DESIGN ANALYSIS AND ALGORITHM

PRACTICAL NO 9

027_Abhishek_Ojha

Experiment No - 9 Date of Experiment :- 12 October 2021

Program :- Write a program to implement multi threaded computation concepts in the generation of Fibonacci numbers.

Algorithm

```
Step 1:   Start
Step 2:   create class prime implements Runnable
Step 3:   create class fib implements Runnable
Step 4:   Thread ct=Thread.currentThread()
Step 5:   print ct.getName()
Step 6:   prime p=new prime()
Step 7:   fib f=new fib()
Step 8:   Thread fib=new Thread(f,"fibo")
Step 9:   Thread prime=new prime(p,"prime")
Step 10:  fib.start()
Step 11:  print fib.getName()
Step 12:  prime.start()
Step 13:  print prime.getName()
Step 14:  stop
```

Function public void run() in class prime

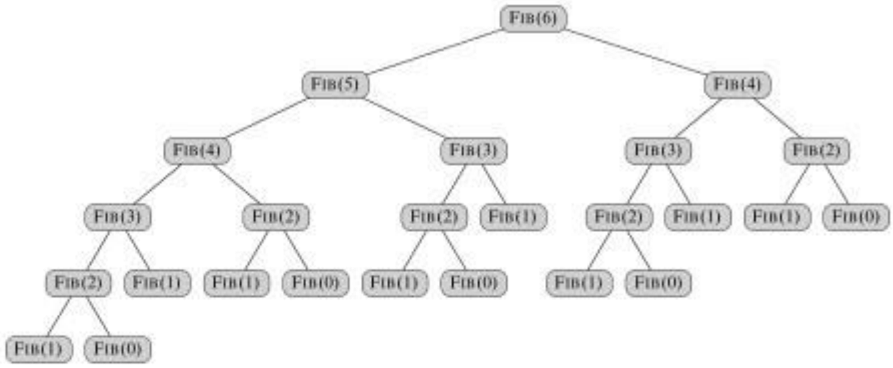
```
Step 1:   Start
Step 2:   Set i=0
Step 3:   Repeat steps 4 to 8 until i<=100 incrementing i by 1
Step 4:   Set j=2
Step 5: Repeat steps 5 to 6 until j<=i incrementing j by 1

Step 6:   if(i%j==0) go to 6
Step 7:     if(i==j) go to 7
Step 8:     increment c by 1
Step 9:     print the value of i
Step 10:    exit
```

Function public void run() of class fib

```
Step 1:     Start
Step 2:     Set n=0,a=0,b=1 and c=0
Step 3:     Repeat steps 4 to 7 until n<75 incrementing n by 1
Step 4:     print the value of a
Step 5:     Compute c=a+b,assign a=b and b=c
Step 6:     if(n==50) go to 7
Step 7:     Thread.sleep(500)
Step 8: Exit.
```

Fig :-



Practical Implementation of Huffman's code algorithm

```
class Prime implements Runnable
{
        long j,c;
        Prime()
        {
```

```
        super(); c=0;
    }
     public void run()
    {
        for(long i=0;i<=100;i++)
        { for(j=2;j<=i;j++)
            {
                if(i%j==0) break;
            } if(j==i)
            {
                c++;
                    System.out.println(c+"th" +" Prime no: = "+i);
            }
        }
    }
}

class Fib implements Runnable
{
    long a,b,c,n;
    Fib()
    {
        a=c=n=0; b=1;
    }
    public void run()
    {
        while(n++<75)
        {
            System.out.println(n+"th" +" Fib no: = "+a);
            c=a+b; a=b; b=c; try
            {
                if(n==50)
                {
System.out.println("Thread fibonacci is put into sleep.");
                     Thread.sleep(500);
```

```
                                    }

                              }
                              catch(InterruptedException e)
                              {
                                      System.out.println("Error : " + e);
                              }
                        }
                  }
            }

            public class MyPriFib {

                  public static void main(String[] args) {
                        Thread ct=Thread.currentThread();
                        System.out.println("Main thread name : "+ct.getName());
                        Prime p=new Prime();
                        Fib f=new Fib();
                        Thread fib=new Thread(f,"fibonacci"); Thread
                        prime=new Thread(p,"prime"); fib.start();
                        System.out.println("Thread "+ fib.getName() + " started.");
                        prime.start();
                        System.out.println("Thread "+ prime.getName() + " started.");

                  }
            }
```

