НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА
Автоматизованих Систем Обробки інформації та Управління

Лабораторна робота №1

з дісципліни «Паралельне програмуваня»

Виконав:
студент 3 курсу
ФІОТ гр. ІП-31
Степанов Олександр

Перевірив:
Корочкін О. В.

Київ 2015 р.

Завданняя:

F1: 1.9: MC = MIN(A) *(MD*MB)
F2: 2.4: MN = MAX(ML) *(MK*MO)
F3: 3.4: Z = SORT(R)*SORT(MW*MV)

Лістинг:

Compiling: lab1.adb (source file time stamp: 2015-09-14 19:16:44)

```
 1. ------------------------------------------------------------------------
 2. --                                                                    --
 3. --                      Parallel Programming                          --
 4. --         Laboratory work #1. Ada. Subprograms and packages          --
 5. --                                                                    --
 6. --  File: lab1.ada                                                   --
 7. --  Task: F1: MC = MIN(A) * (MD * MB)                                --
 8. --        F2: MN = MAX(ML) * (MK * MO)                               --
 9. --        F3: Z = SORT(R) * SORT(MW * MV)                            --
10. --                                                                    --
11. --  Author: Stepanov Alexander, group IP - 31                         --
12. --  Date: 14.09.2015                                                 --
13. --                                                                    --
14. ------------------------------------------------------------------------
15.
16. with Data, Ada.Text_IO, Ada.Integer_Text_IO;
17. use Ada.Text_IO, Ada.Integer_Text_IO;
18.
19. procedure Lab1 is
20.     N : Integer := 3;
21.     package mainData is new Data(3);
22.          use mainData;
23.    A, R, Z : Vector;
24.          MB, MC, MD, MK, ML, MN, MO, MV, MW : Matrix;
25. begin
26.
27.     ------------------------------
28.     --        Input data        --
29.     ------------------------------
30.     Put("All vectors had ");
31.     Put(N);
32.     Put(" items");
33.     New_Line;
34.
35.    ----------- input F1 ----------
36.     Put_Line(" -- Arguments for F1");
37.
38.     Put("Enter vector A: ");
39.     New_Line;
40.     Input_Vector(A);
41.
42.     New_Line;
43.     Put("Enter MD: ");
44.     New_Line;
45.     Input_Matrix(MD);
46.
47.     New_Line;
48.     Put("Enter MB: ");
49.     New_Line;
```

```
    Input_Matrix(MB);

    ----------- input F2 ----------
    Put_Line(" -- Arguments for F2");

    New_Line;
    Put("Enter ML: ");
    New_Line;
    Input_Matrix(ML);

    New_Line;
    Put("Enter MK: ");
    New_Line;
    Input_Matrix(MK);

    New_Line;
    Put("Enter MO: ");
    New_Line;
    Input_Matrix(MO);

    ----------- input F3 ----------
    Put_Line(" -- Arguments for F3");

    Put("Enter vector R: ");
    New_Line;
    Input_Vector(R);

    New_Line;
    Put("Enter MW: ");
    New_Line;
    Input_Matrix(MW);

    New_Line;
    Put("Enter MV: ");
    New_Line;
    Input_Matrix(MV);

    -------------------------------
    --        Calculation        --
    -------------------------------
    F1(A, MD, MB, MC);
    F2(ML, MK, MO, MN);
   F3(R, MW, MV, Z);

    -------------------------------
    --        Output data        --
    -------------------------------
    Put_Line(" -- Arguments for F3");

    New_Line;
    Put("F1 = ");
    Output_Matrix(MC);

    New_Line;
    Put("F2 = ");
    Output_Matrix(MN);

    New_Line;
    Put("F3 = ");
    Output_Vector(Z);

end Lab1;
```

