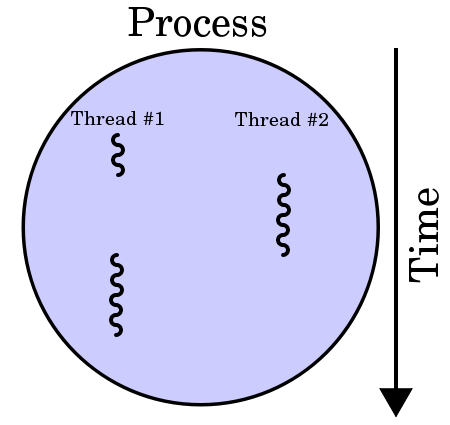
**TUTORIAL NO: 2**

**Aim:-**

To implement calculator program using multithreaded server. Each user will be served by different thread, at the end of the calculations server will respond with all the calculations done by the specific user.

**Theory:-**

In computer architecture, multithreading is the ability of a central processing unit or a single core in a multi-core processor to execute multiple processes or threads concurrently, appropriately supported by the operating system. This approach differs from multiprocessing, as with multithreading the processes and threads share the resources of a single or multiple cores: the computing units, the CPU caches, and the translation lookaside buffer.

Multithreading in Java:-

Java is a multi-threaded programming language which means we can develop multi-threaded program using Java. A multi-threaded program contains two or more parts that can run concurrently and each part can handle a different task at the same time making optimal use of the available resources specially when your computer has multiple CPUs.

By definition, multitasking is when multiple processes share common processing resources such as a CPU. Multi-threading extends the idea of multitasking into applications where you can subdivide specific operations within a single application into individual threads. Each of the threads can run in parallel. The OS divides processing time not only among different applications, but also among each thread within an application.

Multi-threading enables you to write in a way where multiple activities can proceed concurrently in the same program.

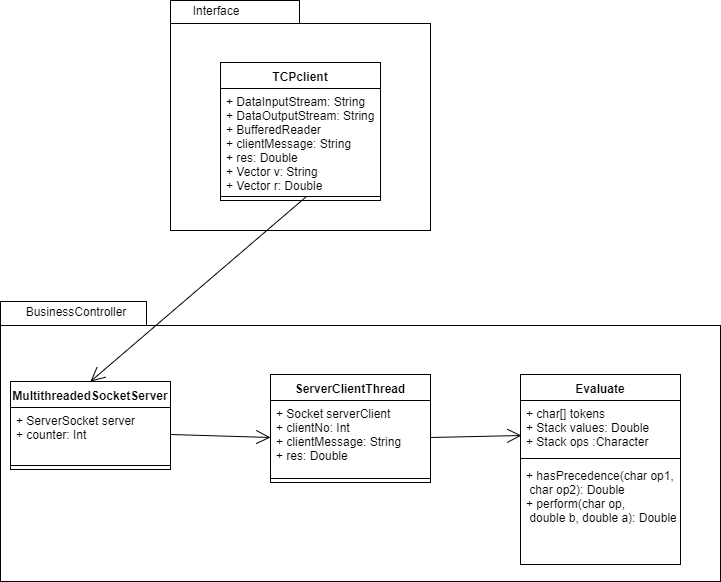
Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.  
Threads can be created by using two mechanisms:  
1. Extending the Thread class  
2. Implementing the Runnable Interface

**Thread creation by extending the Thread class:-**  
We create a class that extends the **java.lang.Thread** class. This class overrides the run() method available in the Thread class. A thread begins its life inside run() method. We create an object of our new class and call start() method to start the execution of a thread, start() invokes the run() method on the Thread object.

If we extend the Thread class, our class cannot extend any other class because Java doesn’t support multiple inheritance.

We can achieve basic functionality of a thread by extending Thread class because it provides some inbuilt methods like yield() , interrupt() etc.

**Class Diagram:-**

****

**Conclusion:-**

Class Evaluate contains all the methods required for calculator operations. On the interface side we have class TCPclient which connects to the MultithreadedSocketServer class. Every time a new client connects a new thread of class ServerClientThread is created. Thus we have implemented the calculator using multithreading concepts in Java.