

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«кИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО»

Кафедра обчислювальної техніки

Лабораторна робота №3

на тему : **«**Потоки в мові C#. Задачі**»**

Виконала

студентка 3 курсу

гр. ІО-51

ФІОТ

Поштацька К. В.

5115

Варіант 1.15, 2.29, 3.21

Перевірив:

Доцент, к.т.н.

Корочкін О.В

.

Київ 2017

**Мета роботи:** вивчення засобів мови C# для роботи з процесами (потоками).

**Завдання**

Розробити програму, що містить паралельні потоки (задачі), кожна з яких реалізує функцію F1, F2, F3 згідно з варіантом з додатку Б.

**Варіант завдання**

**1.15** MD = MAX(A + B + C) \*(MA\*ME)

**2.29** MF = (MG + MH)\*(MK \* ML)\*(MG + ML)

**3.21** S = SORT(O\*MO)\*(MS \*MT)

**Лістинг програми**

**Клас Lab3**

/\*

on the topic "Threads in C# language. Tasks"

Variant: 1.15 2.29 3.21

T1: MD=MAX(A+B+C)\*(MA\*ME)

T2: MF=(MG+MH)\*(MK\*ML)\*(MG+ML)

T3: S=SORT(O\*MO)\*(MS\*MT)

Created: 26.09.2017 8:52 AM

Author: Poshtatska Katerina IO-51

\*/

using System;

using System.Threading;

namespace Lab3

{

class Lab3

{

static Data data;

static void Main()

{

Console.WriteLine("Lab3 started");

int n = 1000;

data = new Data(n);

Thread T1 = new Thread(TF1);

Thread T2 = new Thread(TF2);

Thread T3 = new Thread(TF3);

//Starting the threads

T1.Start();

T2.Start();

T3.Start();

//The message will be displayed after all thread ending

try

{

T1.Join();

}

catch (ThreadInterruptedException e)

{

Console.WriteLine(e.ToString());

}

try

{

T2.Join();

}

catch (ThreadInterruptedException e)

{

Console.WriteLine(e.ToString());

}

try

{

T3.Join();

}

catch (ThreadInterruptedException e)

{

Console.WriteLine(e.ToString());

}

Console.WriteLine("Lab3 finished");

Console.ReadKey();

}

//Thread function TF1

static void TF1()

{

Console.WriteLine("Task T1 started");

int[] A = new int[data.getN()];

int[] B = new int[data.getN()];

int[] C = new int[data.getN()];

int[,] MA = new int[data.getN(),data.getN()];

int[,] ME = new int[data.getN(),data.getN()];

if (data.getN() <= 10)

{

A = data.Vector\_Input("T1 A");

B = data.Vector\_Input("T1 B");

C = data.Vector\_Input("T1 C");

MA = data.Matrix\_Input("T1 MA");

ME = data.Matrix\_Input("T1 ME");

}

else

{

A = data.Vector\_Input1();

B = data.Vector\_Input1();

C = data.Vector\_Input1();

MA = data.Matrix\_Input1();

ME = data.Matrix\_Input1();

}

int[,] MD = { { } };

try

{

Thread.Sleep(1000);

MD = data.Func1(A, B, C, MA, ME);

Thread.Sleep(1000);

}

catch (ThreadInterruptedException e)

{

Console.WriteLine(e);

}

if (data.getN() <= 10)

{

Console.WriteLine("MD = ");

data.Matrix\_Output(MD);

Console.WriteLine("");

}

Console.WriteLine("Task T1 finished");

}

//Thread function TF2

static void TF2()

{

Console.WriteLine("Task T2 started");

int[,] MG = new int[data.getN(),data.getN()];

int[,] MH = new int[data.getN(),data.getN()];

int[,] ML = new int[data.getN(),data.getN()];

int[,] MK = new int[data.getN(),data.getN()];

if (data.getN() <= 1)

{

MG = data.Matrix\_Input("T2 MG");

MH = data.Matrix\_Input("T2 MH");

ML = data.Matrix\_Input("T2 ML");

MK = data.Matrix\_Input("T2 MK");

}

else

{

MG = data.Matrix\_Input1();

MK = data.Matrix\_Input1();

ML = data.Matrix\_Input1();

MH = data.Matrix\_Input1();

}

int[,] MF = { { } };

try

{

Thread.Sleep(1000);

MF = data.Func2(MG, MK, ML, MH);

Thread.Sleep(1000);

}

catch (ThreadInterruptedException e)

{

Console.WriteLine(e);

}

if (data.getN() <= 10)

{

Console.WriteLine("MF = ");

data.Matrix\_Output(MF);

}

Console.WriteLine("Task T2 finished");

}

//Thread function TF3

static void TF3()

{

Console.WriteLine("Task T3 started");

int[] O = new int[data.getN()];

int[,] MO = new int[data.getN(),data.getN()];

int[,] MS = new int[data.getN(),data.getN()];

int[,] MT = new int[data.getN(),data.getN()];

if (data.getN() <= 1)

{

O = data.Vector\_Input("T3 O");

MO = data.Matrix\_Input("T3 MO");

MS = data.Matrix\_Input("T3 MS");

