Exercise Five: Java Network

1. (50 Points) A Simple Chatting Socket Program

Review the programs Socket Client and Socket Server for chat that you learned in the lecture. And modify the programs using Java Thread to show the same result.

An example of the main method in the Client of your answer:

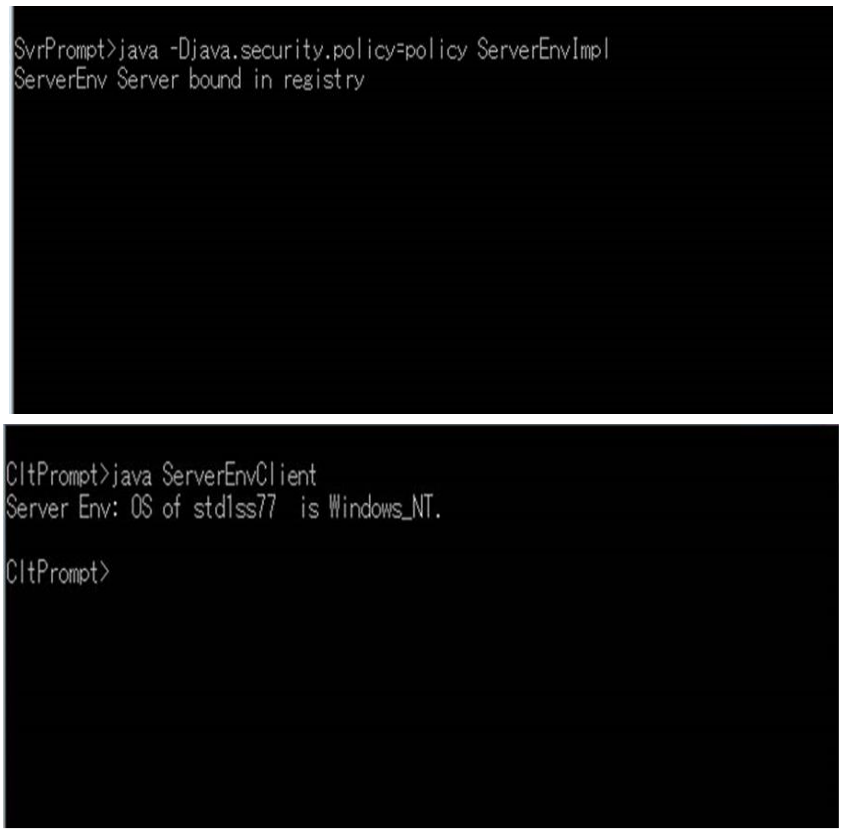
```
-----
public static void main(String args[]) {
    Client c1 = new Client();
    c1.start();
}
-----
```

<Running Example>

2. (50 Points) A Simple RMI Application

let's create an RMI application.

- This application shows name of the server computer and its operating system.
- Let's suppose that you installed and ran the server (RMI) on a remote machine. (Of course, you can run the server on a different shell of the same machine.)
- The client calls a remote interface to get environment value of the computer and its OS at remote side, and then prints the values on screen as the figure below.



3. (Bonus Question: 20 Points) A Simple Mail Application

The objective of this assignment is to implement a server and two clients that exchange messages through the server, as shown in Figure 1.

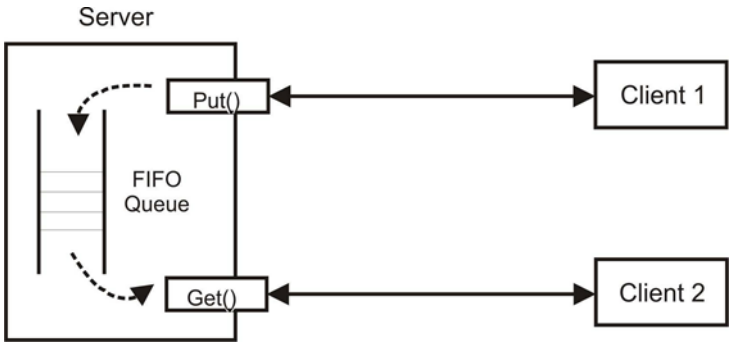


Figure 1: The Message Server

- The server **MessagePoolServer** implements two remote methods defined in the interface class **MessagePool**:

```
import java.rmi.*;

public interface MessagePool extends Remote {
    public void put(String msg) throws RemoteException;
    public String get() throws RemoteException;
}
```

Figure 2: The remote interface MessagePool.java

- the **put()** method accepts a String message from the client, and stores it into the FIFO (First In First Out) queue in the server. In case the queue is full, the **put()** operation will fail, and a **QueueFullException** exception will be thrown. In case the message from the client is null, the server will throw a **MessageNullException** exception.
- the **get()** method retrieves the message out of the queue to the client which invoked it. The retrieved message will be deleted from the queue. In case the queue is empty, the **get()** operation will fail and a **QueueEmptyException** exception will be thrown.

- The FIFO message queue **MessageQueue** has a maximum size of 100 messages. Messages are strings with the maximum length of 100 characters.

- Implement two clients: **MessagePutClient.java** generates messages periodically (1 message per 1 second), and **MessageGetClient.java** retrieves messages periodically (1 message per 2 seconds). The message could be the timestamp of the client or any random generated string.

- Make sure all cases are handled:
  - Retrieving from an empty queue.
  - Trying to add a message to a full queue.

- You can choose Java socket or RMI to implement your program. Of course, using both Java socket and RMI to implement your program and comparing them is highly recommended.