Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Основи паралельного програмування

Лабораторна робота №2

**«Потоки в мові Java»**

Виконав:

студент групи ІО-64

Мельничук В.І.

Перевірив:

Корочкін О. В.

Київ

2018 р.

**Варіант 14**

**Завдання:**

F1: D = (Sort(A+B) + C ) \* (MA\*ME)

F2: ML = Sort(MF + MG\*MH)

F3:T = (O+P) \* (ML\*MS)

**Лістинг програми:**

**Main.java**

*// Основи паралельного програмування*

*// Лабораторна робота №2*

*// Потоки в мові Java*

*// F1: D = (Sort(A+B) + C ) \* (MA\*ME)*

*// F2: ML = Sort(MF + MG\*MH)*

*// F3: T = (O+P) \* (ML\*MS)*

*// Мельничук Віталій ІО-64*

***// 06.10.2018***

***import*** *java.util.Scanner;*

**public class** Main {

**public static void** main(String[] args) {

System.***out***.println(**"Enter n: "**);

Scanner scanner = **new** Scanner(System.***in***);

**int** N = scanner.nextInt();

Operations.*N* = N;

Thread T1 = **new** Thread(**null**, **new** F1(N), **"F1"**, 8000); *//F1: D = (Sort(A+B) + C) \* (MA\*ME)*

Thread T2 = **new** Thread(**null**, **new** F2(N), **"F2"**, 8000); *//F2: ML = Sort(MF + MG\*MH)*

Thread T3 = **new** Thread(**null**, **new** F3(N), **"F3"**, 8000); *//F3: T = (O+P) \* (ML\*MS)*

T1.setPriority(5);

T2.setPriority(7);

T3.setPriority(1);

T1.start();

T2.start();

T3.start();

**try** {

T1.join();

T2.join();

T3.join();

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

**F1.java**

*// Основи паралельного програмування*

*// Лабораторна робота №2*

*// Потоки в мові Java*

*// F1: D = (Sort(A+B) + C ) \* (MA\*ME)*

*// F2: ML = Sort(MF + MG\*MH)*

*// F3: T = (O+P) \* (ML\*MS)*

*// Мельничук Віталій ІО-64*

***// 06.10.2018***

**public class** F1 **implements** Runnable{

**private int N**;

**public** F1(**int** n) {

**super**();

**N** = n;

}

**public void** run() {

System.***out***.println(**"F1 started"**);

**try** {

Thread.*sleep*(2000);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

Vector A = **new** Vector(**N**);

Vector B = **new** Vector(**N**);

Vector C = **new** Vector(**N**);

Matrix MA = **new** Matrix(**N**);

Matrix ME = **new** Matrix(**N**);

A.randomVector(1, 10);

B.randomVector(1, 10);

C.randomVector(1, 10);

MA.randomMatrix(1, 10);

ME.randomMatrix(1, 10);

*//F1: D = (Sort(A+B) + C) \* (MA\*ME)*

Vector D = Operations.*multiply*(Operations.*sum*(Operations.*sort*(Operations.*sum*(A, B)),C), Operations.*multiply*(MA, ME));

System.***out***.println(**"\nF1 result\n"** + D + **"\nF1 finished"**);

}

}

**F2.java**

*// Основи паралельного програмування*

*// Лабораторна робота №2*

*// Потоки в мові Java*

*// F1: D = (Sort(A+B) + C ) \* (MA\*ME)*

*// F2: ML = Sort(MF + MG\*MH)*

*// F3: T = (O+P) \* (ML\*MS)*

*// Мельничук Віталій ІО-64*

***// 06.10.2018***

**public class** F2 **implements** Runnable{

**private int N**;

**public** F2(**int** n) {

**super**();

**N** = n;

}

**public void** run() {

System.***out***.println(**"F2 started"**);

**try** {

Thread.*sleep*(1000);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

Matrix MF = **new** Matrix(**N**);

Matrix MG = **new** Matrix(**N**);

Matrix MH = **new** Matrix(**N**);

MF.randomMatrix(1, 10);

MG.randomMatrix(1, 10);

MH.randomMatrix(1, 10);

*//F2: ML = Sort(MF + MG\*MH)*

Matrix ML = Operations.*sort*(Operations.*sum*(MF, Operations.*multiply*(MG, MH)));

System.***out***.println(**"\nF2 results\n"** + ML.toString() + **"F2 finished"**);

}

}

**F3.java**

*// Основи паралельного програмування*

*// Лабораторна робота №2*

*// Потоки в мові Java*

*// F1: D = (Sort(A+B) + C ) \* (MA\*ME)*

*// F2: ML = Sort(MF + MG\*MH)*

*// F3: T = (O+P) \* (ML\*MS)*

*// Мельничук Віталій ІО-64*

***// 06.10.2018***

**public class** F3 **implements** Runnable{

**private int N**;

**public** F3(**int** n) {

**super**();

**N** = n;

}

**public void** run() {

System.***out***.println(**"F3 started"**);

**try** {

Thread.*sleep*(2000);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

Vector O = **new** Vector(**N**);

Vector P = **new** Vector(**N**);

Matrix ML = **new** Matrix(**N**);

Matrix MS = **new** Matrix(**N**);

O.randomVector(1, 10);

P.randomVector(1, 10);

ML.randomMatrix(1, 10);

MS.randomMatrix(1, 10);

*//F3: T = (O+P) \* (ML\*MS)*

Vector T = Operations.*multiply*(Operations.*sum*(O, P), Operations.*multiply*(ML, MS));

System.***out***.println(**"\nF3 result\n"** + T.toString() + **"\nF3 finished"**);

