

Программный модуль 2

1 Описание

Название программы: Генерация случайных квадратных матриц одинаковой размерности и их перемножение.

Дано: Целое положительное число n — размер квадратных матриц.

Требуется: Сгенерировать 2 случайных квадратных матрицы размерности $n \times n$ и перемножить их.

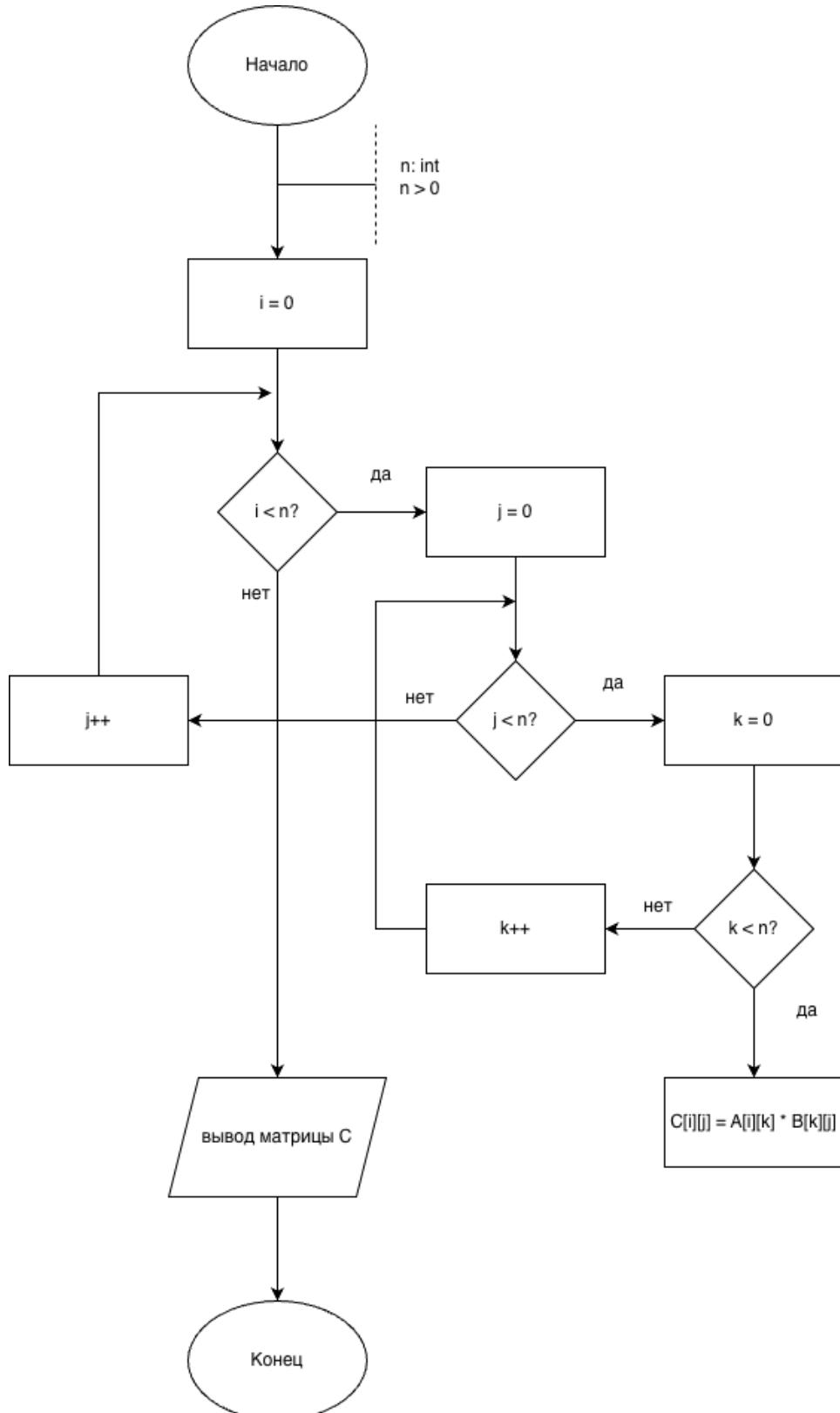
Выходные данные: Выведенная матрица-результат на экран.

Ограничения: $n > 0$.

Спецификация:

№	Вход	Выход	Реакция программы
1	$n > 0$	Вывод матрицы C	Программа генерирует две матрицы A и B, соответственно, после чего последовательно выводит их. Вычисляется результирующая матрица C путем умножения матриц A и B и выводится на экран.
2	$n = 0$	Вывод исключения вида java.lang.ArrayIndexOutOfBoundsException	Программа обнаруживает ошибку и «выбрасывает» исключение.
3	$n < 0$	Вывод исключения вида java.lang.NegativeArraySizeException	Программа обнаруживает ошибку и «выбрасывает» исключение.

2 Блок-схема



3 Исходный код

```
1 import java.util.Random;
2
3 public class MatrixMult {
4
5     // Метод генерации случайной матрицы
6     public static double[][] generateMatrix(int rows, int cols) {
7         double[][] matrix = new double[rows][cols];
8         Random rand = new Random();
9         for (int i = 0; i < rows; i++) {
10             for (int j = 0; j < cols; j++) {
11                 matrix[i][j] = rand.nextDouble();
12             }
13         }
14         return matrix;
15     }
16
17     // Проверка, что матрица не null
18     public static boolean isNotNull(double[][] matrix) {
19         return matrix != null;
20     }
21
22     // Простой метод умножения матриц
23     public static double[][] multiply(double[][] firstMatrix, double[][] secondMatrix) {
24         int n1 = firstMatrix.length;
25         int m1 = firstMatrix[0].length;
26         int n2 = secondMatrix.length;
27         int m2 = secondMatrix[0].length;
28
29         if (m1 != n2) {
30             throw new IllegalArgumentException("Размерности матриц не совпадают для умножения");
31         }
32
33         double[][] result = new double[n1][m2];
34
35         for (int i = 0; i < n1; i++) {
36             for (int j = 0; j < m2; j++) {
37                 double sum = 0;
38                 for (int k = 0; k < m1; k++) {
39                     sum += firstMatrix[i][k] * secondMatrix[k][j];
40                 }
41                 result[i][j] = sum;
42             }
43         }
44         return result;
45     }
46
47     public static void printMatrix(double[][] A) {
48         if (isNotNull(A)) {
49             for (int i = 0; i < A.length; i++) {
50                 for (int j = 0; j < A[0].length; j++) {
51                     System.out.print(A[i][j] + "\t");
52                 }
53                 System.out.println();
54             }
55         } else {
56             System.out.println("It is null matrix!");
57         }
58         System.out.println();
59     }
60 }
```

```
61
62     public static void main(String[] args) {
63         int n = 5;
64
65         double [][] A = generateMatrix(n, n);
66         double [][] B = generateMatrix(n, n);
67
68         System.out.println("First matrix:");
69         printMatrix(A);
70
71         System.out.println("Second matrix:");
72         printMatrix(B);
73
74         double [][] C = multiply(A, B);
75
76         System.out.println("Multiply matrix:");
77         printMatrix(C);
78
79     }
80
81 }
```