Kpi-best

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. І.СІКОРСЬКОГО»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Лабораторна робота №1**

з дисципліни «Паралельні та розподілені обчислення»

на тему: «Процеси в мові Ada. Задачі»

Виконала:

студентка 3-го курсу

факультету ІОТ

групи ІО-43

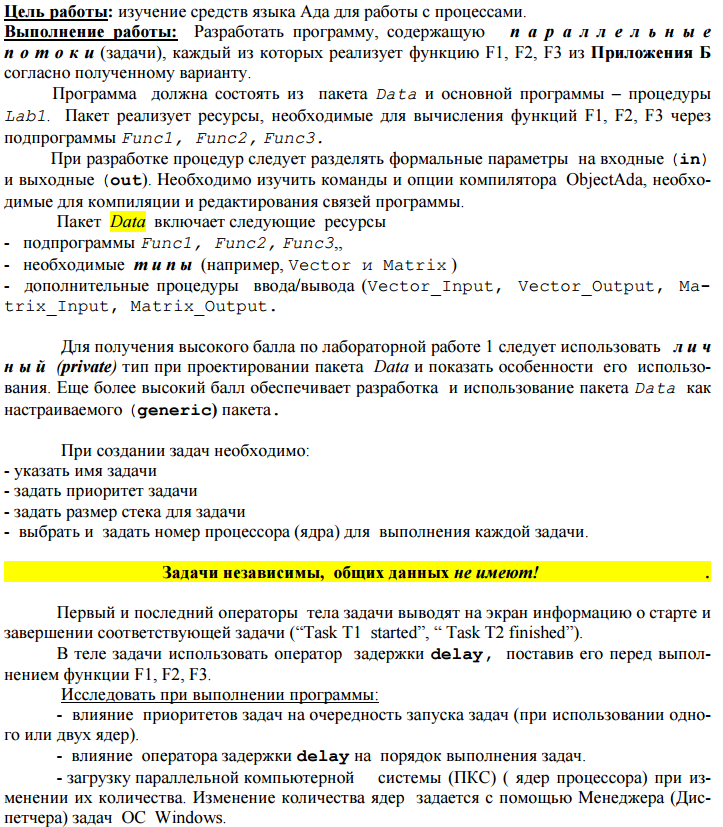
Даніленко Н.

Перевірив:

доц.

Долголенко О. М.

Київ 2016



**Варіант 4**

1.4 C = A + SORT(B) \* (MA \* ME).

2.4 MG = MAX(MH) \* (MK \* ML).

3.4 O = SORT(P) \* SORT (MR \* MS).

**Лістинг:**

**Lab1.adb**

with Data;

with Ada.Text\_IO; use Ada.Text\_IO;

with System.Multiprocessors; use System.Multiprocessors;

procedure Lab1 is

N : Integer := 5;

package Lab1Data is new Data(N);

use Lab1Data;

vectorA, vectorB, resultVector, vectorT, vectorO, vectorP : Vector;

matrixA, matrixB, matrixP, matrixR, matrixS, matrixF, matrixG, matrixH, matrixK: Matrix;

CPU0: CPU\_Range := 0;

CPU1: CPU\_Range := 1;

CPU2: CPU\_Range := 2;

procedure tasks is

task T1 is

pragma Priority(1);

pragma Storage\_Size(100000);

pragma CPU(CPU0);

end;

task body T1 is

vectorA: Vector;

matrixA, matrixE: Matrix;

begin

Put\_Line("T1 started");

randomVector(vectorA);

randomVector(vectorB);

randomMatrix(matrixA);

randomMatrix(matrixE);

F1(vectorA, vectorB, matrixA, matrixE, resultVector);

Put\_Line("T1 finished");

end T1;

task T2;

task body T2 is

matrixH, matrixK, matrixL, matrixF: Matrix;

begin

Put\_Line("T2 started");

fillMatrix(matrixH);

fillMatrix(matrixK);

fillMatrix(matrixL);

F2(matrixH, matrixK, matrixL, matrixF);

Put\_Line("T2 finished");

end T2;

task T3 is

pragma Priority(5);

pragma Storage\_Size(100000);

end;

task body T3 is

vectorP, vectorT: Vector;

matrixR, matrixS: Matrix;

begin

Put\_Line("T3 started");

randomVector(vectorP);

fillMatrix(matrixR);

fillMatrix(matrixS);

F3(matrixR, matrixS, vectorP, vectorT);

Put\_Line("T3 finished");

end T3;

begin

null;

end tasks;

begin

Put\_Line("Lab 1 started");

tasks;

Put\_Line("--- Task 1 results");

putVector(resultVector);

Put\_Line("---");

Put\_Line("--- Task 2 results");

putMatrix(matrixF);

