

Теоретический материал

Массив представляет набор однотипных данных. Объявление массива похоже на объявление переменной за тем исключением, что после указания типа ставятся квадратные скобки:

```
тип_переменной[] название_массива;
```

Определим массив целых чисел:

```
int[] numbers;
```

После определения переменной массива можно присвоить ей определенное значение:

```
int[] nums = new int[4];
```

Также мы сразу можем указать значения для этих элементов:

```
int[] nums2 = new int[4] { 1, 2, 3, 5 };
```

```
int[] nums3 = new int[] { 1, 2, 3, 5 };
```

```
int[] nums4 = new[] { 1, 2, 3, 5 };
```

```
int[] nums5 = { 1, 2, 3, 5 };
```

Все перечисленные выше способы будут равноценны.

Начиная с версии C# 12 для определения массивов можно использовать выражения коллекций, которые предполагают заключение элементов массива в квадратные скобки:

```
int[] nums1 = [ 1, 2, 3, 5 ];
```

```
int[] nums2 = []; // пустой массив
```

Для обращения к элементам массива используются индексы. Индекс представляет номер элемента в массиве, при этом нумерация начинается с нуля, поэтому индекс первого элемента будет равен 0, индекс четвертого элемента - 3.

Используя индексы, можно, как получить элементы массива:

```
int[] numbers = { 1, 2, 3, 5 };
```

```
Console.WriteLine(numbers[3]);
```

```
//получение эл-та массива 5
```

Так и изменить элемент массива по индексу:

```
numbers[1] = 505;
```

```
Console.WriteLine(numbers[1]); // 505
```

Каждый массив имеет свойство `Length`, которое хранит длину массива. Например, получим длину массива `numbers`:

```
int[] numbers = { 1, 2, 3, 5 };
```

```
Console.WriteLine(numbers.Length); // 4
```

Для перебора массивов можно использовать различные типы циклов. Например, цикл **foreach**:

```
int[] numbers = { 1, 2, 3, 4, 5 };
```

```
foreach (int i in numbers)
```

```
{  
  
    Console.WriteLine(i);  
  
}
```

Аналогично подобные действия можно сделать и с помощью цикла **for**:

```
int[] numbers = { 1, 2, 3, 4, 5 };  
  
for (int i = 0; i < numbers.Length; i++)  
  
{  
  
    Console.WriteLine(numbers[i]);  
  
}
```

Также можно использовать и другие виды циклов, например, **while**:

```
int[] numbers = { 1, 2, 3, 4, 5 };  
  
int i = 0;  
  
while(i < numbers.Length)  
  
{  
  
    Console.WriteLine(numbers[i]);  
  
    i++;  
  
}
```

Массивы, которые имеют два измерения (ранг равен 2) называют двухмерными. Например, создадим одномерный и двухмерный массивы, которые имеют одинаковые элементы:

```
int[] nums1 = new int[] { 0, 1, 2, 3, 4, 5 };
```

```
int[,] nums2 = { { 0, 1, 2 }, { 3, 4, 5 } };
```

Для генерации случайных чисел в программах, написанных на C#, предназначен класс «Random».

```
//Создание объекта для генерации чисел
```

```
Random rnd = new Random(245);
```

```
//Получить случайное число (в диапазоне от 0 до 10)
```

```
int value = rnd.Next(0, 10);
```

```
//Вывод числа в консоль
```

```
Console.WriteLine(value);
```

Задание 1

Задача:

Калькулятор матриц

Реализуйте программный продукт средствами языка C# со следующим функционалом:

- 1) Создание двух матриц размерности $n \times m$ (значения n и m вводятся с клавиатуры);
- 2) Заполнение матриц значениями с клавиатуры (по выбору пользователя, с последующим выводом результата на экран);
- 3) Заполнение матриц случайными числами в диапазоне $[a; b]$ (значения a и b вводятся с клавиатуры) (по выбору пользователя, с последующим выводом результата на экран);
- 4) Сложение матриц (предусмотреть проверку на возможность выполнения операции, с последующим выводом результата на экран);
- 5) Умножение матриц (предусмотреть проверку на возможность выполнения операции, с последующим выводом результата на экран);
- 6) Нахождение детерминанта (определителя) матрицы (предусмотреть проверку на возможность выполнения операции, с последующим выводом результата на экран);
- 7) Нахождение обратной матрицы (предусмотреть проверку на возможность выполнения операции, с последующим выводом результата на экран);
- 8) Транспонирование матриц (с последующим выводом результата на экран);
- 9) Нахождение корней системы уравнений, заданных матрицей (с последующим выводом результата на экран).