Compiling: data.adb (source file time stamp: 2015-09-14 18:49:22)

```
 1. ------------------------------------------------------------------------
 2. --                                                                    --
 3. --                      Parallel Programming                          --
 4. --         Laboratory work #1. Ada. Subprograms and packages          --
 5. --                                                                    --
 6. --  File: lab1.ada                                                   --
 7. --  Task: F1: MC = MIN(A) * (MD * MB)                                --
 8. --        F2: MN = MAX(ML) * (MK * MO)                               --
 9. --        F3: Z = SORT(R) * SORT(MW * MV)                            --
10. --                                                                    --
11. --  Author: Stepanov Alexander, group IP - 31                         --
12. --  Date: 14.09.2015                                                 --
13. --                                                                    --
14. ------------------------------------------------------------------------
15.
16. with Ada.Text_IO, Ada.Integer_Text_IO;
17. use Ada.Text_IO, Ada.Integer_Text_IO;
18.
19. package body Data is
20.
21.     ------------------------------------------------------------------
22.     --        additional functions for calculation (prototypes)       --
23.     ------------------------------------------------------------------
24.     -- Matrix * Matrix
25.     function "*" (Left  : Matrix; Right : Matrix) return Matrix;
26.     -- Vector * Matrix
27.     function "*" (Left  : Vector; Right : Matrix) return Vector;
28.     -- Integer * Matrix
29.     function "*" (Left  : Integer; Right : Matrix) return Matrix;
30.     -- maximum value of the matrix
31.     function Max(A : Matrix) return Integer;
32.     -- minimum value of the vector
33.     function Min(A : Vector) return Integer;
34.     -- vector sorting
35.     function Sort(A : Vector) return Vector;
36.     -- matrix sorting
37.     function Sort(MA : Matrix) return Matrix;
38.
39.     ------------------------------------------------------------------
40.     --                         main procedures                      --
41.     ------------------------------------------------------------------
42.     --  F1: MC = MIN(A) * (MD * MB)
43.     procedure F1(A : in Vector; MD, MB: in Matrix; MC: out Matrix) is
44.     begin
45.         MC := MIN(A) * (MD * MB);
46.     end F1;
47.
48.     -- F2: MN = MAX(ML) * (MK * MO)
49.     procedure F2(ML, MK, MO: in Matrix; MN: out Matrix) is
50.     begin
51.         MN := MAX(ML) * (MK * MO);
52.     end F2;
53.
54.     -- F3: Z = SORT(R) * SORT(MW * MV)
55.     procedure F3(R: in Vector; MW, MV: in Matrix; Z: out Vector) is
56.     begin
57.         Z := SORT(R) * SORT(MW * MV);
58.     end F3;
59.
60.     ------------------------------------------------------------------
61.     --        additional functions for calculation (realization)      --
```

```
   ---------------------------------------------------------------------
   -- Matrix * Matrix
   function "*" (Left  :  Matrix; Right :  Matrix) return Matrix is
       Result : Matrix;
   begin
       for i in Index loop
           for J in Index loop
               Result(I)(J) := 0;
                   for K in Index loop
                       Result(I)(J) := Result(I)(J) + Left(I)(K) * Right(K)(J);
                   end loop;
               end loop;
       end loop;
       return Result;
   end  "*";

   -- Vector * Matrix
   function "*" (Left : Vector; Right : Matrix) return Vector is
       R : Vector;
   begin
       for J in Index loop
           R(j) := 0;
           begin
               for K in Index loop
                   R(J) := R(J) + Left(K) * Right(K)(J);
               end loop;
           end;
       end loop;
       return R;
   end "*";

   -- Integer * Matrix
   function "*" (Left : Integer; Right : Matrix) return Matrix is
       Result : Matrix;
   begin
       for i in 1..N loop
           for j in 1..N loop
               Result(i)(j) := Left * Right(i)(j);
           end loop;
       end loop;
       return Result;
   end "*";

   -- Maximum value of the matrix
   function Max(A : Matrix) return Integer is
       Var : Integer := A(1)(1);
   begin
       for i in 1..N loop
           for j in 1..N loop
               if A(i)(j) > Var then
                   Var := A(i)(j);
               end if;
           end loop;
       end loop;
       return Var;
   end Max;

   -- Minimum value of the vector
