



Artificial Intelligence and Data Science Department.

OOPM / Odd Sem 2021-22 / Experiment 13.

YASH SARANG.

47 / D6AD.

EXPERIMENT - 13.

Experiment - 13

056

Aim: Program of Multithreading. To write a program to print the table of 5, T, B using Multithreading.

To write a program to print first 20 prime no.s and 15 fibonacci series. (Using runnable interface)
Program on concept of synchronization.

Theory: Multithreading in Java is a process of executing multiple threads simultaneously. A thread is a lightweight sub-process, the smallest unit of process. Multiprocessing and multithreading, both are used to achieve multitasking.

Advantages of Java Multithreading:

- ① It doesn't block the user because threads are independent and you can perform multiple operations at the same time.
- ② You can perform many operations together, so it saves time.
- ③ Threads are independent, so it doesn't affect other threads if an exception occurs in a single thread.

Thread-based Multitasking

- Threads share the same address space.
- A thread is lightweight.
- Cost of communication between thread is low.

A thread is a lightweight subprocesses, the smallest unit of processing. It is a separate path of execution. Threads are independent. If there occurs exception in one thread, it doesn't affect other threads. It uses a shared memory area. There is context-switching between the threads. There can be multiple processes inside the OS, and process can have multiple threads.

Program Code:

```
import java.lang.Runnable;  
import java.util.Scanner;
```

```
class MultiThreading  
{
```

```
    public static void main(String args[])  
    {  
        System.out.println("Hello");
```

```
        Prime a = new Prime();  
        Fibonnaci b = new Fibonnaci();  
        Thread c = new Thread(b);
```

```
        //start Prime thread  
        a.start();  
        try  
        {  
            //wait for prime a to stop execution
```

```

a.join();
}
catch( InterruptedException e)
{
    System.out.println("ERROR : " + e);
}
//start Fibonacci thread
c.start();
}

```

```

}

```

```

class Prime extends Thread

```

```

{
    public void run()
    {
        System.out.println("In Prime class");
        Scanner in = new Scanner(System.in);
        System.out.print("Enter Count : ");
        int count = in.nextInt();
    }
}

```

```

int status = 1;

```

```

int num = 3;

```

```

if (count >= 1)

```

```

{
    System.out.println("First "+count+" prime numbers are:");
    System.out.println(2);
}

```

```

for ( int i = 2 ; i <=count ; )

```

```

{
    for ( int j = 2 ; j <= Math.sqrt(num) ; j++ )
    {
        if ( num%j == 0 )
        {
            status = 0;
            break;
        }
    }
}
if ( status != 0 )

```

```

{
    System.out.println(num);
    i++;
}
status = 1;
num++;

}
    System.out.println("Prime thread finished! ");
}
}

```

class Fibonnaci implements Runnable

```

{
    public void run()
    {
        System.out.println("In Fibonnaci class");
        int a,b,c;
        a = 0; b = 1;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter Count : ");
        int count = in.nextInt();
        if( count == 1)
        {
            System.out.println("FIBO SERIES : " + a);
            return;
        }
        else
        {
            System.out.println("FIBO SERIES : " + a);
            System.out.println("FIBO SERIES : " + b);
            for(int j = 0; j<count; j++)
            {
                c = a+b;
                System.out.println("FIBO SERIES : " + c);
                a = b;
                b = c;
            }
        }

        System.out.println("Fibonacci Thread finished!");
    }
}

```

}

}

Output:

Hello

In Prime class

Enter Count : 12

First 12 prime numbers are:

2

3

5

7

11

13

17

19

23

29

31

37

Prime thread finished!

In Fibonnaci class

Enter Count :

Program Code:

```
class javasync
```

```
{
```

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

```
{
```

```
    Table obj = new Table();
```

```
    Tableof5 a = new Tableof5(obj);
```

```
    Tableof10 b = new Tableof10(obj);
```

```
    a.start();
```

```
    b.start();
```

```
}
```

```
}
```

```

class Table
{
    synchronized void printTable(int n)
    {
        for(int i = 1; i<=5 ; i++)
        {
            System.out.println(n*i);
        }

        try
        {
            Thread.sleep(500);
        }
        catch(Exception e)
        {
            System.out.println("ERROR : "+e);
        }
    }
}

```

```

class Tableof5 extends Thread
{
    Table t;
    Tableof5(Table t)
    {
        this.t = t;
    }

    public void run()
    {
        t.printTable(5);
    }
}

```

```

class Tableof10 extends Thread
{
    Table t;
    Tableof10(Table t)
    {
        this.t = t;
    }
}

```

```
    public void run()
    {

        t.printTable(10);

    }
}
```

Output:

```
C:\>set path=C:\Program Files\Java\jdk-16.0.2\bin
C:\>cd A00PM
The system cannot find the path specified.
C:\>cd 00PM
C:\00PM>javasync.java
C:\00PM>javac javasync.java
C:\00PM>java javasync
5
10
15
20
25
10
20
30
40
50
C:\00PM>
```
