

Лабораторная работа
по вычислительным методам алгебры на тему:

Нахождение собственных значений методом А.М. Данилевского

Выполнил:
Архангельский И.А.

Проверил:
Кондратюк А.П.

Входные и выходные данные.

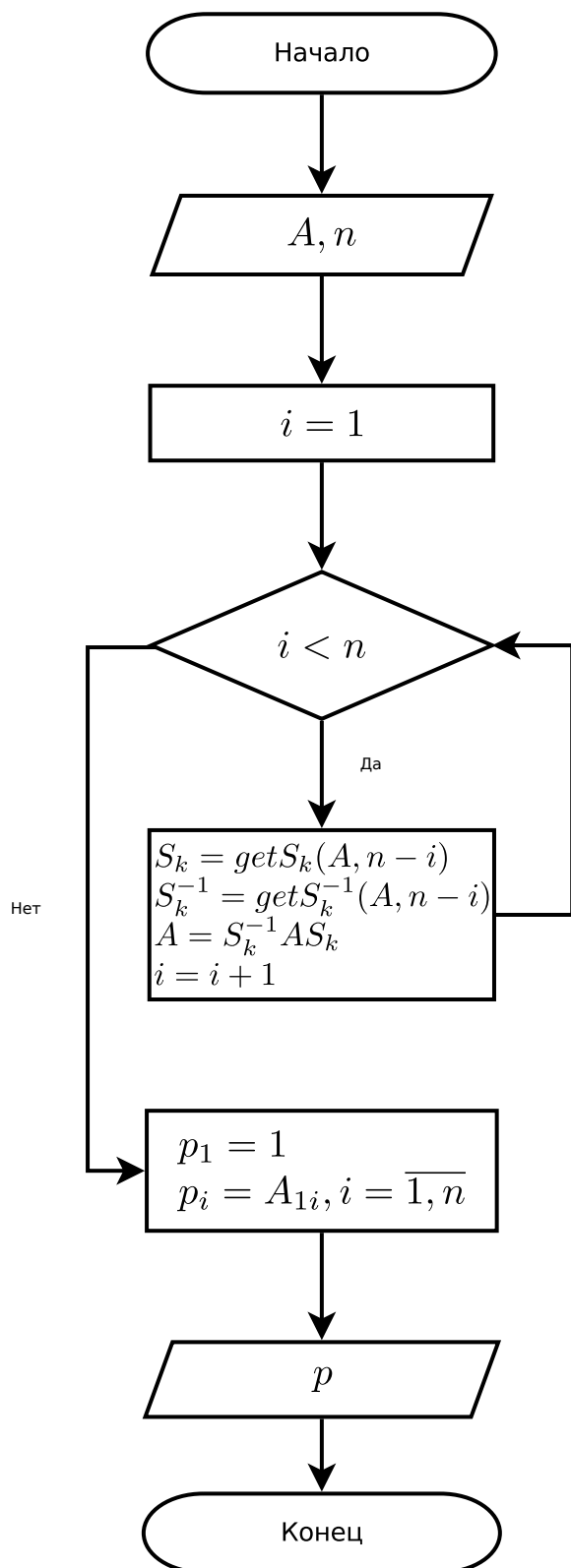
Входные данные

Входной файл содержит размерность матрицы и матрицу A .

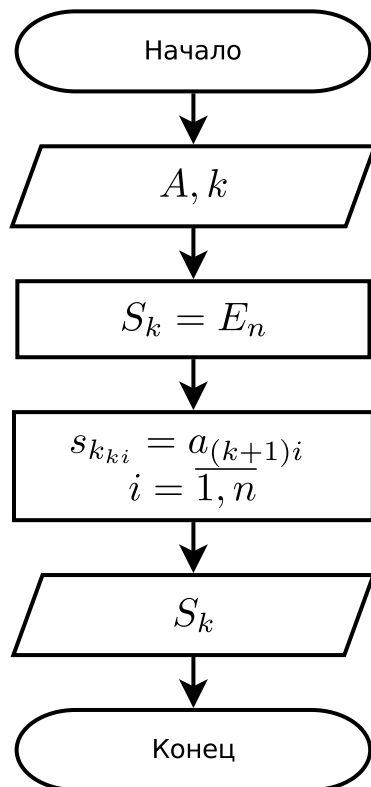
Выходные данные

Программа выводит коэффициенты характеристического многочлена.

