# Test

// instance.test1(2,MAX,4); // T1 hat kleine Zahlen t4 große zahlen

//instance.test2(2,MAX,4); // step=4 t1 hat step 2 t3 hat step 4 schnell

//instance.test3(2,MAX,4); // step=8 t1:3 t2:5 t3:7 gleich schnell

import java.util.concurrent.Semaphore;

**public void test0(int von, int bis) {**

int summe = 0;

for (int i = von; i <= bis; i++) {

boolean isPrimeNumber = true;

for (int j = 2; j < i; j++) {

if (i % j == 0) {

// j ist ein teiler von i

isPrimeNumber = false;

break;

}

} // for

if (isPrimeNumber) {

summe++;

}

} // for

System.out.println("Test0 Summe: " + summe); // 78498

} // test0

**public void test1(int von, int bis, int anzThreads) {**

MyThread1[] feld = new MyThread1[anzThreads];

int diff = bis/anzThreads;

int von1=von;

int bis1=diff; // for loop <bis1

for (int i=0; i<feld.length; i++) {

feld[i] = new MyThread1(i+1,von1,bis1);

von1 = bis1;

if (i==feld.length-2) {

bis1 = bis+1;

}

else {

bis1 = von1+diff;

}

}

for(MyThread1 t : feld) {

t.start();

}

try {

for (MyThread1 t : feld) {

t.join();

}

}

catch (InterruptedException e) {

}

int summe=0;

for(MyThread1 t : feld) {

summe+=t.getSumme();

}

System.out.println("Test1 summe: "+summe);

} // test1

**public void test2(int von, int bis, int anzThreads) {**

MyThread2[] feld = new MyThread2[anzThreads];

int step = anzThreads;

for (int i=0; i<feld.length; i++) {

feld[i] = new MyThread2(i+1,von+i, bis, step);

}

for(MyThread2 t : feld) {

t.start();

}

try {

for (MyThread2 t : feld) {

t.join();

}

}

catch (InterruptedException e) {

}

int summe=0;

for(MyThread2 t : feld) {

summe+=t.getSumme();

}

System.out.println("Test2 summe: "+summe);

} // test2

**public void test3(int von, int bis, int anzThreads) {**

MyThread2[] feld = new MyThread2[anzThreads];

int step = anzThreads<<1;

int start=0;

if (von%2==0) {

start=von+1;

}

else {

start=von;

}

for (int i=0; i<feld.length; i++) {

feld[i] = new MyThread2(i+1,start+(i+i), bis, step);

}

for(MyThread2 t : feld) {

t.start();

}

try {

for (MyThread2 t : feld) {

t.join();

}

}

catch (InterruptedException e) {

}

int summe=0;

for(MyThread2 t : feld) {

summe+=t.getSumme();

}

System.out.println("Test3 summe: "+summe);

} // test3

**public void test4(int von, int bis, int anzThreads) {**

MyThread4[] feld = new MyThread4[anzThreads];

int step = anzThreads<<1;

int start=0;

if (von%2==0) {

start=von+1;

}

else {

start=von;

}

Result result = new Result();

for (int i=0; i<feld.length; i++) {

feld[i] = new MyThread4(i+1,start+(i+i), bis, step, result);

}

for(MyThread4 t : feld) {

t.start();

}

try {

for (MyThread4 t : feld) {

t.join();

}

}

catch (InterruptedException e) {

}

System.out.println("Test4 ");

result.print();

} // test4

**public void test5(int von, int bis, int anzThreads) {**

MyThread5[] feld = new MyThread5[anzThreads];

int step = anzThreads<<1;

int start=0;

if (von%2==0) {

start=von+1;

}

else {

start=von;

}

Result5 result = new Result5();

result.crit = new Semaphore(1);

for (int i=0; i<feld.length; i++) {

feld[i] = new MyThread5(i+1,start+(i+i), bis, step, result);

}

for(MyThread5 t : feld) {

t.start();

}

try {

for (MyThread5 t : feld) {

t.join();

