```
 1 with Ada.Float_Text_IO;         use Ada.Float_Text_IO;
 2 with Ada.Text_IO;             use Ada.Text_IO;
 3 with Ada.Integer_Text_IO;     use Ada.Integer_Text_IO;
 4
 5 with Ada.Numerics.Elementary_Functions;
 6 use Ada.Numerics.Elementary_Functions;
 7
 8 with Ada.Numerics.Discrete_Random;
 9
10 procedure matrix_inverse is
11
12     type matrix is array(integer range <>, integer range <>) of Long_Long_F
   loat;
13
14     proc_num: integer := 5;
15     dim: constant integer := 10;
16     a: matrix(1..dim, 1..dim);
17
18     procedure fill is
19         subtype Range1_10 is Positive range 1..10;
20         package Rand is
21             new Ada.Numerics.Discrete_Random(Range1_10);
22         seed : Rand.Generator;
23     begin
24         Rand.Reset(seed);
25         for i in 1..dim loop
26             for j in 1..dim loop
27                 a(i,j) := Long_Long_Float(Rand.Random(seed))/100.0;
28                 if i>j then
29                     a(i,j) := 0.0;
30                 end if;
31             end loop;
32         end loop;
33     end fill;
34
35     procedure test(inverted: in matrix) is
36         eps : constant Long_Long_Float := 0.5;
37         fail_flag : boolean;
38         c: matrix(1..dim, 1..dim);
39     begin
40         c := (others => (others => 0.0));
41         for i in 1..dim loop
42             for j in 1..dim loop
43                 for k in 1..dim loop
44                     c(i, j) := c(i, j) + a(i, k) * inverted(k,j);
45                 end loop;
46             end loop;
47         end loop;
48         for i in 1..dim loop
49             for j in 1..dim loop
50                 if(i = j) and abs(c(i, j) - 1.0) > eps then
51                     put("Error("); put(i);
52                     put(j);
53                     put(") =");
54                     put(float(c(i,j)),5,3);
55                     new_line;
56                     fail_flag := true;
57                 end if;
58                 if(i /= j) and abs(c(i, j)) > eps then
59                     put("Error("); put(i);
60                     put(j);
61                     put(") =");
```

```
62                      put(float(c(i,j)),5,3);
63                      new_line;
64                      fail_flag := true;
65                  end if;
66              end loop;
67          end loop;
68          if fail_flag then
69              put("Test Failed");
70          else
71              put("Ok");
72          end if;
73      end test;
74
75      function inv(a: in matrix; proc_num: in integer) return matrix is
76
77          h:integer;
78          inverted: matrix(1..dim, 1..dim);
79
80          task type part is
81              entry set(left, right:in integer);
82          end part;
83
84          parts: array(1..proc_num) of part;
85
86          task body part is
87              l, r: integer;
88              s: Long_Long_Float;
89          begin
90              accept set(left,right: in integer) do
91                  l := left;
92                  r := right;
93              end set;
94              for col in l..r loop
95                  for row in reverse 1 .. col - 1 loop
96                      s := 0.0;
97                      for j in row + 1 .. col loop
98                          s := s + a(row, j)*inverted(j, col);
99                      end loop;
100                     inverted(row, col) := - s*inverted(row, row);
101                 end loop;
102             end loop;
103         end part;
104
105     begin
106         inverted := (others => (others => 0.0));
107         h := dim/proc_num;
108         for i in 1..dim loop
109             inverted(i, i) := 1.0/a(i, i);
110         end loop;
111         for i in 1..proc_num loop
112             parts(i).set((i - 1)*h + 1, i*h);
113         end loop;
114         return(inverted);
115     end inv;
116
117 begin
118     fill;
119     test(inv(a, proc_num));
120 end matrix_inverse;
121
122 --Таракчян  Левон  К5-224
123 --Вывод  программы  :
```

```
124 --Ok
125 --
126 --Или , например :
127 --Error(           1          10) =    5.556E-01
128 --Error(           2           9) =   -1.331E+00
129 --Error(           2          10) =    6.667E-01
130 --Error(           3           9) =   -7.762E-01
131 --Error(           6          10) =    7.778E-01
132 --Error(           8          10) =    8.889E-01
133 --Test Failed
134
```