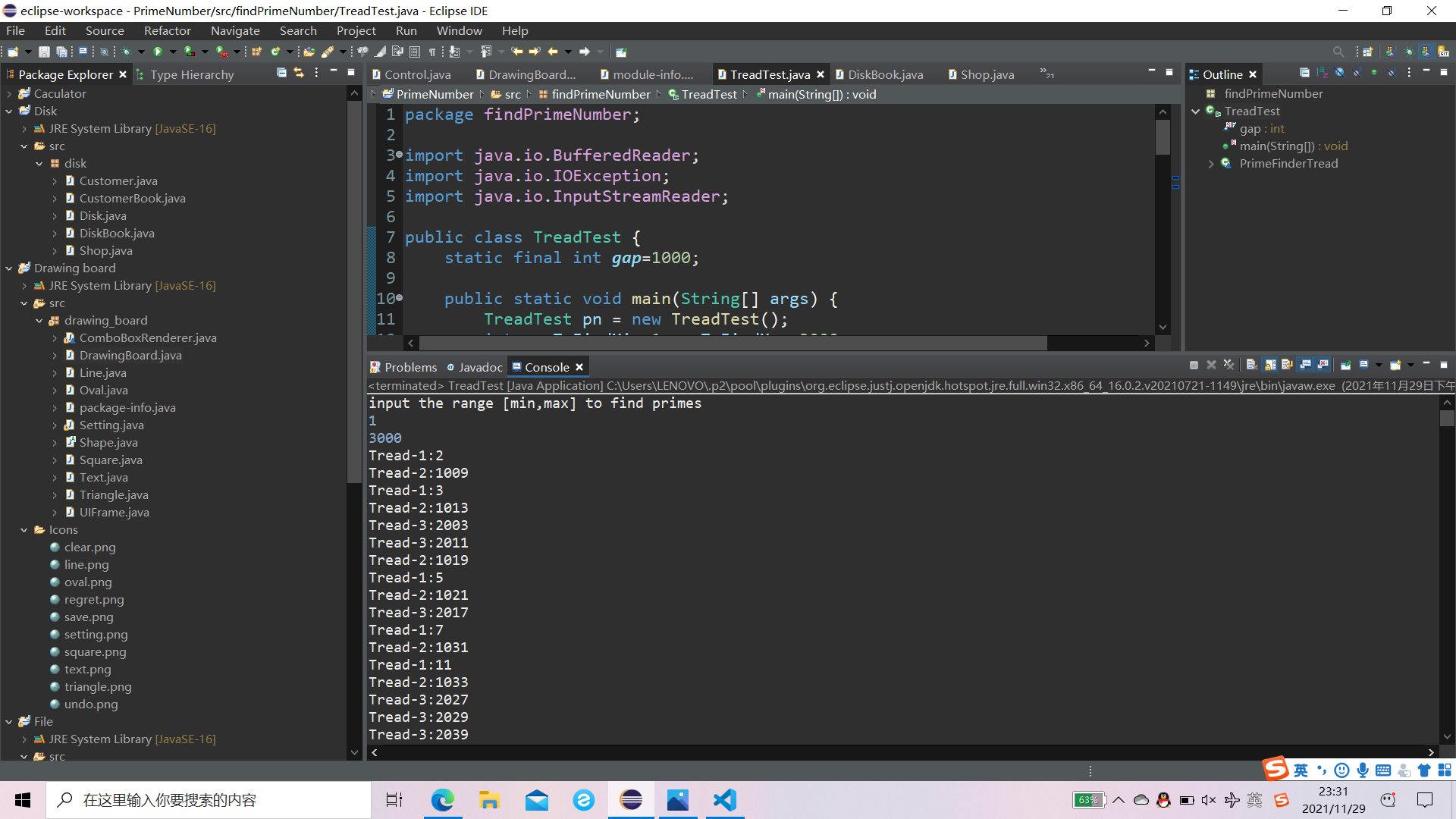
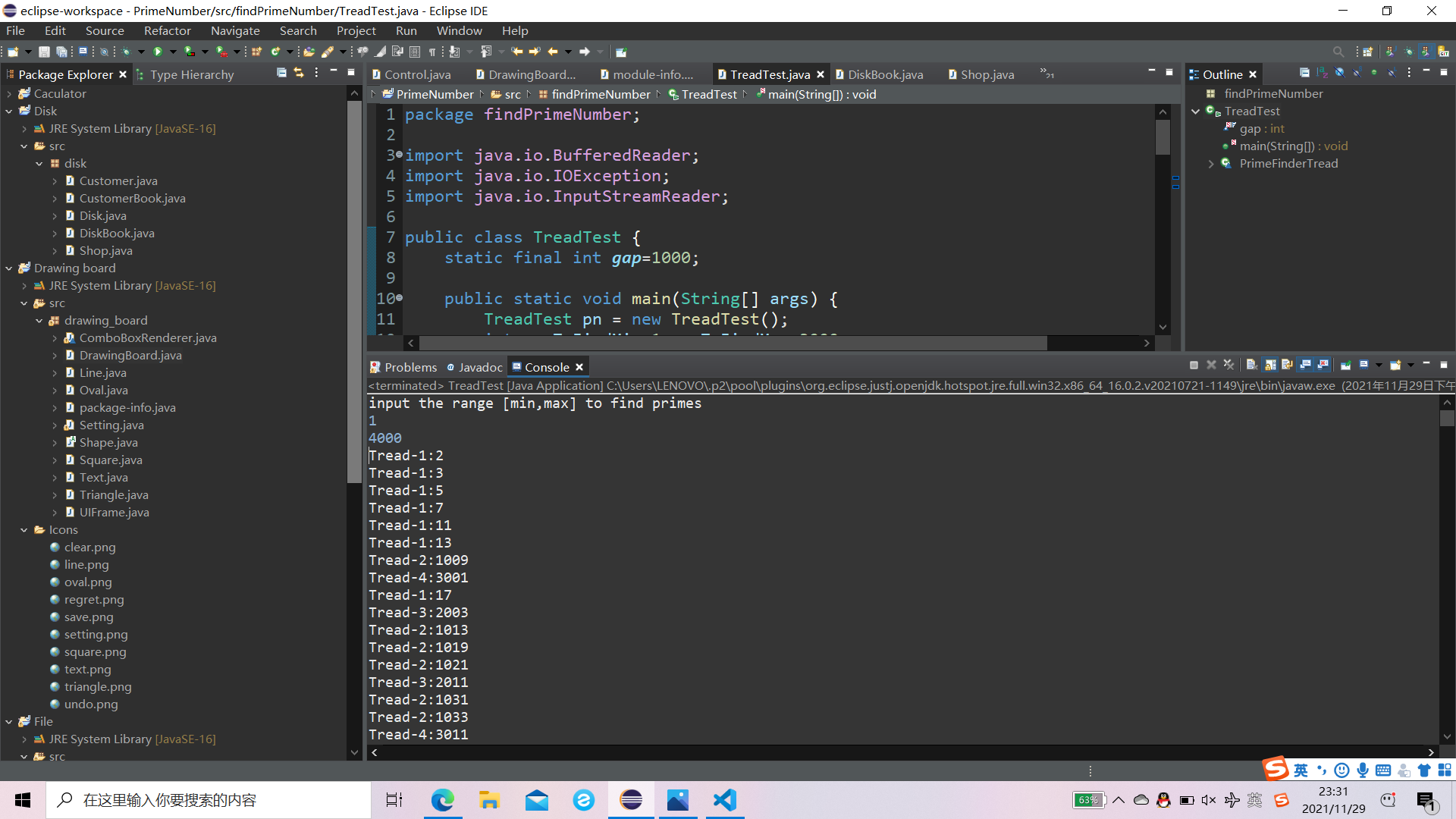
Java多线程作业

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:  




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).start();

        };

    }

    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;

            }

        }

    }

}