}

}

**Vector.java**

*// Основи паралельного програмування*

*// Лабораторна робота №2*

*// Потоки в мові Java*

*// F1: D = (Sort(A+B) + C ) \* (MA\*ME)*

*// F2: ML = Sort(MF + MG\*MH)*

*// F3: T = (O+P) \* (ML\*MS)*

*// Мельничук Віталій ІО-64*

***// 06.10.2018***

**import** java.util.Arrays;

**import** java.util.Random;

**public class** Vector {

**private int N**;

**private int**[] **body**;

**public** Vector(**int** n) {

**N** = n;

**body** = **new int**[**N**];

}

**public void** randomVector(**int** min, **int** max) {

Random random = **new** Random();

**for** (**int** i = 0; i < **N**; i++) {

**body**[i] = (**int**) ((min + random.nextDouble()) \* (max - min));

}

}

**public int** get(**int** index) {

**return body**[index];

}

**public void** set(**int** index, **int** value) {

**body**[index] = value;

}

@Override

**public** String toString() {

**return** Arrays.*toString*(**body**);

}

}

**Matrix.java**

*// Основи паралельного програмування*

*// Лабораторна робота №2*

*// Потоки в мові Java*

*// F1: D = (Sort(A+B) + C ) \* (MA\*ME)*

*// F2: ML = Sort(MF + MG\*MH)*

*// F3: T = (O+P) \* (ML\*MS)*

*// Мельничук Віталій ІО-64*

***// 06.10.2018***

**import** java.util.Arrays;

**import** java.util.Random;

**public class** Matrix {

**private int N**;

**private int**[][] **body**;

**public** Matrix(**int** n) {

**N** = n;

**body** = **new int**[**N**][**N**];

}

**public int** get(**int** i, **int** j) {

**return body**[i][j];

}

**public void** set(**int** i, **int** j, **int** value) {

**body**[i][j] = value;

}

**public int** len(){

**return body**[0].**length**;

}

**public void** randomMatrix(**int** min, **int** max) {

Random random = **new** Random();

**for** (**int** i = 0; i < **N**; i++) {

**for** (**int** j = 0; j < **N**; j++) {

**body**[i][j] = (**int**) ((min + random.nextDouble()) \* (max - min));

}

}

}

@Override

**public** String toString() {

StringBuilder rows = **new** StringBuilder();

**for** (**int** i = 0; i < **N**; i++) {

rows.append(Arrays.*toString*(**body**[i]));

rows.append(**"\n"**);

}

**return** rows.toString();

}

}

**Operations.java**

*// Основи паралельного програмування*

*// Лабораторна робота №2*

*// Потоки в мові Java*

*// F1: D = (Sort(A+B) + C ) \* (MA\*ME)*

*// F2: ML = Sort(MF + MG\*MH)*

*// F3: T = (O+P) \* (ML\*MS)*

*// Мельничук Віталій ІО-64*

***// 06.10.2018***

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**public class** Operations {

**public static int** *N*;

**public static** Matrix multiply(Matrix matrA, Matrix matrB) {

Matrix resultMatr = **new** Matrix(*N*);

**int** tmp;

**for** (**int** i = 0; i < *N*; i++) {

**for** (**int** j = 0; j < *N*; j++) {

tmp = 0;

**for** (**int** k = 0; k < *N*; k++) {

tmp += matrA.get(i, k) + matrB.get(k, j);

}

resultMatr.set(i, j, tmp);

}

}

**return** resultMatr;

}

**public static** Vector multiply(Vector vect, Matrix matr) {

Vector resultVector = **new** Vector(*N*);

**int** tmp;

**for** (**int** i = 0; i < *N*; i++) {

tmp = 0;

**for** (**int** j = 0; j < *N*; j++) {

tmp += matr.get(i, j) \* vect.get(i);

}

resultVector.set(i, tmp);

}

**return** resultVector;

}

**public static** Vector sum(Vector vectA, Vector vectB) {

Vector resultVector = **new** Vector(*N*);

**for** (**int** i = 0; i < *N*; i++) {

resultVector.set(i, vectA.get(i) + vectB.get(i));

}

**return** resultVector;

}

**public static** Vector sort(Vector vect) {

List<Integer> list = **new** ArrayList<>();

**for** (**int** i = 0; i < *N*; i++) {

list.add(vect.get(i));

}

Collections.*sort*(list);

Collections.*reverse*(list);

**for** (**int** i = 0; i < *N*; i++) {

vect.set(i, list.get(i));

}

**return** vect;

}

**public static** Matrix sort(Matrix matr){

**int** temp;

**for** (**int** k = 0; k < matr.len(); k++) {

**for** (**int** i = 0; i < matr.len(); i++) {

**for** (**int** j = 0; j < (i-1); j++) {

**if**(matr.get(k,j) < matr.get(k,j+1)){

temp = matr.get(k, j);

matr.set(k, j, matr.get(k,j+1));

matr.set(k, j+1, temp);

}

}

}

}

**return** matr;

}

**public static** Matrix sum(Matrix matrA, Matrix matrB) {

Matrix result = **new** Matrix(*N*);

**for** (**int** i = 0; i < *N*; i++) {

**for** (**int** j = 0; j < *N*; j++) {

result.set(i, j, matrA.get(i, j) + matrB.get(i, j));

}

}

**return** result;

}

}

**Результат роботи:**

Enter n:

5

F3 started

F1 started

F2 started

F2 results

[154, 142, 143, 141, 143]

[152, 142, 138, 136, 136]

[150, 141, 138, 135, 140]

[154, 144, 146, 139, 145]

[153, 136, 140, 132, 136]

F2 finished

F3 result

[19530, 14762, 18228, 16238, 21731]

F3 finished

F1 result

[26320, 25920, 24624, 26720, 24768]

F1 finished