**Tutorial No. 2**

**Aim** :- Implement a client-server calculator program using multithreading.

**Theory** :-

Multithreading

Multithreading in java is a process of executing multiple threads simultaneously. Thread is basically a lightweight sub-process, a smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.But we use multithreading than multiprocessing because threads share a common memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.Java Multithreading is mostly used in games, animation etc.

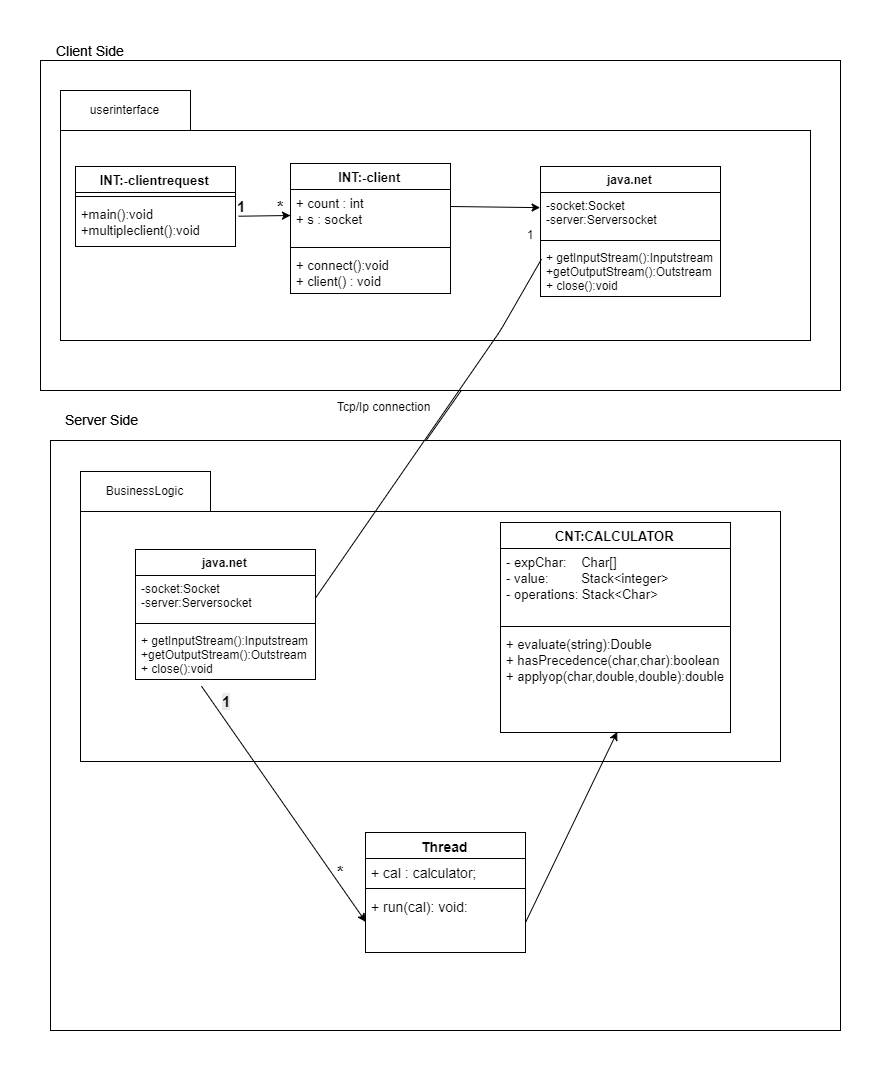
Advantages of Java Multithreading

* It doesn't block the user because threads are independent and you can perform multiple operations at same time.
* You can perform many operations together so it saves time.
* Threads are independent so it doesn't affect other threads if exception occur in a single thread.

We use multithreading in client-server architecture because we don’t want only a single client to connect to server at a particular time but many clients simultaneously. We want our architecture to **support multiple clients at the same time**. For this reason, we must use threads on server side so that whenever a client request comes, a separate thread can be assigned for handling each request.