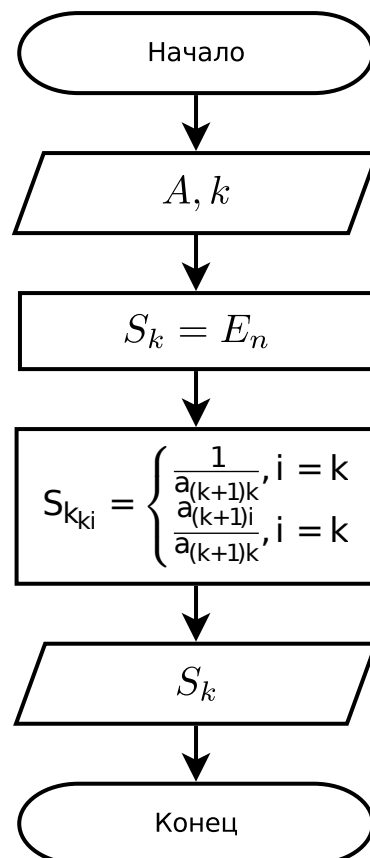
Блок-схема



getS_k⁻¹



getS_k



Реализация

Danilevsky.java

```
1  import java.io.BufferedReader;
2  import java.io.FileReader;
3  import java.io.IOException;
4  import java.util.StringTokenizer;
5
6
7  public class Danilevsky {
8
9      private double[][] matrix;
10     public Danilevsky(String filename) throws IOException
11     {
12         BufferedReader br = new BufferedReader(new FileReader(filename));
13         int dim = Integer.parseInt(br.readLine());
14         matrix = new double[dim][dim];
15         for (int i = 0; i < dim; i++)
16         {
17             StringTokenizer tmp = new StringTokenizer(br.readLine());
18             for (int j = 0; j < dim; j++)
19             {
20                 matrix[i][j] = Double.parseDouble(tmp.nextToken());
21             }
22         }
23     }
24
25     double [][] mulMatrix (double [][] A, double [][] B)
26     {
27         double [][] C = new double [A.length][A.length];
28         for (int i=0;i<C.length;i++)
29         {
30             for (int j=0;j<C.length;j++)
31             {
32                 for (int k=0;k<C.length;k++)
33                 {
34                     C[i][j] += A[i][k]*B[k][j];
35                 }
36             }
37         }
38         return C;
39     }
40
41     private double [][] getSkInv (double [][] A, int k)
42     {
43         double [][] Sk = new double [A.length][A.length];
44         for (int i=0;i<A.length;i++)
45         {
46             Sk[i][i]=1;
47         }
48         for (int i=0;i<A.length;i++)
49         {
50             Sk[k][i]=A[k+1][i];
51         }
52         return Sk;
53     }
54
55     private double [][] getSk (double [][] A, int k)
56     {
57         double [][] Sk = new double [A.length][A.length];
58         for (int i=0;i<A.length;i++)
59         {
60             Sk[i][i]=1;
61         }
62         for (int i=0;i<A.length;i++)
63         {
64             Sk[k][i]=(i==k?1:-A[k+1][i])/A[k+1][k];
65         }
66         return Sk;
67     }
68
69     private double [][] makePhrobenius ()
70     {
71         double [][] phrobenius = new double [matrix.length][matrix.length];
72         for (int i=0;i<matrix.length;i++)
73         {
74             for (int j=0;j<matrix.length;j++)
```

```

75         phrobenius[i][j]=matrix[i][j];
76     }
77     for (int i=0;i<phrobenius.length-1;i++)
78     {
79         double [][] SkInv = getSkInv(phrobenius, phrobenius.length-2-i);
80         double [][] Sk = getSk (phrobenius,phrobenius.length-2-i);
81         phrobenius = mulMatrix(SkInv,mulMatrix(phrobenius, Sk));
82     }
83     return phrobenius;
84 }
85
86 public double [] getPoly ()
87 {
88     double [] poly = new double [matrix.length+1];
89     poly[0]=1;
90     double [][] phrobenius = makePhrobenius();
91     for (int i=0;i<matrix.length;i++)
92     {
93         poly[i+1] = -1*phrobenius[0][i];
94     }
95     return poly;
96 }
97
98 }

```

Тестовые данные

Матрица:

$$A = \begin{pmatrix} -5.509882 & 1.870086 & 0.422908 & 0.008814 \\ 0.287865 & -11.811654 & 5.711900 & 0.058717 \\ 0.049099 & 4.308033 & -12.970687 & 0.229326 \\ 0.006235 & 0.269851 & 1.397369 & -17.596207 \end{pmatrix}$$

Полученный многочлен:

$$\varphi(t) = t^4 + 47.888430t^3 + 797.278765t^2 + 5349.455515t + 12296.550566$$

Корни многочлена(собственные значения):

$$\lambda_1 = -17.8633, \lambda_2 = -17.1524, \lambda_3 = -7.5740, \lambda_4 = -5.2987$$