```
with ADA.Numerics.Elementary_Functions;
use ADA.Numerics.Elementary_Functions;
with Ada.Text_IO;
use Ada.Text_IO;
with Ada.Float_Text_IO;
use Ada.Float_Text_IO;
with Ada.Integer_Text_IO;
use Ada.Integer_Text_IO;

procedure LineLen is
    n_proc : integer := 110;
    num : float := 20.0;
    symb : integer := 10;
    spaces : integer := 0;

    function Calc(l, r, acc: in float) return float is
        cur: float := 0.0;
        prev: float := 1000.0;
        proc_ar: array (1..n_proc) of float;

        function f(x: in float) return float is
        begin
            return sqrt(4.0 - x * x);
        end f;
        pragma inline(f);

        task type calc is
            entry set_id(i:integer);
            entry start_calc;
            entry end_calc;
        end calc;

        proc: array (1..n_proc) of calc;

        task body calc is
        x, val, proc_l, proc_r, step: float;
        ii: integer;
        begin
            accept set_id (i:integer) do
                ii:= i;
            end set_id;
            proc_l := l + (r - l) / float(n_proc) * float(ii - 1);
            proc_r := l + (r - l) / float(n_proc) * float(ii);
            loop
                select
                    accept start_calc;
                        step := (proc_r - proc_l) / num;
                        x := proc_l;
                        proc_ar(ii) := 0.0;
                        for i in 1..integer(num) loop
                            val := sqrt( ( f(x) - f(x + step) ) ** 2 + step ** 2);
                            proc_ar(ii) := proc_ar(ii) + val;
                            x := x + step;
                        end loop;
                or
                    accept end_calc;
                or
```

```
58                      terminate;
59                  end select;
60              end loop;
61          end calc;
62
63          procedure step is
64          begin
65              proc_ar := (others => 0.0);
66              for id in 1..n_proc loop
67                  proc(id).start_calc;
68              end loop;
69              for id in 1..n_proc loop
70                  proc(id).end_calc;
71              end loop;
72          end step;
73
74          procedure init is
75          begin
76              for id in 1..n_proc loop
77                  proc(id).set_id(id);
78              end loop;
79          end init;
80
81      begin
82          init;
83          while abs(prev - cur) > acc loop
84              prev := cur;
85              step;
86              cur := proc_ar(1);
87              for id in 2..n_proc loop
88                  cur := cur + proc_ar(id);
89              end loop;
90          end loop;
91          return cur;
92      end Calc;
93
94  begin
95      put("f(x) = sqrt(4.0 - x * x) ");
96      new_line;
97      put ("Len = ");
98      put( Calc(-2.0, 2.0, 0.1e-3), spaces, symb);
99  end LineLen;
100
101 --Минаков Александр
102 --K5-224
103
104 --Вывод программы
105
106 --f(x) = sqrt(4.0 - x * x)
107 --Len = 6.2821893692E+00
108
```