

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра программных систем

**Задание №1**

**Работа с массивами. Матрицы.**

Выполнил: Мороз

Илья Олегович

Группа № K3220

Преподаватель: Иванов Сергей Евгеньевич

Санкт-Петербург

2020

**Задание:** для квадратной матрицы nxn, размер который задается, с помощью массивов реализовать методы сложения, перемножения, нахождения обратной матрицы.

**Исходный код:**

using System;  
  
namespace task1  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 */\*  
 \* Enter size of matrix  
 \*/* Console.Write("Enter n (size of matrix): ");  
 int n = Int32.Parse(Console.ReadLine());  
  
 */\*  
 \* Enter matrices  
 \*/* Console.WriteLine("Enter matrix #1:");  
 double[,] matrix1 = InputMatrix(n);  
 Console.WriteLine("Enter matrix #2:");  
 double[,] matrix2 = InputMatrix(n);  
  
 */\*  
 \* Clear console  
 \*/* Console.Clear();  
  
 */\*  
 \* Write matrix #1 to console  
 \*/* Console.WriteLine("Matrix #1:");  
 PrintMatrix(matrix1);  
  
 */\*  
 \* Write matrix #2 to console  
 \*/* Console.WriteLine("Matrix #2:");  
 PrintMatrix(matrix2);  
  
 */\*  
 \* Sum matrices and print result  
 \*/* Console.WriteLine("Matrix sum:");  
 PrintMatrix(SumMatrices(matrix1, matrix2));  
  
 */\*  
 \* Multiply matrices and print result  
 \*/* Console.WriteLine("Matrix multiply:");  
 PrintMatrix(MultiplyMatrices(matrix1, matrix2));  
  
 */\*  
 \* Invert matrices and print them  
 \*/* Console.WriteLine("Inverse matrix #1:");  
 PrintMatrix(InverseMatrix(matrix1));  
 Console.WriteLine("Inverse matrix #2:");  
 PrintMatrix(InverseMatrix(matrix2));  
 }  
  
 */\*  
 \* Read elements from console and return matrix  
 \*/* private static double[,] InputMatrix(int n)  
 {  
 double[,] matrix = new double[n, n];  
 for (int i = 0; i < n; i++)  
 {  
 for (int j = 0; j < n; j++)  
 {  
 matrix[i, j] = Double.Parse(Console.ReadLine());  
 }  
 }  
  
 return matrix;  
 }  
  
 */\*  
 \* Print matrix in console  
 \*/* private static void PrintMatrix(double[,] matrix)  
 {  
 for (int i = 0; i < matrix.GetLength(0); i++)  
 {  
 for (int j = 0; j < matrix.GetLength(1); j++)  
 {  
 Console.Write(matrix[i, j] + "\t");  
 }  
  
 Console.WriteLine();  
 }  
 }  
  
 */\*  
 \* Sum two matrices and return result  
 \*/* private static double[,] SumMatrices(double[,] matrix1, double[,] matrix2)  
 {  
 int matrixSize = matrix1.GetLength(0);  
 double[,] result = new double[matrixSize, matrixSize];  
 for (int i = 0; i < matrixSize; i++)  
 {  
 for (int j = 0; j < matrixSize; j++)  
 {  
 result[i, j] = matrix1[i, j] + matrix2[i, j];  
 }  
 }  
  
 return result;  
 }  
  
 */\*  
 \* Multiply two matrices and return result  
 \*/* private static double[,] MultiplyMatrices(double[,] matrix1, double[,] matrix2)  
 {  
 int n = matrix1.GetLength(0);  
 double[,] result = new double[n, n];  
 */\*  
 \* Row of first matrix  
 \*/* for (int k = 0; k < n; k++)  
 {  
 */\*  
 \* Column of second matrix  
 \*/* for (int i = 0; i < n; i++)  
 {  
 double element = 0;  
 */\*  
 \* Column of first matrix and row of second matrix  
 \*/* for (int j = 0; j < n; j++)  
 {  
 element += matrix1[k, j] \* matrix2[j, i];  
 }  
  
 result[k, i] = element;  
 }  
 }  
  
 return result;  
 }  
  
 */\*  
 \* Return algebraic complement for element at position [row, column]  
 \*/* private static double[,] AlgebraicComplement(double[,] matrix, int row, int column)  
 {  
 int n = matrix.GetLength(0);  
 int x = 0, y = 0;  
 double[,] result = new double[n - 1, n - 1];  
 for (int i = 0; i < n; i++)  
 {  
 */\*  
 \* Skip element row  
 \*/* if (i == row) continue;  
  
 for (int j = 0; j < n; j++)  
 {  
 */\*  
 \* Skip element column  
 \*/* if (j == column) continue;  
 result[x, y] = matrix[i, j];  
 y++;  
 }  
  
 x++;  
 }  
  
 return result;  
 }  
  
 */\*  
 \* Return matrix determinant  
 \*/* private static double MatrixDeterminant(double[,] matrix)  
 {  
 double result = 0;  
 int n = matrix.GetLength(0);  
 if (n == 1) return matrix[0, 0];  
 if (n == 2) return matrix[0, 0] \* matrix[1, 1] - matrix[0, 1] \* matrix[1, 0];  
 for (int i = 0; i < matrix.GetLength(0); i++)  
 {  
 result += Math.Pow(-1, i + 2) \* matrix[i, 0] \* MatrixDeterminant(AlgebraicComplement(matrix, i, 0));  
 }  
  
 return result;  
 }  
  
 */\*  
 \* Return inverse matrix  
 \*/* private static double[,] InverseMatrix(double[,] matrix)  
 {  
 int n = matrix.GetLength(0);  
 double[,] result = new double[n, n];  
 double[,] transposedMatrix = new double[n, n];  
  
 for (int i = 0; i < n; i++)  
 {  
 for (int j = 0; j < n; j++)  
 {  
 transposedMatrix[i, j] = matrix[j, i];  
 }  
 }  
  
 for (int i = 0; i < n; i++)  
 {  
 for (int j = 0; j < n; j++)  
 {  
 result[i, j] = Math.Pow(-1, i + j + 2) \*  
 MatrixDeterminant(AlgebraicComplement(transposedMatrix, i, j)) /  
 MatrixDeterminant(matrix);  
 }  
 }  
  
 return result;  
 }  
 }  
}

**Результат работы:**

****

**Вывод:** с помощью двумерных массивов были реализованы действия с матрицами: сумма, произведение, получение обратной матрицы.

Исходный код также доступен на GitHub: [ilyamore88/applied-informatics/task1](https://github.com/ilyamore88/applied-informatics/tree/master/task1).