Class (multiple)**

- Try to receive queries from the buffer, answer queries stored in the buffer (increase value by 1, notify associated client that it can continue)

- If it cannot retrieve a message, it will use active waiting to try again, however must yield the processor after every attempt

- If successfully retrieved a message, it generates an answer and notifies the associated Client

- Tell the client that sent the query (notify?? through the buffer??) that an answer is ready

- Server thread runs as long as there are clients, ends when all Clients have finished

- Server communicates to Client through a Buffer

= Is a thread

= Constructor that specifies number of queries

***\* Communication scheme to handle queries from multiple clients***

***\* To synchronize thread behavior you only may use the following***

Java instructions: synchronized, wait, notify, notifyAll and yield.

***\* File I/O to read from a file the:***

- Number of clients

- Number of servers

- Number of queries per client

- Buffer size

***\* Main method reads the File, and initializes:***

- Number of clients

- Number of servers

- Number of queries per client

- Buffer size

***\*\*Questions\*\****

Is the server also a thread??

By "continue" from the Server notification, it means the client thread can send another query message or end?

Where is the main class? In the Buffer or in a separate RunProgram class??

Does the Server request queries from the Buffer or does the Buffer send it to the Server? - Buffer = Producer & Server = Consumer?

What is "yield" again?? - give time to other thread???

***\*\*\* They will test with 5 different config file scenarios\*\****