```
 1  with ADA.Numerics.Elementary_Functions;
 2  use ADA.Numerics.Elementary_Functions;
 3  with Ada.Text_IO;
 4  use Ada.Text_IO;
 5  with Ada.Float_Text_IO;
 6  use Ada.Float_Text_IO;
 7  with Ada.Integer_Text_IO;
 8  use Ada.Integer_Text_IO;
 9
10  with Ada.Numerics.Discrete_Random;
11
12  procedure midmatrmult is
13      flag : boolean := true;
14      dim : constant integer := 10;
15      eps : constant float := 0.001;
16      type vector is array (1..dim) of float;
17      type matrix is array (1..dim) of vector;
18
19      A : matrix;
20      B : matrix;
21
22      ANS1 : matrix;
23      ANS2 : matrix;
24      tmp_1, tmp_2 : vector;
25      oper : integer;
26      delt : float;
27
28      task type calc is
29          entry set_id(id: in integer);
30          entry start_calc;
31          entry end_calc;
32      end calc;
33
34      proc: array (1..dim) of calc;
35
36      task body calc is
37          j: integer;
38      begin
39          accept set_id(id: in integer) do
40              j := id;
41          end set_id;
42          loop
43              select
44                  accept start_calc;
45                  case oper is
46                      when 1 => tmp_2(j) := tmp_1(j) + tmp_2(j);
47                      when 2 => tmp_2(j) := tmp_1(j) * tmp_2(j);
48                          when 3 => tmp_2(j) := tmp_1(1);
49                          when others => null;
50                      end case;
51                  or
52                      accept end_calc;
53                  or
54                      terminate;
55              end select;
56          end loop;
57      end calc;
```

```
58
59      procedure step is
60      begin
61          for id in 1..dim loop
62              proc(id).start_calc;
63          end loop;
64          for id in 1..dim loop
65              proc(id).end_calc;
66          end loop;
67      end step;
68
69      procedure init is
70      begin
71          for id in 1..dim loop
72              proc(id).set_id(id);
73          end loop;
74      end init;
75
76      function "+"(a, b: in vector) return vector is
77      begin
78          tmp_1 := a;
79          tmp_2 := b;
80          oper :=1;
81          step;
82          return tmp_2;
83      end "+";
84
85      function "*"(a, b: in vector) return vector is
86      begin
87          tmp_1 := a;
88          tmp_2 := b;
89          oper := 2;
90          step;
91          return tmp_2;
92      end "*";
93
94      function expn(S : in float) return vector is
95      begin
96          tmp_1(1) := S;
97          oper := 3;
98          step;
99          return tmp_2;
100     end expn;
101
102     procedure matr_init is
103         subtype value is Positive range 1..100;
104         package Rand is
105             new Ada.Numerics.Discrete_Random(value);
106         seed : Rand.Generator;
107     begin
108         Rand.Reset(seed);
109         for row in 1..dim loop
110             for col in 1..dim loop
111                 A(row)(col) := Float(Rand.Random(seed))/1000.0;
112                 B(row)(col) := A(row)(col);
113             end loop;
114         end loop;
```

```
115      end matr_init;
116
117  begin
118      init;
119      matr_init;
120
121      ANS1 := (others => (others => 0.0));
122      ANS2 := (others => (others => 0.0));
123
124      for j in 1..dim loop
125          for k in 1..dim loop
126              ANS1(j) := ANS1(j) + A(k) * expn(B(j)(k));
127          end loop;
128      end loop;
129
130      for i in 1..dim loop
131          for j in 1..dim loop
132              for k in 1..dim loop
133                  ANS2(i)(j) := ANS2(i)(j) + A(i)(k) * B(k)(j);
134              end loop;
135          end loop;
136      end loop;
137
138      for j in 1..dim loop
139          for k in 1..dim loop
140              delt := abs(ANS2(j)(k)) - abs(ANS1(j)(k));
141              if ( abs(delt) > eps)
142              then
143                  flag := false;
144              end if;
145          end loop;
146      end loop;
147
148      if ( flag = false )
149      then
150          put("Some problems with accuracy :");
151          put(eps);
152      else
153          put("OK");
154      end if;
155  end midmatrmult;
156
157  --Минаков Александр
158  --K5-224
159
160  --Вывод программы
161
162  --OK
163  --или
164  --Some problems with accuracy : eps
165  --где eps заданная точность
166
```