При тестировании продемонстрировать успешное выполнение всех пунктов (положительный сценарий), а также обработку следующих ситуаций (негативный сценарий):

- 1) Невозможность сложения матриц по причине несоответствия их размерностей;
- 2) Невозможность умножения матриц в связи с их несовместимостью;
- 3) Невозможность нахождения детерминанта у не квадратных матриц ($n \neq m$);
- 4) Невозможность нахождения обратной матрицы в случае, если детерминант равен нулю ($d=0$);
- 5) Невозможность нахождения корней систему уравнений, если она не имеет решения или не имеет однозначного решения.

Весь функционал должен быть реализован вами, программы, разработанные с использованием сторонних решений (библиотеки, фреймворки и т.д.) реализующих функционал, приниматься не будут.

Решение:

```
using System;

class MatrixCalculator
{
    static double[,] matrixA, matrixB;

    static void Main()
    {
        while (true)
        {
            Console.WriteLine("\nВыберите операцию:");
            Console.WriteLine("1. Создать матрицу A");
            Console.WriteLine("2. Создать матрицу B");
            Console.WriteLine("3. Заполнить матрицу с клавиатуры");
            Console.WriteLine("4. Заполнить матрицу случайными числами");
```

```

Console.WriteLine("5. Сложение матриц");
Console.WriteLine("6. Умножение матриц");
Console.WriteLine("7. Найти определитель матрицы");
Console.WriteLine("8. Найти обратную матрицу");
Console.WriteLine("9. Транспонировать матрицу");
Console.WriteLine("10. Решить систему уравнений");
Console.WriteLine("0. Выход");

int choice = int.Parse(Console.ReadLine());

if (choice == 0) break;

switch (choice)
{
    case 1:
        CreateMatrix(ref matrixA, "A");
        break;
    case 2:
        CreateMatrix(ref matrixB, "B");
        break;
    case 3:
        FillMatrixManually();
        break;
    case 4:
        FillMatrixRandomly();
        break;
    case 5:
        AddMatrices();
        break;
    case 6:
        MultiplyMatrices();
        break;
    case 7:
        FindDeterminant();
        break;
    case 8:
        FindInverseMatrix();
        break;
    case 9:
        TransposeMatrix();
        break;
    case 10:
        SolveEquationSystem();
        break;
}

```

```

        default:
            Console.WriteLine("Неверный выбор. Попробуйте снова.");
            break;
        }
    }
}

static void CreateMatrix(ref double[,] matrix, string name)
{
    Console.Write($"Введите количество строк для матрицы {name}: ");
    int n = int.Parse(Console.ReadLine());
    Console.Write($"Введите количество столбцов для матрицы {name}: ");
    int m = int.Parse(Console.ReadLine());

    matrix = new double[n, m];
    Console.WriteLine($"Матрица {name} создана.");
}

static void FillMatrixManually()
{
    Console.WriteLine("Выберите матрицу для заполнения (А или В):");
    string choice = Console.ReadLine().ToUpper();
    double[,] matrix = choice == "А" ? matrixA : matrixB;

    if (matrix == null)
    {
        Console.WriteLine("Матрица не создана.");
        return;
    }

    for (int i = 0; i < matrix.GetLength(0); i++)
    {
        for (int j = 0; j < matrix.GetLength(1); j++)
        {
            Console.Write($"Элемент [{i}, {j}]: ");
            matrix[i, j] = double.Parse(Console.ReadLine());
        }
    }

    Console.WriteLine("Матрица заполнена.");
}

static void FillMatrixRandomly()