Put\_Line("---");

Put\_Line("--- Task 3 results");

putVector(vectorT);

Put\_Line("---");

end Lab1;

**Data.ads**

with Data;

with Ada.Text\_IO; use Ada.Text\_IO;

with System.Multiprocessors; use System.Multiprocessors;

procedure Lab1 is

N : Integer := 5;

package Lab1Data is new Data(N);

use Lab1Data;

vectorA, vectorB, resultVector, vectorT, vectorO, vectorP : Vector;

matrixA, matrixB, matrixP, matrixR, matrixS, matrixF, matrixG, matrixH, matrixK: Matrix;

CPU0: CPU\_Range := 0;

CPU1: CPU\_Range := 1;

CPU2: CPU\_Range := 2;

procedure tasks is

task T1 is

pragma Priority(1);

pragma Storage\_Size(100000);

pragma CPU(CPU0);

end;

task body T1 is

vectorA: Vector;

matrixA, matrixE: Matrix;

begin

Put\_Line("T1 started");

randomVector(vectorA);

randomVector(vectorB);

randomMatrix(matrixA);

randomMatrix(matrixE);

F1(vectorA, vectorB, matrixA, matrixE, resultVector);

Put\_Line("T1 finished");

end T1;

task T2;

task body T2 is

matrixH, matrixK, matrixL, matrixF: Matrix;

begin

Put\_Line("T2 started");

fillMatrix(matrixH);

fillMatrix(matrixK);

fillMatrix(matrixL);

F2(matrixH, matrixK, matrixL, matrixF);

Put\_Line("T2 finished");

end T2;

task T3 is

pragma Priority(5);

pragma Storage\_Size(100000);

end;

task body T3 is

vectorP, vectorT: Vector;

matrixR, matrixS: Matrix;

begin

Put\_Line("T3 started");

randomVector(vectorP);

fillMatrix(matrixR);

fillMatrix(matrixS);

F3(matrixR, matrixS, vectorP, vectorT);

Put\_Line("T3 finished");

end T3;

begin

null;

end tasks;

begin

Put\_Line("Lab 1 started");

tasks;

Put\_Line("--- Task 1 results");

putVector(resultVector);

Put\_Line("---");

Put\_Line("--- Task 2 results");

putMatrix(matrixF);

Put\_Line("---");

Put\_Line("--- Task 3 results");

putVector(vectorT);

Put\_Line("---");

end Lab1;

**Data.adb**

with Data;

with Ada.Text\_IO; use Ada.Text\_IO;

with System.Multiprocessors; use System.Multiprocessors;

procedure Lab1 is

N : Integer := 5;

package Lab1Data is new Data(N);

use Lab1Data;

vectorA, vectorB, resultVector, vectorT, vectorO, vectorP : Vector;

matrixA, matrixB, matrixP, matrixR, matrixS, matrixF, matrixG, matrixH, matrixK: Matrix;

CPU0: CPU\_Range := 0;

CPU1: CPU\_Range := 1;

CPU2: CPU\_Range := 2;

procedure tasks is

task T1 is

pragma Priority(1);

pragma Storage\_Size(100000);

pragma CPU(CPU0);

end;

task body T1 is

vectorA: Vector;

matrixA, matrixE: Matrix;

begin

Put\_Line("T1 started");

randomVector(vectorA);

randomVector(vectorB);

randomMatrix(matrixA);

randomMatrix(matrixE);

F1(vectorA, vectorB, matrixA, matrixE, resultVector);

Put\_Line("T1 finished");

end T1;

task T2;

task body T2 is

matrixH, matrixK, matrixL, matrixF: Matrix;

begin

Put\_Line("T2 started");

fillMatrix(matrixH);

fillMatrix(matrixK);

fillMatrix(matrixL);

F2(matrixH, matrixK, matrixL, matrixF);

Put\_Line("T2 finished");

end T2;

task T3 is

pragma Priority(5);

pragma Storage\_Size(100000);

end;

task body T3 is

vectorP, vectorT: Vector;

matrixR, matrixS: Matrix;

begin

Put\_Line("T3 started");

randomVector(vectorP);

fillMatrix(matrixR);

fillMatrix(matrixS);

F3(matrixR, matrixS, vectorP, vectorT);

Put\_Line("T3 finished");

end T3;

begin

null;

end tasks;

begin

Put\_Line("Lab 1 started");

tasks;

Put\_Line("--- Task 1 results");

putVector(resultVector);

Put\_Line("---");

Put\_Line("--- Task 2 results");

putMatrix(matrixF);

Put\_Line("---");

Put\_Line("--- Task 3 results");

putVector(vectorT);

Put\_Line("---");

end Lab1;

**Результати виконання програми:**

