**Name: Akhila Anand**

**Roll No:3**

**Batch:S2 RMCA**

**Date:07\06\2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 20**

**Aim**

Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads.

**Source code:**

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();

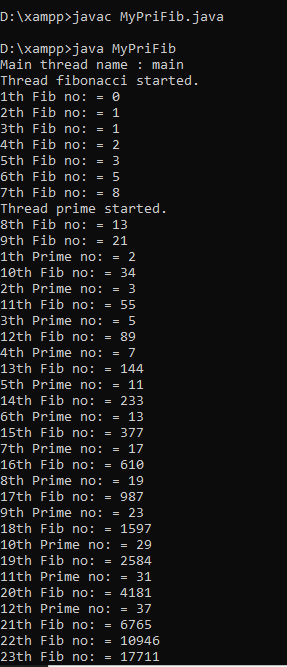
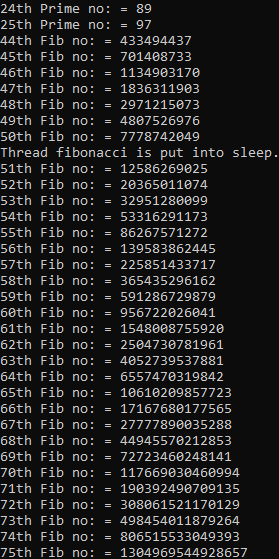
System.out.println("Thread "+ fib.getName() + " started.");

prime.start();

System.out.println("Thread "+ prime.getName() + " started.");

}

}

**Output Screenshot:**