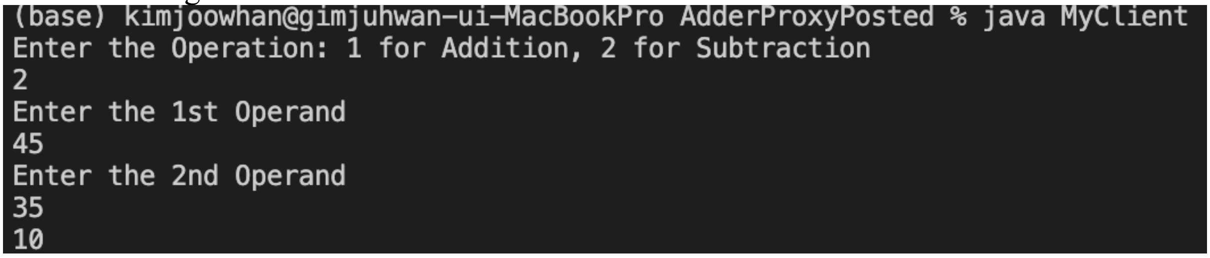
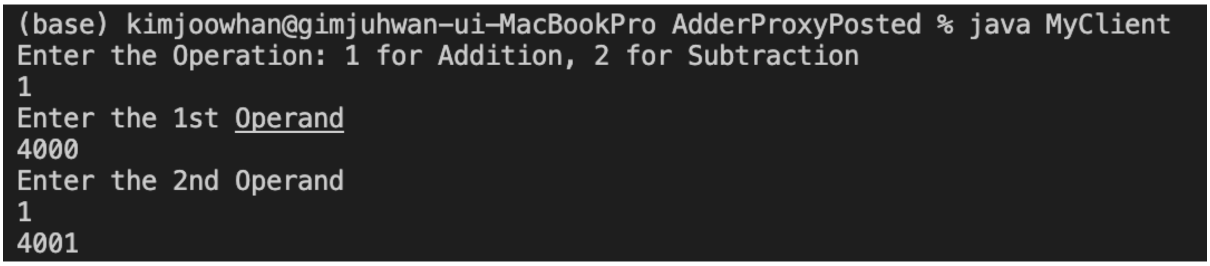
21700165 Joowhan Kim(김주환)

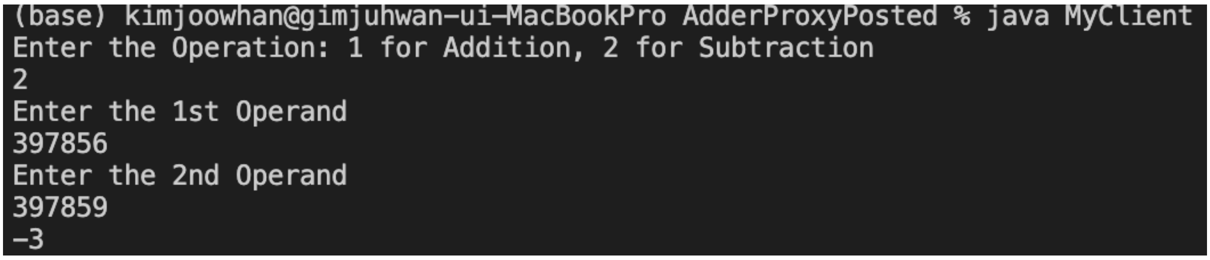
Results of Program



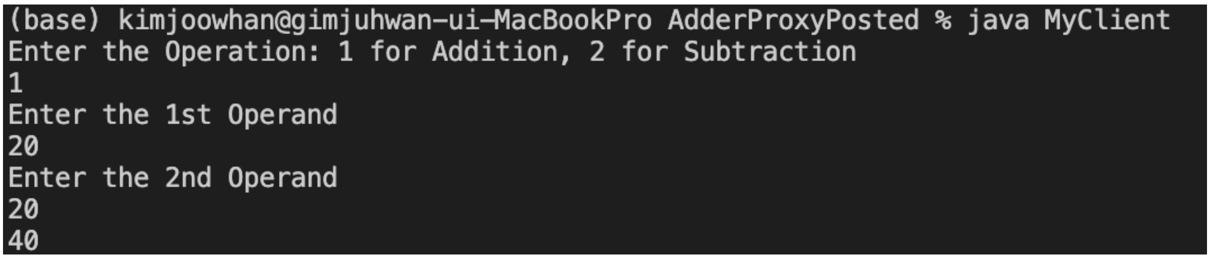
```
(base) kimjoowhan@gimjuhwan-ui-MacBookPro AdderProxyPosted % java MyClient
Enter the Operation: 1 for Addition, 2 for Subtraction
2
Enter the 1st Operand
45
Enter the 2nd Operand
35
10
```