```



```

{
    Console.WriteLine("Выберите матрицу для заполнения (А или В):");
    string choice = Console.ReadLine().ToUpper();
    double[,] matrix = choice == "А" ? matrixA : matrixB;

    if (matrix == null)
    {
        Console.WriteLine("Матрица не создана.");
        return;
    }

    Console.Write("Введите минимальное значение: ");
    int min = int.Parse(Console.ReadLine());
    Console.Write("Введите максимальное значение: ");
    int max = int.Parse(Console.ReadLine());

    Random rand = new Random();

    for (int i = 0; i < matrix.GetLength(0); i++)
    {
        for (int j = 0; j < matrix.GetLength(1); j++)
        {
            matrix[i, j] = rand.Next(min, max + 1);
        }
    }

    Console.WriteLine("Матрица заполнена случайными числами.");
    PrintMatrix(matrix, $"Матрица {choice}:");
}

static void AddMatrices()
{
    if (matrixA == null || matrixB == null)
    {
        Console.WriteLine("Сначала создайте обе матрицы.");
        return;
    }

    if (matrixA.GetLength(0) != matrixB.GetLength(0) ||
matrixA.GetLength(1) != matrixB.GetLength(1))
    {
        Console.WriteLine("Невозможно сложить матрицы разных
размеров.");
        return;
    }
}

```

```

    }

    double[,] result = new double[matrixA.GetLength(0),
matrixA.GetLength(1)];

    for (int i = 0; i < matrixA.GetLength(0); i++)
    {
        for (int j = 0; j < matrixA.GetLength(1); j++)
        {
            result[i, j] = matrixA[i, j] + matrixB[i, j];
        }
    }

    PrintMatrix(result, "Результат сложения:");
}

static void MultiplyMatrices()
{
    if (matrixA == null || matrixB == null)
    {
        Console.WriteLine("Сначала создайте обе матрицы.");
        return;
    }

    if (matrixA.GetLength(1) != matrixB.GetLength(0))
    {
        Console.WriteLine("Невозможно умножить матрицы. Количество
столбцов первой матрицы должно быть равно количеству строк второй
матрицы.");
        return;
    }

    double[,] result = new double[matrixA.GetLength(0),
matrixB.GetLength(1)];

    for (int i = 0; i < matrixA.GetLength(0); i++)
    {
        for (int j = 0; j < matrixB.GetLength(1); j++)
        {
            for (int k = 0; k < matrixA.GetLength(1); k++)
            {
                result[i, j] += matrixA[i, k] * matrixB[k, j];
            }
        }
    }
}

```

```

    }

    PrintMatrix(result, "Результат умножения:");
}

static void FindDeterminant()
{
    Console.WriteLine("Выберите матрицу (А или В):");
    string choice = Console.ReadLine().ToUpper();
    double[,] matrix = choice == "А" ? matrixA : matrixB;

    if (matrix == null)
    {
        Console.WriteLine("Матрица не создана.");
        return;
    }

    if (matrix.GetLength(0) != matrix.GetLength(1))
    {
        Console.WriteLine("Определитель можно найти только для квадратной матрицы.");
        return;
    }

    double det = CalculateDeterminant(matrix);
    Console.WriteLine($"Определитель матрицы: {det}");
}

static double CalculateDeterminant(double[,] matrix)
{
    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];

    double det = 0;
    for (int j = 0; j < n; j++)
    {
        det += Math.Pow(-1, j) * matrix[0, j] *
CalculateDeterminant(GetMinor(matrix, 0, j));
    }
    return det;
}

static double[,] GetMinor(double[,] matrix, int row, int col)

```

```

{
    int n = matrix.GetLength(0);
    double[,] minor = new double[n - 1, n - 1];
    int r = 0, c = 0;

    for (int i = 0; i < n; i++)
    {
        if (i == row) continue;
        c = 0;
        for (int j = 0; j < n; j++)
        {
            if (j == col) continue;
            minor[r, c] = matrix[i, j];
            c++;
        }
        r++;
    }
    return minor;
}

static void FindInverseMatrix()
{
    Console.WriteLine("Выберите матрицу (A или B):");
    string choice = Console.ReadLine().ToUpper();
    double[,] matrix = choice == "A" ? matrixA : matrixB;

    if (matrix == null)
    {
        Console.WriteLine("Матрица не создана.");
        return;
    }

    if (matrix.GetLength(0) != matrix.GetLength(1))
    {
        Console.WriteLine("Обратную матрицу можно найти только для квадратной матрицы.");
        return;
    }
    double det = CalculateDeterminant(matrix);
    if (Math.Abs(det) < 1e-10)
    {
        Console.WriteLine("Невозможно найти обратную матрицу, так как определитель равен нулю.");
        return;
    }
}

