```ada
with Ada.Float_Text_IO;
use Ada.Float_Text_IO;
with Ada.Text_IO;
use Ada.Text_IO;
with Ada.Integer_Text_IO;
use Ada.Integer_Text_IO;

with Ada.Numerics.Elementary_Functions;
use Ada.Numerics.Elementary_Functions;

with Ada.Numerics.Discrete_Random;

procedure inverse is

    type matrix is array(integer range <>, integer range <>) of float;

    spaces : constant integer := 2;
    symb : constant integer := 3;
    proc : integer := 5;
    dim : constant integer := 10;
    A : matrix(1..dim, 1..dim);

    procedure matr_init is
        subtype value is Positive range 1..10;
        package Rand is
            new Ada.Numerics.Discrete_Random(value);
        seed : Rand.Generator;
    begin
        Rand.Reset(seed);
        for row in 1..dim loop
            for col in 1..dim loop
                A(row, col) := float(Rand.Random(seed))/1000.0;
                    if row < col then
                        A(row, col) := 0.0;
                    end if;
            end loop;
        end loop;
    end matr_init;

    procedure check(inverted: in matrix) is
        eps : constant float := 0.5;
        flag : boolean;
        C : matrix(1..dim, 1..dim);
    begin
        C := (others => (others => 0.0));
        for row in 1 .. dim loop
            for col in 1 .. dim loop
                for pos in 1 .. dim loop
                    C(row, col) := C(row, col) + A(row, pos) * inverted(pos, col);
                end loop;
            end loop;
        end loop;
        for row in 1 .. dim loop
            for col in 1 .. dim loop
                if ( (row = col) and abs(C(row, col) - 1.0) > eps )
                then
                    put("Element");
```

```
                        put(row);
                        put(col);
                        put(" = ");
                        put(C(row, col), spaces, symb);
                        put(" failed");
                        new_line;
                        flag := true;
                    end if;
                    if(row /= col) and abs(C(row, col)) > eps
                    then
                        put("Element");
                        put(row);
                        put(col);
                        put(" = ");
                        put(C(row, col), spaces, symb);
                        put(" failed");
                        new_line;
                        flag := true;
                    end if;
                end loop;
            end loop;
            if flag then
                put("Some problems with accuracy : ");
                put(eps, spaces, symb);
                new_line;
                put("Test failed");
            else
                put("OK");
            end if;
        end check;

        function inv(A: in matrix; proc: in integer) return matrix is

            h:integer;
            inverted: matrix(1..dim, 1..dim);

            task type part is
                entry set(left, right:in integer);
            end part;

            parts: array(1..proc) of part;

            task body part is
                l, r: integer;
                sum: float;
            begin
                accept set(left,right: in integer)
                do
                    l := left;
                    r := right;
                end set;
                for col in l..r loop
                    for row in col + 1 .. dim loop
                        sum := 0.0;
                        for j in col .. row - 1 loop
                            sum := sum + A(row, j)*inverted(j, col);
                        end loop;
                        inverted(row, row) := 1.0 / a(row, row);
                        inverted(row, col) := - sum * inverted(row, row);
                    end loop;
                end loop;
            end part;

        begin
            inverted := (others => (others => 0.0));
            h := dim/proc;
            for i in 1..dim loop
                inverted(i, i) := 1.0/A(i, i);
            end loop;
            for i in 1..proc loop
                parts(i).set((i - 1)*h + 1, i*h);
            end loop;
            return(inverted);
        end inv;

    begin
        matr_init;
        check(inv(A, proc));
    end inverse;

    --Минаков Александр К5-224
    --Вывод программы:
    --OK
    --
    --Или, например:
    --Element         10           9 =  1.429E+00 failed
    --Some problems with accuracy :  5.000E-01
    --Test failed



```