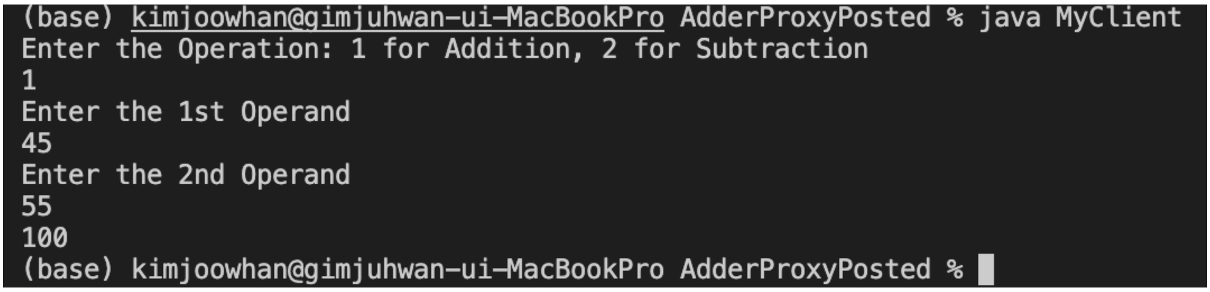
```
(base) kimjoowhan@gimjuhwan-ui-MacBookPro AdderProxyPosted % java MyClient
Enter the Operation: 1 for Addition, 2 for Subtraction
1
Enter the 1st Operand
4000
Enter the 2nd Operand
1
4001
```



```
(base) kimjoowhan@gimjuhwan-ui-MacBookPro AdderProxyPosted % java MyClient
Enter the Operation: 1 for Addition, 2 for Subtraction
2
Enter the 1st Operand
397856
Enter the 2nd Operand
397859
-3
```



```
(base) kimjoowhan@gimjuhwan-ui-MacBookPro AdderProxyPosted % java MyClient
Enter the Operation: 1 for Addition, 2 for Subtraction
1
Enter the 1st Operand
20
Enter the 2nd Operand
20
40
```



```
(base) kimjoowhan@gimjuhwan-ui-MacBookPro AdderProxyPosted % java MyClient
Enter the Operation: 1 for Addition, 2 for Subtraction
1
Enter the 1st Operand
45
Enter the 2nd Operand
55
100
(base) kimjoowhan@gimjuhwan-ui-MacBookPro AdderProxyPosted % █
```

- Logic of program

1. At first, compile all the java files
   Javac *.java
2. Start rmi registry in one command prompt. This example, 5000 means port number is 5000.
   Rmiregistry 5000
3. Start the server in another command prompt.

   java Myserver

   In Myserver.java, On the server, the ‘stub’(skeletion) acts as a proxy. Therefore, when exchanging data with the client, information is converted and exchanged through this proxy.

4. Start the client application in another command prompt

   java MyClient

   The client has two options, plus and minus. Select whether to add or subtract from the client and enter two numbers. And add or subtract two numbers through adder and minus method with stub connected to the server. Similar to the server, the stub acts as a proxy here.

<Adder.java>

```
import java.rmi.*;
//define abstact method in the interface
public interface Adder extends Remote {


    public int add(int x, int y) throws RemoteException;
    public int minus(int x, int y) throws RemoteException;
}
```