With our basic server-client program, the request which comes even a nano-second first would be able to connect to the server and the other request would be rejected as no mechanism is provided for handling multiple requests simultaneously. To overcome this problem, we use threading in network programming.

**Class Diagram :-**

****

Testing :-

The following is a list of the items to be tested:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Combined

Some more modules were included and changed during development phase, which are not fully recognized or documented yet. So, test items of those modules will be included in the next version of the test plan.

**Features to be tested**

1. **Expression** 
   1. Test Id.: cal.0
   2. Description: To check correct functioning of the Calculate Module
   3. Pre-Condition: Calculator Ui must be loaded and Input must be provided
   4. Equivalence Classes Evaluate field:
      1. Length of expression :
         1. Length must be limited till 2^20
      2. Use of other function
         1. Expression containing function other than algebraic not allowed
         2. Expression containing characters as a input are not allowed
      3. Undefined in math
         1. Expression containing divide by 0 are not allowed as it is not defined in math
      4. Closing of brackets
         1. Expression must be with all closing of the brackets
      5. Use of two function without number in between
2. Expression must be with successive operation and numbers in between i.e use of two functions successively is not allowed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Input** | **Steps** | **Expected Output** | **Actual Output** | **Result** |
| 1 | Case 1  :- 4+9+85 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 94 | 94 | pass |
| 2 | Case 1  :- 20-10 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 10 | 10 | pass |
| 3 | Case 1  :- 5\*6 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 30 | 30 | Pass |
| 4 | Case 1  :- 6/3 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 2 | 2 | Pass |
| 5 | Case 2  :-9\*6-3 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 51 | 51 | Pass |
| 6 | Case 2  :- 9/3+2 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 5 | 5 | Pass |
| 7 | Case 2  :- 9\*4/12 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 3 | 3 | Pass |
| 8 | Case 3  :- (2+3)\*5 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 25 | 25 | Pass |
| 9 | Case 3:-  (6-4)/2 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 1 | 1 | Pass |
| 10 | Case 4:-  4\*5+6/3\*5 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 30 | 30 | Pass |
| 11 | Case:-2  (20\*3)/4-52\*(10+6/3) | 1)Compile CalUi.java file  2)Enter expression  3) run file | -609.0 | -609.0 | Pass |
| 12 | Case:-2  5.5/1.2\*3.1 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 14.2083 | 14.2083 | Pass |
| 13 | Case:-1  10/3 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 3.333 | 3.333 | Pass |
| 14 | Case:-  10\*2.3+9-1 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 31.0 | 31.0 | Pass |
| 15 | Case:-  (5.6/8)\*5-0.2 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 3.3 | 3.3 | Pass |
| 16 | Case:-  (5.5/1.2)\*3.1 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 14.2083 | 14.2083 | Pass |
| 17 | Case:-  15\*0.0 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 0.00 | 0.0 | Pass |
| 18 | Case:-  15/0 | 1)Compile CalUi.java file  2)Enter expression  3) run file | Infinite | Exception | Fail |
| 19 | Case:-  5.5/1.2\*3.1 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 14.2083 | 14.2083 | Pass |
| 20 | Case:-  (4+6)/5-(5\*1 | 1)Compile CalUi.java file  2)Enter expression  3) run file | -3.0 | Error Display for  Not closing bracket | Fail |
| 21 | Case:-  8\*2+/3 | 1)Compile CalUi.java file  2)Enter expression  3) run file | 19 or  5.6 | Error Display for  Using two brackets at successive | Fail |
| 22 | Case:-  80/77-(73\*96+89)\*3-56 | 1)Compile CalUi.java file  2)Enter expression  3) run file | -21345.9610 | -21345.9610Display for  Not closing bracket | Pass |

**Conclusion**:- Thus we have studied the use of multithreading in client server architecture using our calculator program. For each client a server side thread will be created which implements the program and returns the user to the client.