МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний авіаційний університет

Лабораторна робота № 1

з дисципліни «Високопродуктивні обчислювальні системи»

Тема: Процеси в мові Java.

Київ – 2015

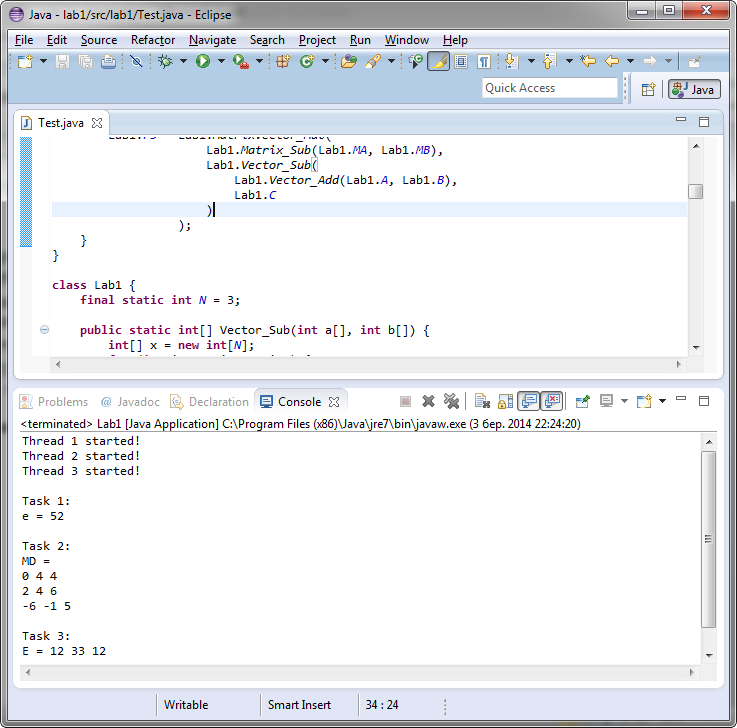
**Завдання:**

Розробити паралельну програму, що містить потоки мови Java для обчислення трьох функцій, що представлено відповідним варіантом.

Дослідити вплив пріоритетів задач і оператора затримки на послідовність виконання задач.

1. e = ((A - B)\*(C + D))
2. MU = SORT(MA + TRANS(MB) - TRANS(MC))
3. E = (A + B - C)\*(MA - MB)

**Результат роботи програми:**



**Лістинг програми:**

**package** lab1;

**class** Thr1 **extends** Thread {

**public void** run(){

System.*out*.println("Thread 1 started!");

Lab1.*F1* = Lab1.*Vector\_Mul*(

Lab1.*Vector\_Sub*(Lab1.*A*, Lab1.*B*),

Lab1.*Vector\_Add*(Lab1.*C*, Lab1.*D*)

);

}

}

**class** Thr2 **extends** Thread {

**public void** run(){

System.*out*.println("Thread 2 started!");

Lab1.*F2* = Lab1.*Matrix\_Sort*(Lab1.*Matrix\_Sub*(

Lab1.*Matrix\_Add*(

Lab1.*Matrix\_Trans*(Lab1.*MB*),

Lab1.*MA*

),

Lab1.*Matrix\_Trans*(Lab1.*MC*)

));

}

}

**class** Thr3 **extends** Thread {

**public void** run(){

System.*out*.println("Thread 3 started!");

Lab1.*F3* = Lab1.*MatrixVector\_Mul*(

Lab1.*Matrix\_Sub*(Lab1.*MA*, Lab1.*MB*),

Lab1.*Vector\_Sub*(

Lab1.*Vector\_Add*(Lab1.*A*, Lab1.*B*),

Lab1.*C*

)

);

}

}

**class** Lab1 {

**final static int** *N* = 3;

**public static int**[] Vector\_Sub(**int** a[], **int** b[]) {

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

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

x[i] = a[i] - b[i];

}

**return** x;

}

**public static int**[] Vector\_Add(**int** a[], **int** b[]) {

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

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

x[i] = a[i] + b[i];

}

**return** x;

}

p**ublic static void** PrintVector(**int** a[]){

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

System.*out*.print(a[i] + " ");

}

System.*out*.println();

}

**public static int**[][] Matrix\_Add(**int** MA[][], **int** MB[][]){

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

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

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

MX[i][j] = MA[i][j] + MB[i][j];

}

}

**return** MX;

}

**public static int**[][] Matrix\_Sub(**int** MA[][], **int** MB[][]){

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

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

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

MX[i][j] = MA[i][j] - MB[i][j];

}

}

**return** MX;

}

**public static int**[] MatrixVector\_Mul(**int** MA[][], **int** A[]){

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

**int**tmp;

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

tmp = 0;

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

tmp = tmp + MA[i][j] \* A[j];

}

x[i] = tmp;

}

**return** x;

}

**public static void** Matrix\_Output(**int** MA[][]){

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

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

System.*out*.print(MA[i][j] + " ");

}

System.*out*.println();

}

}

**public static int**[][] Matrix\_Trans(**int** MA[][]){

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

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

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

MX[i][j] = MA[j][i];

}

}

**return** MX;

}

**public static int** Vector\_Mul(**int** A[], **int** B[]){

**int** x = 0;

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

x = x + A[i] \* B[i];

}

**return** x;

}

**public static int**[][] Matrix\_Sort(**int** MA[][]) {

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

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

**for** (**int** k = 1; k <*N* - j; k++) {

**if** (MA[i][k] < MA[i][k - 1]) {

**int**temp = MA[i][k];

MA[i][k] = MA[i][k - 1];

MA[i][k - 1] = temp;

}

}

}

}

**return** MA;

}

**public static int**[] *A* = {3, 3, 3};

**public static int**[] *B* = {1, 1, 1};

**public static int**[] *C* = {2, 3, 1};

**public static int**[] *D* = {9, 7, 4};

**public static int**[][] *MA* = {{1, 7, 3},

{4, 9, 6},

{7, 3, 2}};

**public static int**[][] *MB* = {{1, 1, 1},

{1, 0, 0},

{3, 2, 1}};

**publicstaticint**[][] *MC* = {{2, 3, 3},

{4, 5, 4},

{2, 2, 9}};

**public static int***F1* = 0;

**public static int**[][] *F2* = **newint**[*N*][*N*];

**public static int**[] *F3* = **new int**[*N*];

**public static void** main(Stringargs[]) **throws**InterruptedException{

Thr1 T1 = **new** Thr1();

Thr2 T2 = **new** Thr2();

Thr3 T3 = **new** Thr3();

T1.start();

T2.start();

T3.start();

T1.join();

T2.join();

T3.join();

System.*out*.println();

System.*out*.println("Task 1:");

System.*out*.println("e = " + *F1*);

System.*out*.println();

System.*out*.println("Task 2:");

System.*out*.println("MD = ");

*Matrix\_Output*(*F2*);

System.*out*.println();

System.*out*.println("Task 3:");

System.*out*.print("E = ");

*PrintVector*(*F3*);

System.*out*.println();

}

}

Мова Java, як і Ада, має вбудовані можливості для роботи з потоками. Для створення потоку потрібно унаслідуватись від класу Threadта створити метод run(), в якому описати дії, які буде виконувати даний потік. Для запуску потоку в основній програмі слід створити об’єкт потоку та викликати його метод start(). Для зміни пріоритету використовуються метод setPriority(). Призупинення виконання потоку відбувається за допомогою методу sleep(). Для очікування завершення потоку використовується метод join().