

Java 多线程作业

2013599 田佳业

一、设计目标：

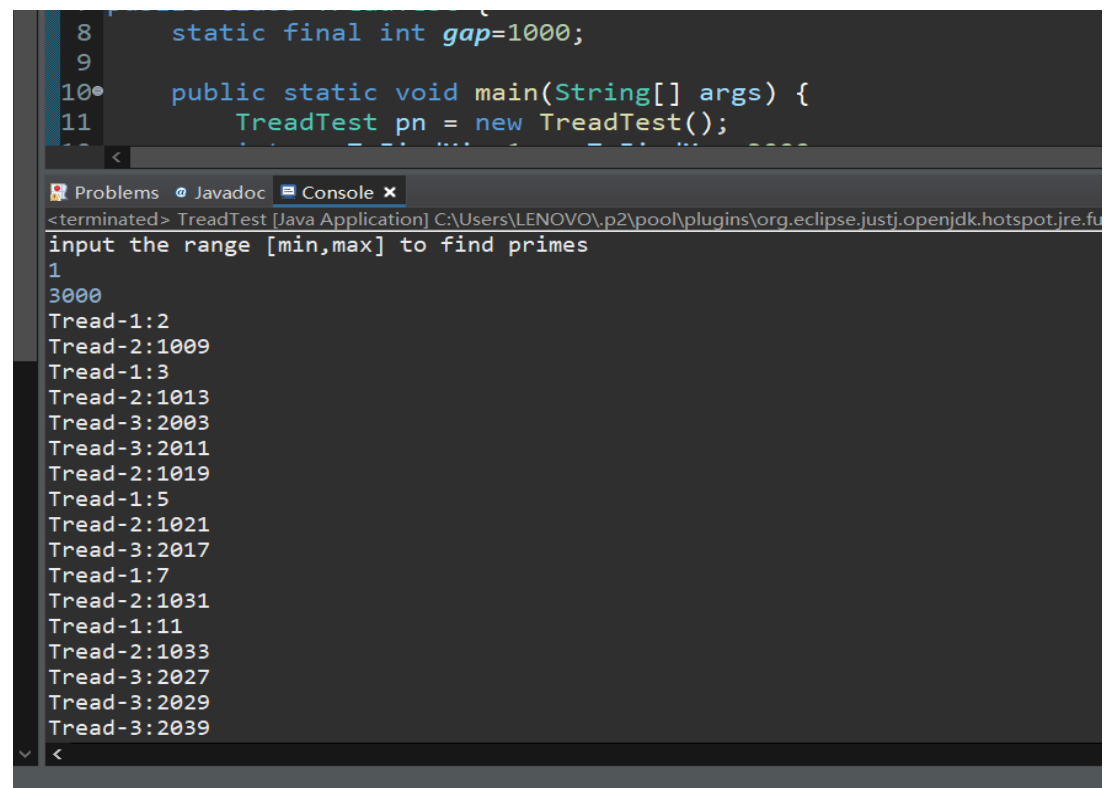
编写类 ThreadTest，实现利用多线程求解某范围素数，每个线程负责 1000 范围：线程 1 找 1-1000；线程 2 找 1001-2000；线程 3 找 2001-3000。编写程序，将每个线程找到的素数及时打印。输出格式要求：

Thread-0:3

Thread-1:1003

Thread-2:2003

二、运行实例：



The screenshot shows the Eclipse IDE with a Java file named ThreadTest.java. The code defines a static final int gap=1000; and a main method that creates a ThreadTest object. The console output shows the program's execution, including the prompt 'input the range [min,max] to find primes' and the output of the three threads: Thread-1:2, Thread-2:1009, Thread-1:3, Thread-2:1013, Thread-3:2003, Thread-3:2011, Thread-2:1019, Thread-1:5, Thread-2:1021, Thread-3:2017, Thread-1:7, Thread-2:1031, Thread-1:11, Thread-2:1033, Thread-3:2027, Thread-3:2029, and Thread-3:2039.

```
8     static final int gap=1000;
9
10    public static void main(String[] args) {
11        ThreadTest pn = new ThreadTest();
12    }
```

Problems Javadoc Console x

<terminated> ThreadTest [Java Application] C:\Users\LENOVO\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu

input the range [min,max] to find primes

1

3000

Thread-1:2

Thread-2:1009

Thread-1:3

Thread-2:1013

Thread-3:2003

Thread-3:2011

Thread-2:1019

Thread-1:5

Thread-2:1021

Thread-3:2017

Thread-1:7

Thread-2:1031

Thread-1:11

Thread-2:1033

Thread-3:2027

Thread-3:2029

Thread-3:2039

```
11 TreadTest pn = new TreadTest();
12 ...

Problems Javadoc Console x
<terminated> TreadTest [Java Application] C:\Users\LENOVO\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full...
input the range [min,max] to find primes
1
4000
Tread-1:2
Tread-1:3
Tread-1:5
Tread-1:7
Tread-1:11
Tread-1:13
Tread-2:1009
Tread-4:3001
Tread-1:17
Tread-3:2003
Tread-2:1013
Tread-2:1019
Tread-2:1021
Tread-3:2011
Tread-2:1031
Tread-2:1033
Tread-4:3011
```

三、核心代码：

```
package findPrimeNumber;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class TreadTest {
    static final int gap=1000;

    public static void main(String[] args) {
        TreadTest pn = new TreadTest();
        //default value
        int numToFindMin=1,numToFindMax=3000;
        System.out.println("input the range [min,max] to find
primes");
        BufferedReader in=new BufferedReader(new
InputStreamReader(System.in));
        try {
            numToFindMin=Integer.parseInt(in.readLine());
            numToFindMax=Integer.parseInt(in.readLine());
        } catch (NumberFormatException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

```

    }
    int treadNum=(numToFindMax-numToFindMin+1)/1000;
    for (int i = 0; i < treadNum; i++) {
        pn.new
PrimeFinderTread(numToFindMin+i*gap,numToFindMin+(i+1)*gap,i+1).s
tart();
    };
}
class PrimeFinderTread extends Thread {
    private int start;
    private int end;
    private int treadName;
    public PrimeFinderTread(int start, int end,int treadName)
{
        this.start = start;
        this.end = end;
        this.treadName=treadName;
    }
    public void run() {
        for (int i = start; i < end; i++) {
            synchronized (this) {

                if (isPrime(i)) {
                    System.out.println("Tread-
"+treadName+": "+i);
                }
            }
        }
    }
    private boolean isPrime(int i) {
        if(i==1)
            return false;
        else if(i==2)
            return true;
        else {
            for (int j = 2; j <= Math.ceil(Math.sqrt(i)); j++)
            {
                if (i % j == 0) {
                    return false;
                }
            }
            return true;
        }
    }
}

```

```
    }  
  }  
}
```