}

}

catch (InterruptedException e) {

}

System.out.println("Test5 ");

result.print();

} // test5

}

# Result

import java.util.ArrayList;

public class Result {

private ArrayList<Integer> liste = new ArrayList<>(100000);

public void addNumber(int number) {

liste.add(number);

}

public void print(){

System.out.println("summe: "+liste.size() );

}

}

# public class MyThread1 extends Thread{

private int id=0;

private int von=0;

private int bis=0;

private int summe=0;

**public MyThread1(int id, int von, int bis) {**

this.id=id;

this.von=von;

this.bis=bis;

System.out.println("id: "+id+" von: "+von+" bis:"+bis);

}

**public int getSumme() {**

return summe;

}

**public void run() {**

summe = 0;

for (int i = von; i < bis; i++) {

boolean isPrimeNumber = true;

for (int j = 2; j < i; j++) {

if (i % j == 0) {

// j ist ein teiler von i

isPrimeNumber = false;

break;

}

} // for

if (isPrimeNumber) {

summe++;

}

} // for

System.out.println("Thread "+id+" nun beendet");

}

}

# public class MyThread2 extends Thread{

private int id=0;

private int von=0;

private int bis=0;

private int step=0;

private int summe=0;

**public MyThread2(int id, int von, int bis, int step) {**

this.id=id;

this.von=von;

this.bis=bis;

this.step = step; System.out.println("id: "+id+" von: "+von+ " bis:"+bis+" step"+step);

}

**public int getSumme() {**

return summe;

}

**public void run() {**

summe = 0;

for (int i = von; i <=bis; i+=step) {

boolean isPrimeNumber = true;

for (int j = 2; j < i; j++) {

if (i % j == 0) {

// j ist ein teiler von i

isPrimeNumber = false;

break;

}

} // for

if (isPrimeNumber) {

summe++;

}

} // for

System.out.println("Thread "+id+" nun beendet");

}

}

## public class MyThread4 extends Thread{

private int id=0;

private int von=0;

private int bis=0;

private int step=0;

private Result result=null;

**public MyThread4(int id, int von, int bis, int step, Result result) {**

this.id=id;

this.von=von;

this.bis=bis;

this.step = step;

this.result = result;

System.out.println("id: "+id+" von: "+von+

" bis:"+bis+" step"+step);

}

**public void run() {**

for (int i = von; i <=bis; i+=step) {

boolean isPrimeNumber = true;

for (int j = 2; j < i; j++) {

if (i % j == 0) {

// j ist ein teiler von i

isPrimeNumber = false;

break;

}

} // for

if (isPrimeNumber) {

result.addNumber(i);

}

} // for

System.out.println("Thread "+id+" nun beendet");

}

}

import java.util.ArrayList;

import java.util.concurrent.Semaphore;

# public class Result5 {

int summe =0;

public Semaphore crit = null;

public void addNumber(int number) {

summe++;

}

public void print(){

System.out.println("summe: "+summe );

}

}

import java.util.concurrent.Semaphore;

# public class MyThread5 extends Thread{

private int id=0;

private int von=0;

private int bis=0;

private int step=0;

private Result5 result=null;

**public MyThread5(int id, int von, int bis, int step, Result5 result) {**

this.id=id;

this.von=von;

this.bis=bis;

this.step = step;

this.result = result;

System.out.println("id: "+id+" von: "+von+

" bis:"+bis+" step"+step);

}

**public void run() {**

for (int i = von; i <=bis; i+=step) {

boolean isPrimeNumber = true;

for (int j = 2; j < i; j++) {

if (i % j == 0) {

// j ist ein teiler von i

isPrimeNumber = false;

break;

}

} // for

if (isPrimeNumber) {

try {

//System.out.println("vor down");

result.crit.acquire(); // Down Dijstra

//System.out.println("vor summe");

result.addNumber(i);

result.crit.release(); // Up

//System.out.println("nach release");

}

catch (InterruptedException e) {

}

}

} // for

System.out.println("Thread "+id+" nun beendet");

}

}