

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1

Ада. Пакети

Виконав:
студент групи ІП-32
Ковальчук О. М.
Перевірів:
Корочкін О. В.

Київ - 2015

Лабораторна робота №1. АДА. Пакети

Мета роботи: вивчення структури програми та особливостей реалізації механізму підпрограм та пакетів у мові Ада

Мова програмування: Ада

Завдання: розробити програму мовою Ада, яка реалізує обчислення трьох функцій F1, F2, F3, згідно варіанту. Програма повинна складатися із пакету Data і основної програми – процедури Lab1. У пакеті мають бути реалізовані ресурси, необхідні для обчислення функцій F1, F2, F3.

Функції:

F1: $C = A - B * (MA * MD)$

F2: $o = \text{Min}(MK * MM)$

F3: $T = (MS * MZ) * (W + X)$

Лістинг програми

GNAT 4.5.4

Copyright 1992-2010, Free Software Foundation, Inc.

Compiling: lab1.adb (source file time stamp: 2015-09-23 20:39:56)

```
1. -----
2. --      Parallel programming      --
3. --      Lab 1                     --
4. --                               --
5. -- Func1: C = A - B * (MA * MD)   --
6. -- Func2: o = Min(MK * MM)        --
7. -- Func3: T = (MS * MZ) * (W + X) --
8. --                               --
9. -- @author Olexandr Kovalchuk     --
10. -- @group IP 32                   --
11. --                               --
12. -- @date 2015-09-23               --
13. -----
14.
15. with Data;
16. with Ada.Text_IO; use Ada.Text_IO;
17. with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
18.
19. procedure Lab1 is
20.   package Data3 is new Data(SIZE => 3); use Data3;
21.   MA, MD, MK, MM, MS, MZ: Data3.Matrix;
22.   C, A, B, W, X, T: Data3.Vector;
23.   o : Integer;
24. begin
25.   -- Generating dummy data
26.   GetMatrix(MA);
27.   GetMatrix(MD);
28.   GetMatrix(MK);
29.   GetMatrix(MM);
30.   GetMatrix(MS);
31.   GetMatrix(MZ);
32.
33.   GetVector(A);
34.   GetVector(B);
```

```

35.  GetVector(W);
36.  GetVector(X);
37.
38.  -- Do calc
39.  C := Func1(A, B, MA, MD);
40.  o := Func2(MK, MM);
41.  T := Func3(MS, MZ, W, X);
42.
43.  -- Print results
44.  Put("C = A - B * (MA * MD) = ("); Put(C); Put_Line("");
45.  Put("o = Min(MK * MM) = "); Put(o, 2); New_Line;
46.  Put("T = (MS * MZ) * (W + X) = ("); Put(T); Put_Line("");
47. end Lab1;
48.

```

48 lines: No errors

GNAT 4.5.4

Copyright 1992-2010, Free Software Foundation, Inc.

Compiling: data.adb (source file time stamp: 2015-09-23 20:40:15)

```

1.  with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
2.  with Ada.Text_IO; use Ada.Text_IO;
3.
4.  package body Data is
5.
6.    procedure Fill(filler: in Integer; mtrx: out Matrix) is
7.    begin
8.      for r in mtrx'range(1) loop
9.        for c in mtrx'range(2) loop
10.          mtrx(r, c) := filler;
11.        end loop;
12.      end loop;
13.    end;
14.
15.    procedure Fill(filler: in Integer; vec: out Vector) is
16.    begin
17.      for i in vec'range(1) loop
18.        vec(i) := filler;
19.      end loop;
20.    end;
21.
22.    function "*" (left, right : in Matrix) return Matrix is
23.      temp : Matrix;
24.    begin
25.      Fill(0, temp);
26.      for i in left'range(1) loop
27.        for j in right'range(2) loop
28.          for k in left'range(2) loop
29.            temp(i, j) := temp(i, j) + left(i, k) * right(k, j);
30.          end loop;
31.        end loop;
32.      end loop;
33.      return temp;
34.    end "*";
35.
36.    function "*" (left : in Vector; right : in Matrix) return Vector is
37.      temp : Vector;
38.      helper : Matrix;
39.    begin
40.      Fill(0, helper);
41.      for i in left'range(1) loop
42.        helper(1, i) := left(i);
43.      end loop;