Output :

```
Main thread name : main Thread fibonacci started.

1th Fib no: = 0

2th Fib no: = 1 Thread prime

started.

3th Fib no: = 1

4th Fib no: = 2
```

```
5th Fib no: = 3

1th Prime no: = 2

6th Fib no: = 5

2th Prime no: = 3

7th Fib no: = 8

3th Prime no: = 5

8th Fib no: = 13

4th Prime no: = 7

9th Fib no: = 21

5th Prime no: = 11

10th Fib no: = 34

6th Prime no: = 13

11th Fib no: = 55

7th Prime no: = 17

12th Fib no: = 89

8th Prime no: = 19

13th Fib no: = 144

9th Prime no: = 23

14th Fib no: = 233

10th Prime no: = 29

15th Fib no: = 377

11th Prime no: = 31

16th Fib no: = 610

12th Prime no: = 37

17th Fib no: = 987

13th Prime no: = 41

18th Fib no: = 1597
```

```
14th Prime no: = 43
19th Fib no: = 2584
15th Prime no: = 47
20th Fib no: = 4181
16th Prime no: = 53
21th Fib no: = 6765
17th Prime no: = 59
22th Fib no: = 10946
18th Prime no: = 61
23th Fib no: = 17711
19th Prime no: = 67
24th Fib no: = 28657
20th Prime no: = 71
25th Fib no: = 46368
21th Prime no: = 73
26th Fib no: = 75025
22th Prime no: = 79
27th Fib no: = 121393
23th Prime no: = 83
28th Fib no: = 196418
24th Prime no: = 89
29th Fib no: = 317811
25th Prime no: = 97 30th Fib no: = 514229
31th Fib no: = 832040
32th Fib no: = 1346269
33th Fib no: = 2178309
34th Fib no: = 3524578
```

```
35th Fib no: = 5702887

36th Fib no: = 9227465

37th Fib no: = 14930352

38th Fib no: = 24157817

39th Fib no: = 39088169

40th Fib no: = 63245986

41th Fib no: = 102334155

42th Fib no: = 165580141

43th Fib no: = 267914296

44th Fib no: = 433494437

45th Fib no: = 701408733

46th Fib no: = 1134903170

47th Fib no: = 1836311903

48th Fib no: = 2971215073

49th Fib no: = 4807526976

50th Fib no: = 7778742049 Thread fibonacci is

put into sleep.

51th Fib no: = 12586269025

52th Fib no: = 20365011074

53th Fib no: = 32951280099

54th Fib no: = 53316291173

55th Fib no: = 86267571272

56th Fib no: = 139583862445

57th Fib no: = 225851433717

58th Fib no: = 365435296162

59th Fib no: = 591286729879
```

```
60th Fib no: = 956722026041

61th Fib no: = 1548008755920

62th Fib no: = 2504730781961

63th Fib no: = 4052739537881

64th Fib no: = 6557470319842

65th Fib no: = 10610209857723

66th Fib no: = 17167680177565

67th Fib no: = 27777890035288

68th Fib no: = 44945570212853

69th Fib no: = 72723460248141

70th Fib no: = 117669030460994

71th Fib no: = 190392490709135

72th Fib no: = 308061521170129

73th Fib no: = 498454011879264

74th Fib no: = 806515533049393

75th Fib no: = 1304969544928657
```