<AdderRemote.java>

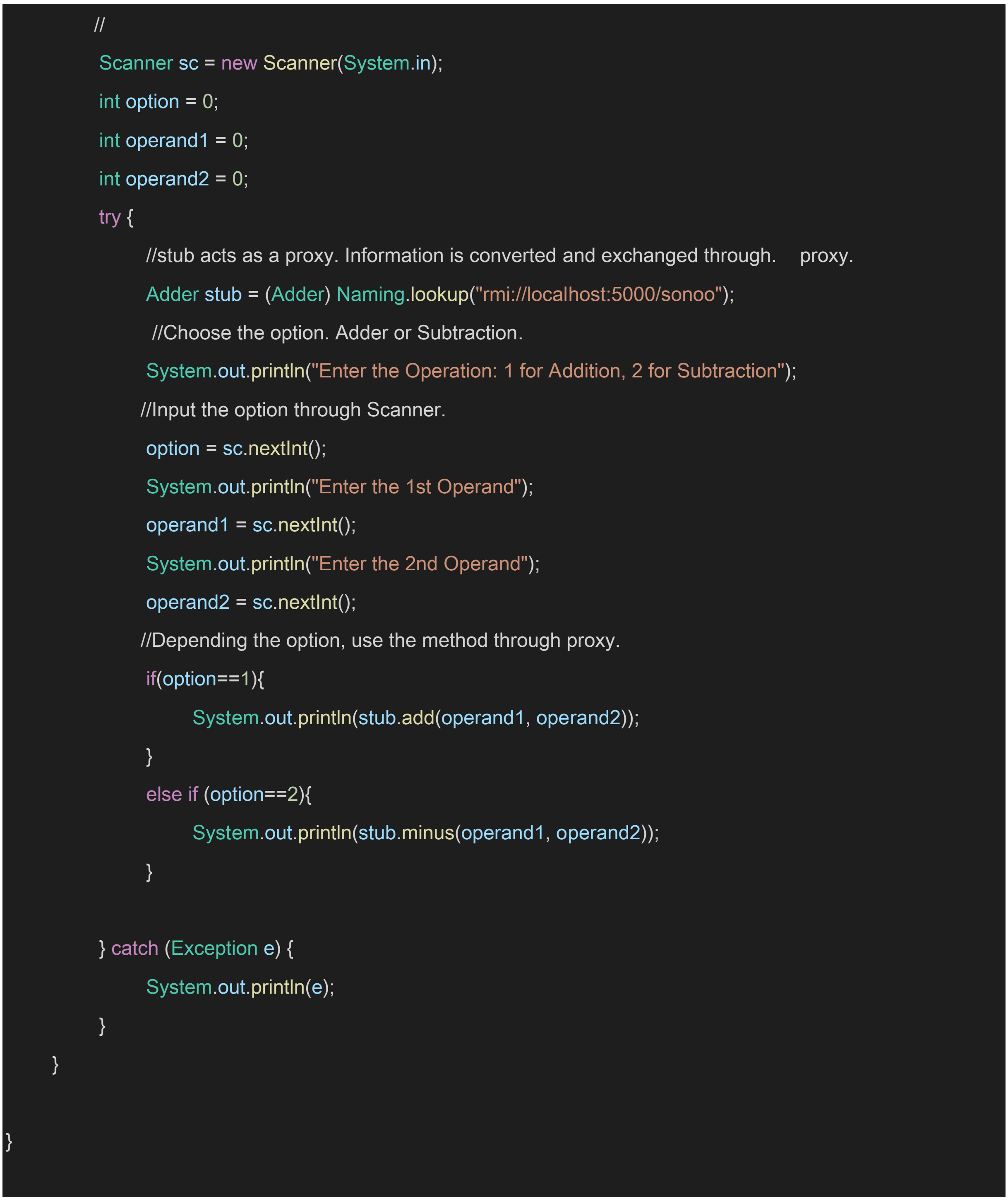
```
import java.rmi.*;
import java.rmi.server.*;


public class AdderRemote extends UnicastRemoteObject implements Adder {


    AdderRemote() throws RemoteException {
        super();
    }
    //this is implementation of method. Add method add two operands.
    public int add(int x, int y) {
        return x + y;
    }
    //minus method subtract two numbers.
    public int minus(int x, int y) {
        return x - y;
    }

}
```

<MyServer.java>

```
import java.rmi.*;
import java.rmi.registry.*;

public class MyServer {

    public static void main(String args[]) {
        try {
            //stub(or skeleton) acts as a proxy.
            //MyServer can generate AdderRemote object.
             Adder stub = new AdderRemote();
             Naming.rebind("rmi://localhost:5000/sonoo", stub);

        } catch (Exception e) {
             System.out.println(e);
        }
    }

}
```

<MyClient.java>

```
import java.rmi.*;
import java.util.Scanner;

public class MyClient {

    public static void main(String args[]) {
```



```
        //
        Scanner sc = new Scanner(System.in);
        int option = 0;
        int operand1 = 0;
        int operand2 = 0;
        try {
            //stub acts as a proxy. Information is converted and exchanged through.    proxy.
            Adder stub = (Adder) Naming.lookup("rmi://localhost:5000/sonoo");
             //Choose the option. Adder or Subtraction.
            System.out.println("Enter the Operation: 1 for Addition, 2 for Subtraction");
           //Input the option through Scanner.
            option = sc.nextInt();
            System.out.println("Enter the 1st Operand");
            operand1 = sc.nextInt();
            System.out.println("Enter the 2nd Operand");
            operand2 = sc.nextInt();
           //Depending the option, use the method through proxy.
            if(option==1){
                System.out.println(stub.add(operand1, operand2));
            }
            else if (option==2){
                System.out.println(stub.minus(operand1, operand2));
            }

        } catch (Exception e) {
            System.out.println(e);
        }
    }

}
```