

/\*\*\*\*\* Please read me before test \*\*\*\*\*/

### About the interface “Calculator”:

There is an addition parameter “int id” in every remote function, which is used for thread identify. And the addition function checkconnection() in “SingleStack” solution which is used for thread management. Based on different assumption, I got two different solutions for the Main Question (Single stack).

- i) In the folder of “SingleStack”: assume that there is only one client allow to use the common stack at one time, Client should send all the operators which need to be executed to the serve at one time.
- ii) In the folder of “SingleStack\_v2”: assume that there is only one client allow to use the common stack at one time, Client could send the operators separately to the serve.

The folder of “MultiStack” is for Bonus Question, each client has their own stack.

### About Text documents:

The size of “TestMultiClient.txt” are 30 operators\* 100

The size of “TestSingleClient.txt” are 100 operators \* 1

The form of text documents is shown as below:

Operators + correct answer

The last number is the correct answer for this calculation.

For instance:

5 6 + 11

Equal to  $5+6 = 11$ . Operators are “5 6 +” and correct answer is “11”.

1 2 + 4 – 2 + 2 - -1

Equal to  $1+2-4+2-2=-1$ . Operators are “1 2 + 4 – 2 + 2 -” and correct answer is “-1”. In this program, it compares the finally result which return from Server with the correct answer from test document in “CalculatorClient”. Print out “Test Passed” if there is no difference between two values. Otherwise, print out “Test Failed”

To test those three programs, please follow the following commands:

- a) Go to MultiStack, SingleStack or SingleStack\_v2 and Start the rmiregistry  
Start rmiregistry // Windows system

Or

rmiregistry & //Linux System

- b) Compile the documents `javac *.java`

- c) Start the Server.

`java CalculatorServer`

- d) Start the Client. If you want to test multiclient/SingleClient with normal pop(), please type:

`java CalculatorClient TestMultiClient.txt`

`java CalculatorClient TestSingleClient.txt`

If you want to test multiclient/SingleClient with delaypop(), please type:

`java CalculatorClient TestMultiClient.txt 1`

`java CalculatorClient TestSingleClient.txt 1`

The default delay time is set to 150ms. It's free to change it in following documents:

"CalculatorClient", Line 76	in MultiStack
"CalculatorImplementation", Line 51	in SingleStack
"CalculatorClient", Line 78	in SingleStack_v2

- ✓ It will return "Test Passed" if the calculation is correct. Otherwise, return "Test Failed"

- e) Open the rest of two folders, repeat the steps **b) – d)**