   function Min(A : Vector) return Integer is
       Var : Integer := A(1);
   begin
       for i in 1..N loop
           if A(i) < Var then
```

```
                Var := A(i);
            end if;
        end loop;
        return Var;
    end Min;

    -- vector sorting
    function Sort(A : Vector) return Vector is
        M : Vector := A;
        Buf : Integer;
        K : Integer;
    begin
        for i in 1..(n - 1) loop
            K := i;
            for j in (i + 1)..n loop
                if M(k) > M(j) then
                    K:=j;
                end if;
            end loop;
            Buf := M(k);
            M(k) := M(i);
            M(i) := buf;
        end loop;
        return M;
    end Sort;

    -- matrix sorting
    function Sort(MA : Matrix) return Matrix is
        MT : Matrix;
    begin
        for i in 1..n loop
            MT(i) := Sort(MA(i));
        end loop;
        return MT;
    end Sort;

    ------------------------------------------------------------------------
    --                      input\output functions:                       --
    ------------------------------------------------------------------------
    -- input values for vectors
    procedure Input_Vector (V : out Vector) is
    begin
        for I in Index loop
            Get(V(I));
        end loop;
    end Input_Vector;

    -- input values for matrices
    procedure Input_Matrix (MA : out Matrix) is
    begin
        for I in Index loop
            for J in Index loop
                Get(MA(I)(J));
            end loop;
        end loop;
    end Input_Matrix;

    -- output values of vectors
    procedure Output_Vector (V : in Vector) is
    begin
        New_Line;
        for I in Index loop
            Put(Item => V(I), Width => 5);
```

```
   188.         end loop;
   189.         New_Line;
   190.     end Output_Vector;
   191.
   192.     -- output values for matrices
   193.     procedure Output_Matrix (MA : in Matrix) is
   194.     begin
   195.         New_Line;
   196.         for I in Index loop
   197.             for J in Index loop
   198.                 Put(Item => MA(i)(j), Width => 5);
   199.             end loop;
   200.             New_line;
   201.         end loop;
   202.     end Output_Matrix;
   203.
   204. end Data;

Compiling: data.ads (source file time stamp: 2015-09-14 18:50:16)

     1. ------------------------------------------------------------------------
     2. --                                                                    --
     3. --                      Parallel Programming                          --
     4. --         Laboratory work #1. Ada. Subprograms and packages          --
     5. --                                                                    --
     6. --  File: lab1.ada                                                   --
     7. --  Task: F1: MC = MIN(A) * (MD * MB)                                --
     8. --        F2: MN = MAX(ML) * (MK * MO)                               --
     9. --        F3: Z = SORT(R) * SORT(MW * MV)                            --
    10. --                                                                    --
    11. --  Author: Stepanov Alexander, group IP - 31                         --
    12. --  Date: 14.09.2015                                                 --
    13. --                                                                    --
    14. ------------------------------------------------------------------------
    15.
    16. generic
    17.
    18.     N : in Integer;
    19.
    20. package Data is
    21.
    22.     type Vector is private;
    23.     type Matrix is private;
    24.
    25.     -- F1: MC = MIN(A) * (MD * MB)
    26.     procedure F1(A : in Vector; MD, MB: in Matrix; MC: out Matrix);
    27.     -- F2: MN = MAX(ML) * (MK * MO)
    28.     procedure F2(ML, MK, MO: in Matrix; MN: out Matrix);
    29.     -- F3: Z = SORT(R) * SORT(MW * MV)
    30.     procedure F3(R: in Vector; MW, MV: in Matrix; Z: out Vector);
    31.
    32.     -- input values for vectors
    33.     procedure Input_Vector(V : out Vector);
    34.     -- input values for matrices
    35.     procedure Input_Matrix(MA : out Matrix);
    36.     -- output values of vectors
    37.     procedure Output_Vector(V : in Vector);
    38.     -- output values of matricescr
    39.     procedure Output_Matrix(MA : in Matrix);
    40.
    41.     private
    42.         subtype Index is Integer range 1..N;
    43.         type Vector is array (Index) of Integer;
```

```
44.         type Matrix is array (Index) of Vector;
45.
46. end Data;
```