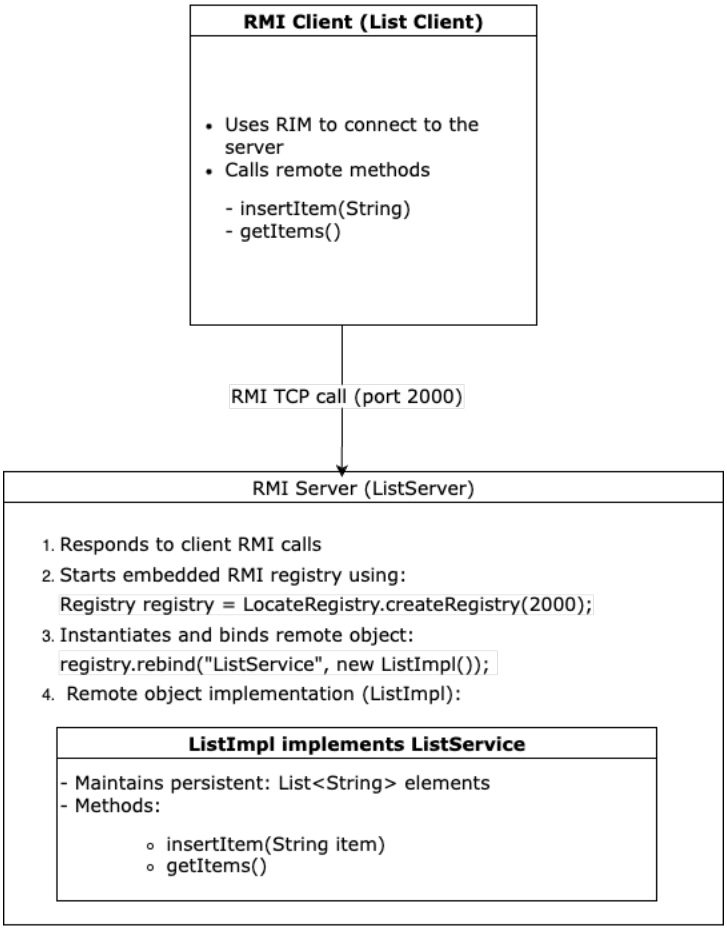
COLLABORATIVE SESSION 1

Name: Sk Md Shariful Islam Arafat
Student ID: a1983627
Number of Group Members: 1

Introduction:

RPC is the remote procedure call whereas RMI (Remote Method Invocation) is the JAVA implementation of RPC. It's a client-server architecture.



RMI Client (List Client)

- Uses RIM to connect to the server
- Calls remote methods

  - insertItem(String)
  - getItems()

RMI Server (ListServer)

1. Responds to client RMI calls
2. Starts embedded RMI registry using:
   `Registry registry = LocateRegistry.createRegistry(2000);`
3. Instantiates and binds remote object:
   `registry.rebind("ListService", new ListImpl());`
4. Remote object implementation (ListImpl):

ListImpl implements ListService

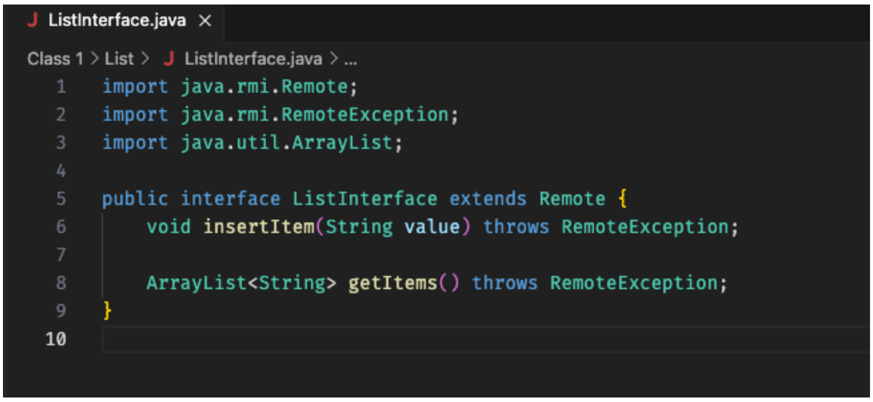
- Maintains persistent: List<String> elements
- Methods:
  - insertItem(String item)
  - getItems()

Experience:

To achieve the expected result, I started from a template to build a calculator including an interface class, its implementation class, a server class and client class. With this I'm able to submit two numbers from the client side and the server will return the sum of the numbers.