```

```

    }

    int n = matrix.GetLength(0);
    double[,] inverse = new double[n, n];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            inverse[j, i] = Math.Pow(-1, i + j) *
CalculateDeterminant(GetMinor(matrix, i, j)) / det;
        }
    }

    PrintMatrix(inverse, "Обратная матрица:");
}

static void TransposeMatrix()
{
    Console.WriteLine("Выберите матрицу (А или В):");
    string choice = Console.ReadLine().ToUpper();
    double[,] matrix = choice == "А" ? matrixA : matrixB;

    if (matrix == null)
    {
        Console.WriteLine("Матрица не создана.");
        return;
    }

    int rows = matrix.GetLength(0);
    int cols = matrix.GetLength(1);
    double[,] transposed = new double[cols, rows];

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            transposed[j, i] = matrix[i, j];
        }
    }

    PrintMatrix(transposed, "Транспонированная матрица:");
}

```

```

static void SolveEquationSystem()
{
    if (matrixA == null)
    {
        Console.WriteLine("Сначала создайте матрицу коэффициентов
(A).");
        return;
    }

    int n = matrixA.GetLength(0);
    if (n != matrixA.GetLength(1))
    {
        Console.WriteLine("Матрица коэффициентов должна быть
квадратной.");
        return;
    }

    Console.WriteLine("Введите свободные члены:");
    double[] b = new double[n];
    for (int i = 0; i < n; i++)
    {
        Console.Write($"b[{i}]: ");
        b[i] = double.Parse(Console.ReadLine());
    }

    // Вывод системы уравнений
    PrintEquationSystem(matrixA, b);

    double det = CalculateDeterminant(matrixA);
    if (Math.Abs(det) < 1e-10)
    {
        Console.WriteLine("Система уравнений не имеет единственного
решения, так как определитель матрицы коэффициентов равен нулю.");
        return;
    }

    double[] x = new double[n];
    for (int i = 0; i < n; i++)
    {
        double[,] tempMatrix = (double[,])matrixA.Clone();
        for (int j = 0; j < n; j++)
        {
            tempMatrix[j, i] = b[j];
        }
    }
}

```

```

        x[i] = CalculateDeterminant(tempMatrix) / det;
    }

    Console.WriteLine("Решение системы уравнений:");
    for (int i = 0; i < n; i++)
    {
        Console.WriteLine($"x[{i}] = {x[i]}");
    }
}

static void PrintEquationSystem(double[,] matrix, double[] b)
{
    Console.WriteLine("Система уравнений:");
    for (int i = 0; i < matrix.GetLength(0); i++)
    {
        for (int j = 0; j < matrix.GetLength(1); j++)
        {
            Console.Write($"{matrix[i, j]}*x{j + 1}");
            if (j < matrix.GetLength(1) - 1)
            {
                Console.Write(" + ");
            }
        }
        Console.WriteLine($" = {b[i]}");
    }
}

static void PrintMatrix(double[,] matrix, string title = "")
{
    Console.WriteLine(title);
    int rows = matrix.GetLength(0);
    int cols = matrix.GetLength(1);

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            Console.Write($"{matrix[i, j],8:F2} ");
        }
        Console.WriteLine();
    }
}
}

```

Ответ:

1)

```
C:\Users\tuchi\source\repos\l  X + v
```

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход
1
Введите количество строк для матрицы A: 3
Введите количество столбцов для матрицы A: 4
Матрица A создана.

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход
2
Введите количество строк для матрицы B: 5
Введите количество столбцов для матрицы B: 4
Матрица B создана.

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход
4
Выберите матрицу для заполнения (A или B):
A
Введите минимальное значение: 5
Введите максимальное значение: 10
Матрица заполнена случайными числами.
Матрица A:

7,00	6,00	8,00	7,00
8,00	8,00	6,00	5,00
6,00	10,00	10,00	7,00

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами

Выберите операцию:

1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

4

Выберите матрицу для заполнения (A или B):

B

Введите минимальное значение: 3

Введите максимальное значение: 7

Матрица заполнена случайными числами.

