```
 1 with Ada.Float_Text_IO;                    use Ada.Float_Text_IO;
 2 with Ada.Text_IO;                        use Ada.Text_IO;
 3
 4 with Ada.Numerics.Elementary_Functions;
 5 use Ada.Numerics.Elementary_Functions;
 6
 7 procedure iter_method is
 8
 9     function f(x: in float) return float is
10     begin
11         return exp(x/2.0) - 1.0;
12     end f;
13     pragma INLINE(f);
14
15     num_points : constant integer := 3;
16
17     type points is array (1..num_points) of float;
18
19     function iter(pt : in points; eps : in float) return float is
20
21         task type calc_task is
22             entry set(i: in integer);
23             entry calculate;
24             entry check;
25         end calc_task;
26
27         type tasks is array (1..num_points) of calc_task;
28
29         processes: tasks;
30         x: points := pt;
31         new_x: points := (others => 0.0);
32
33         task body calc_task is
34             p1, p2:integer;
35         begin
36             accept set(i: in integer) do
37                 p1:=i;
38             end set;
39
40             p2 := (p1 mod num_points) + 1;
41
42             loop
43                 select
44                     accept calculate;
45                     new_x(p1) := x(p1) - f(x(p1))*(x(p1) - x(p2))/(f(x(p1))
    - f(x(p2)));
46                 or
47                     accept check;
48                 or
49                     terminate;
50                 end select;
51             end loop;
52         end calc_task;
53
54         procedure init is
55         begin
56             for i in 1..num_points loop
57                 processes(i).set(i);
58             end loop;
59         end init;
60
61         procedure step is
```

```
62          begin
63              for i in 1..num_points loop
64                  processes(i).calculate;
65              end loop;
66              for i in 1..num_points loop
67                  processes(i).check;
68              end loop;
69          end step;
70
71          d : float;
72          best : integer;
73
74      begin
75          init;
76          loop
77              step;
78              d := Float'Last;
79              for i in 1..num_points loop
80                  if abs(x(i) - new_x(i)) < d then
81                      d := abs(x(i) - new_x(i));
82                      best := i;
83                  end if;
84              end loop;
85              exit when d < eps;
86              x := new_x;
87          end loop;
88          return (x(best) + new_x(best))/2.0;
89      end iter;
90
91      eps : constant float := 1.0E-5;
92
93  begin
94      put("f(x) = exp(x/2.0) - 1.0");
95      new_line;
96      put("x = ");
97      put(iter((-1.5, 1.0, 0.2), eps), 1, 3);
98      put(" +- "); put(eps, 1, 1);
99  end iter_method;
100
101 --Таракчян  Левон  К5-224
102 --Вывод  программы  :
103 --f(x) = exp(x/2.0) - 1.0
104 --x = 5.162E-07 +- 1.0E-05
105
```