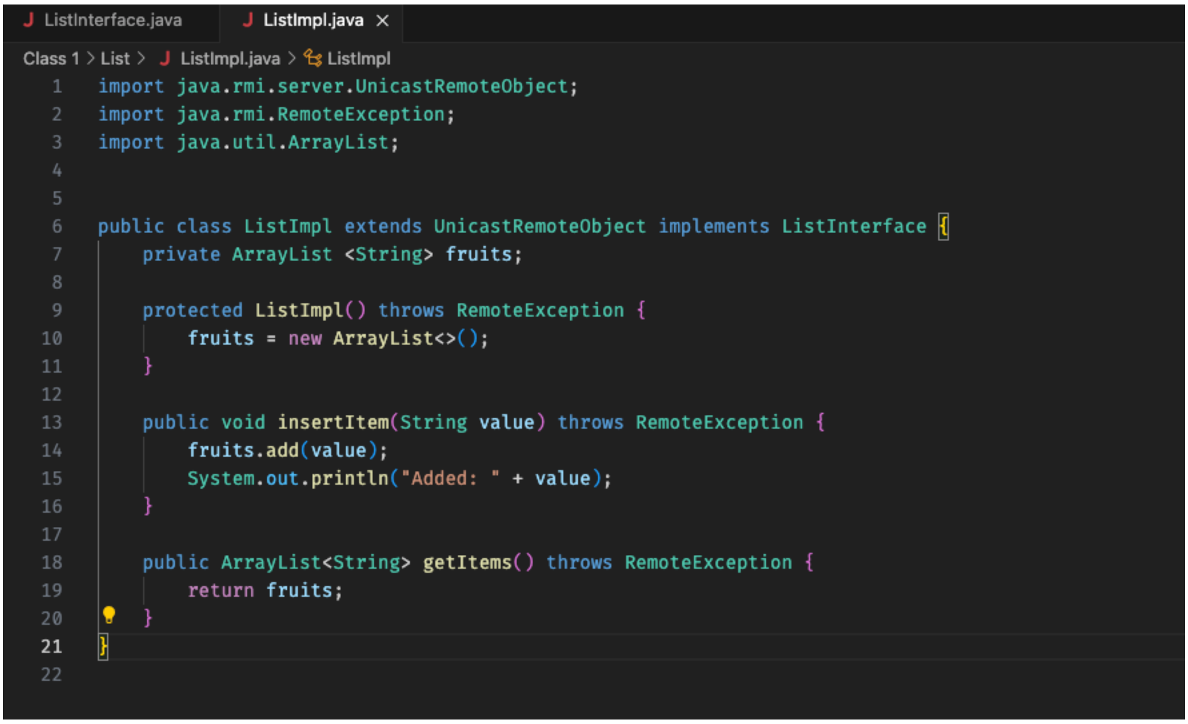
Then in the similar structure, I have implemented the List where the server maintains a persistent state. (e.g. adding fruits into the array and show the full list)

ListInterface.java



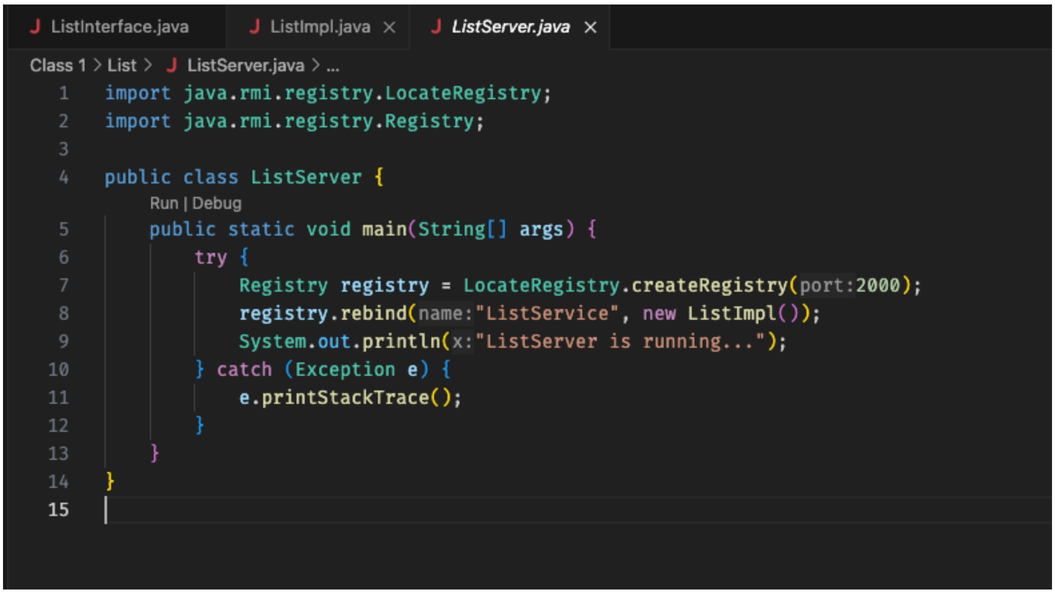
```java
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.util.ArrayList;

public interface ListInterface extends Remote {
    void insertItem(String value) throws RemoteException;

    ArrayList<String> getItems() throws RemoteException;
}