MT = data.Matrix\_Input("T3 MT");

}

else

{

O = data.Vector\_Input1();

MO = data.Matrix\_Input1();

MS = data.Matrix\_Input1();

MT = data.Matrix\_Input1();

}

int[] S = data.Func3(O, MO, MS, MT);

if (data.getN() <= 10)

{

Console.WriteLine("S =");

data.Vector\_Output(S);

}

Console.WriteLine("Task T3 finished");

}

}

}

**Клас Data**

//Implementation of class called Data

using System;

namespace Lab3

{

public class Data

{

private int n;

public Data(int n) {

this.n = n;

}

public int getN() {

return n;

}

//Input Vector from keyboard

public int[] Vector\_Input(String name) {

int[] A = new int[n];

for (int i = 0; i < n; i++) {

Console.Write(name + "[" + i + "] = ");

A[i] = Convert.ToInt32(Console.ReadLine());

Console.WriteLine();

}

return A;

}

//Fill Vector with 1

public int[] Vector\_Input1() {

int[] A = new int[n];

for (int i = 0; i < n; i++) {

A[i] = 1;

}

return A;

}

//Print Vector on screen

public void Vector\_Output(int[] A) {

for (int i = 0; i < n; i++) {

Console.Write(A[i] + " ");

}

}

//Input Matrix from keyboard

public int[,] Matrix\_Input(String name) {

int[,] A = new int[n,n];

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

Console.Write(name + "[" + i + "]" + "[" + j + "] = ");

A[i,j] = Convert.ToInt32(Console.ReadLine());

Console.WriteLine();

}

}

return A;

}

//Fill Matrix with 1

public int[,] Matrix\_Input1() {

int[,] A = new int[n,n];

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

A[i,j] = 1;

}

}

return A;

}

//Print Matrix on screen

public void Matrix\_Output(int[,] A) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

Console.Write(A[i,j] + " ");

}

Console.WriteLine();

}

}

//Adds the vector A to the vector B

private int[] Vector\_Add(int[] A, int[] B, int[] C) {

int[] D = new int[n];

for (int i = 0; i < n; i++) {

D[i] = A[i] + B[i] + C[i];

}

return D;

}

//Adds the matrix A to the matrix B

private int[,] Matrix\_Add(int[,] A, int[,] B) {

int[,] C = new int[n,n];

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

C[i,j] = A[i,j] + B[i,j];

}

}

return C;

}

//Subtracts the matrix B from the matrix A

private int[,] Matrix\_Multiply(int[,] A, int[,] B) {

int[,] C = new int[n,n];

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

C[i,j] = 0;

for (int k = 0; k < n; k++) {

C[i,j] += A[i,k] \* B[k,j];

}

}

}

return C;

}

//Multiply Vector on Matrix

private int[] Matrix\_Vector\_Multiply(int[,] A, int[] B) {

int[] C = new int[n];

for (int i = 0; i < n; i++) {

C[i] = 0;

for (int j = 0; j < n; j++) {

C[i] += A[j,i] \* B[i];

}

}

return C;

}

//Multiply Value on Matrix

private int[,] Matrix\_Value\_Multiply(int[,] A, int B) {

int[,] C = new int[n,n];

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

C[i,j] = A[j,i] \* B;

}

}

return C;

}

//Search for maximum element in Vector

private int Vector\_Max(int[] A) {

int B = A[0];

for (int i = 0; i < n; i++) {

if (A[i] > B) {

B = A[i];

}

}

return B;

}

//Sorting Matrix. Uses bubble sorting like sorting simple vector.

private int[] Vector\_Sort(int[] A) {

Boolean Finished;

int Temp = 0;

int[] B = new int[n];

for (int i = 0; i < n; i++) {

B[i] = A[i];

}

for (int i = 0; i < n; i++) {

Finished = true;

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

if (B[j - 1] > B[j]) {

Temp = B[j - 1];

B[j - 1] = B[j];

B[j] = Temp;

Finished = false;

}

}

if (Finished) {

break;

}

}

return B;

}

/\*

Calculating Func1

MD=MAX(A+B+C)\*(MA\*ME)

\*/

public int[,] Func1(int[] A, int[] B, int[] C, int[,] MA, int[,] ME) {

int[,] MD = Matrix\_Value\_Multiply(Matrix\_Multiply(MA, ME), Vector\_Max(Vector\_Add(A, B, C)));

return MD;

}

/\*

Calculating Func2

MF=(MG+MH)\*(MK\*ML)\*(MG+ML)

\*/

public int[,] Func2(int[,] MG, int[,] MH, int[,] MK, int[,] ML) {

int[,] MF = Matrix\_Multiply(Matrix\_Multiply(Matrix\_Add(MG, MH), Matrix\_Multiply(MK, ML)), Matrix\_Add(MG, ML));

return MF;

}

/\*

Calculating Func3

S=SORT(O\*MO)\*(MS\*MT)

\*/

public int[] Func3(int[] O, int[,] MO, int[,] MS, int[,] MT) {

int[] S = Matrix\_Vector\_Multiply(Matrix\_Multiply(MS, MT), Vector\_Sort(Matrix\_Vector\_Multiply(MO, O)));

return S;

}

}

}