Матрица B:

5,00	6,00	6,00	3,00
3,00	6,00	5,00	5,00
4,00	5,00	5,00	3,00
5,00	5,00	3,00	7,00
6,00	3,00	6,00	6,00

Выберите операцию:

1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

5

Невозможно сложить матрицы разных размеров.

Выберите операцию:

1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

2)

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

6

Невозможно умножить матрицы. Количество столбцов первой матрицы должно быть равно количеству строк второй матрицы.

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

3)

Выберите операцию:

1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

1

Введите количество строк для матрицы A: 3

Введите количество столбцов для матрицы A: 4

Матрица A создана.

Выберите операцию:

1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

4

Выберите матрицу для заполнения (A или B):

A

Введите минимальное значение: 1

Введите максимальное значение: 7

Матрица заполнена случайными числами.

Матрица A:

3,00	2,00	5,00	3,00
1,00	4,00	1,00	4,00
1,00	3,00	2,00	3,00

Выберите операцию:

1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

7

Выберите матрицу (A или B):

A

Определитель можно найти только для квадратной матрицы.

Выберите операцию:

1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу

4)

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход
1
Введите количество строк для матрицы A: 3
Введите количество столбцов для матрицы A: 3
Матрица A создана.

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход
3

Выберите матрицу для заполнения (A или B):

A

Элемент [0,0]: 4

Элемент [0,1]: 2

Элемент [0,2]: -3

Элемент [1,0]: -8

Элемент [1,1]: -7

Элемент [1,2]: 1

Элемент [2,0]: 4

Элемент [2,1]: 2

Элемент [2,2]: -3

Матрица заполнена.

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход
7

Выберите матрицу (A или B):

A

Определитель матрицы: 0

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход
8

Выберите матрицу (A или B):

A

Невозможно найти обратную матрицу, так как определитель равен нулю.

Выберите операцию:
1. Создать матрицу A
2. Создать матрицу B
3. Заполнить матрицу с клавиатуры
4. Заполнить матрицу случайными числами
5. Сложение матриц
6. Умножение матриц
7. Найти определитель матрицы
8. Найти обратную матрицу
9. Транспонировать матрицу
10. Решить систему уравнений
0. Выход

Выберите операцию:
 1. Создать матрицу A
 2. Создать матрицу B
 3. Заполнить матрицу с клавиатуры
 4. Заполнить матрицу случайными числами
 5. Сложение матриц
 6. Умножение матриц
 7. Найти определитель матрицы
 8. Найти обратную матрицу
 9. Транспонировать матрицу
 10. Решить систему уравнений
 0. Выход
 1
 Введите количество строк для матрицы A: 2
 Введите количество столбцов для матрицы A: 2
 Матрица A создана.

Выберите операцию:
 1. Создать матрицу A
 2. Создать матрицу B
 3. Заполнить матрицу с клавиатуры
 4. Заполнить матрицу случайными числами
 5. Сложение матриц
 6. Умножение матриц
 7. Найти определитель матрицы
 8. Найти обратную матрицу
 9. Транспонировать матрицу
 10. Решить систему уравнений
 0. Выход
 3
 Выберите матрицу для заполнения (A или B):
 A
 Элемент [0, 0]: 1
 Элемент [0, 1]: 2
 Элемент [1, 0]: 2
 Элемент [1, 1]: 4
 Матрица заполнена.

Выберите операцию:
 1. Создать матрицу A
 2. Создать матрицу B
 3. Заполнить матрицу с клавиатуры
 4. Заполнить матрицу случайными числами
 5. Сложение матриц
 6. Умножение матриц
 7. Найти определитель матрицы
 8. Найти обратную матрицу
 9. Транспонировать матрицу
 10. Решить систему уравнений
 0. Выход
 10
 Введите свободные члены:
 b[0]: 4
 b[1]: 8
 Система уравнений:
 $1 \cdot x_1 + 2 \cdot x_2 = 4$
 $2 \cdot x_1 + 4 \cdot x_2 = 8$
 Система уравнений не имеет единственного решения, так как определитель матрицы коэффициентов равен нулю.

Выберите операцию:
 1. Создать матрицу A
 2. Создать матрицу B
 3. Заполнить матрицу с клавиатуры
 4. Заполнить матрицу случайными числами
 5. Сложение матриц
 6. Умножение матриц
 7. Найти определитель матрицы