```

ListImpl.java



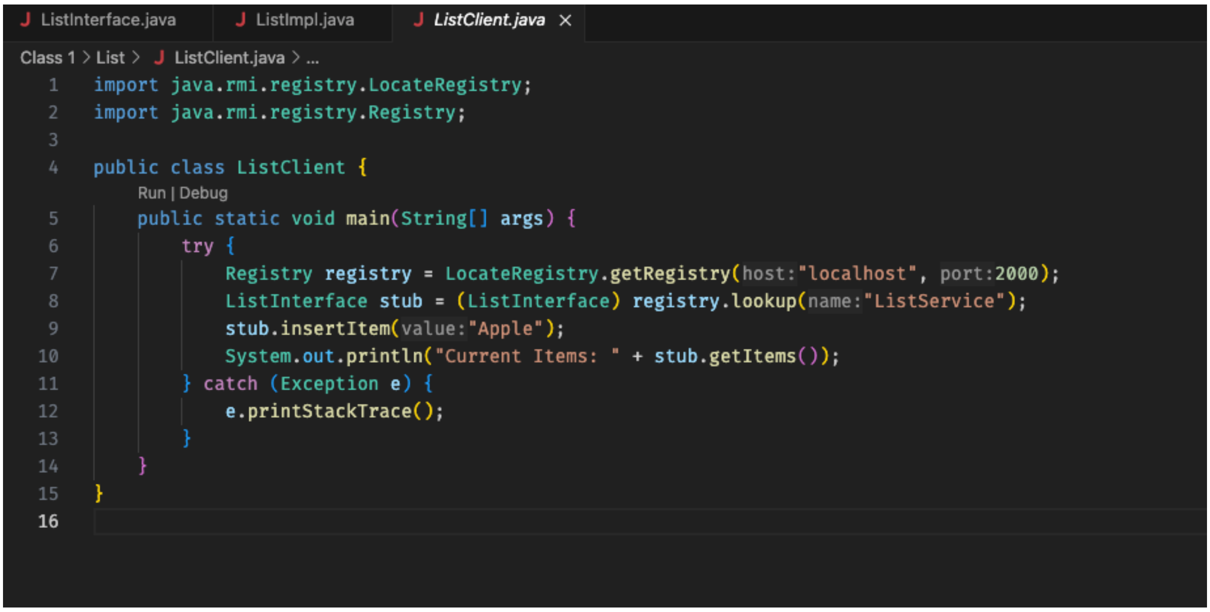
```java
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
import java.util.ArrayList;


public class ListImpl extends UnicastRemoteObject implements ListInterface {
    private ArrayList <String> fruits;

    protected ListImpl() throws RemoteException {
        fruits = new ArrayList<>();
    }

    public void insertItem(String value) throws RemoteException {
        fruits.add(value);
        System.out.println("Added: " + value);
    }

    public ArrayList<String> getItems() throws RemoteException {
        return fruits;
    }
}

```

ListServer.java



```java
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class ListServer {
    public static void main(String[] args) {
        try {
            Registry registry = LocateRegistry.createRegistry(port:2000);
            registry.rebind(name:"ListService", new ListImpl());
            System.out.println(x:"ListServer is running...");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

ListClient.java



```java
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class ListClient {
    public static void main(String[] args) {
        try {
            Registry registry = LocateRegistry.getRegistry(host:"localhost", port:2000);
            ListInterface stub = (ListInterface) registry.lookup(name:"ListService");
            stub.insertItem(value:"Apple");
            System.out.println("Current Items: " + stub.getItems());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Acceptance Testing:

1. Run the List Server, which is showing “ListServer is running...”
2. Run the List Client, which has added “Apple”
3. Server Terminal Showing “Added: Apple”
4. Everything is working as expected