```

```

44.     helper := helper * right;
45.     for i in temp'range(1) loop
46.         temp(i) := helper(1, i);
47.     end loop;
48.     return temp;
49. end "*";
50.
51. function "*" (left : in Matrix; right : in Vector) return Vector is
52.     temp : Vector;
53.     helper : Matrix;
54. begin
55.     Fill(0, helper);
56.     for i in right'range(1) loop
57.         helper(i, 1) := right(i);
58.     end loop;
59.     helper := left * helper;
60.     for i in temp'range(1) loop
61.         temp(i) := helper(i, 1);
62.     end loop;
63.     return temp;
64. end "*";
65.
66. function "+" (left, right : in Vector) return Vector is
67.     temp: Vector;
68. begin
69.     for i in temp'range(1) loop
70.         temp(i) := left(i) + right(i);
71.     end loop;
72.     return temp;
73. end;
74.
75. function "-" (left, right : in Vector) return Vector is
76.     temp: Vector;
77. begin
78.     for i in temp'range(1) loop
79.         temp(i) := left(i) - right(i);
80.     end loop;
81.     return temp;
82. end;
83.
84.
85. function Min(mtrx: in Matrix) return Integer is
86.     min : Integer;
87. begin
88.     min := mtrx(1, 1);
89.     for row in mtrx'range(1) loop
90.         for col in mtrx'range(2) loop
91.             if min > mtrx(row, col) then
92.                 min := mtrx(row, col);
93.             end if;
94.         end loop;
95.     end loop;
96.     return min;
97. end;
98.
99. function Func1(A, B: in Vector; MA, MD: in Matrix) return Vector is
100. begin
101.     return A - B * (MA * MD);
102. end;
103.
104. function Func2(MK, MM: in Matrix) return Integer is
105. begin
106.     return Min(MK * MM);
107. end;
108.
109. function Func3(MS, MZ: in Matrix; W, X: in Vector) return Vector is

```

```

110. begin
111.     return (MS * MZ) * (W + X);
112. end;
113.
114. procedure GetMatrix(mtrx: out Matrix) is
115. begin
116.     Fill(1, mtrx);
117. end;
118.
119. procedure GetVector(vec: out Vector) is
120. begin
121.     Fill(1, vec);
122. end;
123.
124. procedure Put(value: in Matrix) is
125. begin
126.     for row in value'range(1) loop
127.         for col in value'range(2) loop
128.             Put(value(row, col), 4);
129.         end loop;
130.         New_Line;
131.     end loop;
132. end;
133.
134. procedure Put(value: in Vector) is
135. begin
136.     for i in value'range(1) loop
137.         Put(value(i), 4);
138.     end loop;
139. end;
140.
141. end Data;
142.

```

Compiling: data.ads (source file time stamp: 2015-09-23 20:40:30)

```

1. generic
2.   SIZE : in Natural := 2;
3. package Data is
4.   type Vector is private;
5.   type Matrix is private;
6.
7.   function Func1(A, B: in Vector; MA, MD: in Matrix) return Vector;
8.   function Func2(MK, MM: in Matrix) return Integer;
9.   function Func3(MS, MZ: in Matrix; W, X: in Vector) return Vector;
10.
11.   procedure GetMatrix(mtrx: out Matrix);
12.   procedure GetVector(vec: out Vector);
13.
14.   procedure Put(value: in Matrix);
15.   procedure Put(value: in Vector);
16.
17.   private
18.     type Vector is array (1 .. SIZE) of Integer;
19.     type Matrix is array (1 .. SIZE, 1 .. SIZE) of Integer;
20. end Data;
21.

```

142 lines: No errors