

### 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 Program {
    static void Main(string[] args){
        int n1, m1, n2, m2;
        Console.WriteLine("Введите кол-во строк для первой матрицы: ");
        n1 = int.Parse(Console.ReadLine());
        Console.WriteLine("Введите кол-во столбцов для первой матрицы: ");
        m1 = int.Parse(Console.ReadLine());
        Console.WriteLine("Введите кол-во строк для второй матрицы: ");
        n2 = int.Parse(Console.ReadLine());
        Console.WriteLine("Введите кол-во столбцов для второй матрицы: ");
        m2 = int.Parse(Console.ReadLine());

        double[,] matr1 = new double[n1, m1];
        double[,] matr2 = new double[n2, m2];

        Console.WriteLine("Выберите способ заполнения: ");
        Console.WriteLine("1. Свои значения");
        Console.WriteLine("2. Рандомные числа");
```

```

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

if (choice == 1){
    ReadyMatr(matr1, "Первая матрица");
    ReadyMatr(matr2, "Вторая матрица");
} else if (choice == 2){
    Console.WriteLine("Введите диапазон [a; b] для случайных чисел:");

    Console.WriteLine("Введите a: ");
    double a = double.Parse(Console.ReadLine());
    Console.WriteLine("Введите b: ");
    double b = double.Parse(Console.ReadLine());
    ReadyMatrRandom(matr1, a, b);
    ReadyMatrRandom(matr2, a, b);
}

PrintMatr(matr1, "Первая матрица");
PrintMatr(matr2, "Вторая матрица");

while (true){
    Console.WriteLine("Выберите действие: ");
    Console.WriteLine("1. Сложение");
    Console.WriteLine("2. Умножение");
    Console.WriteLine("3. Нахождение определителя");
    Console.WriteLine("4. Нахождение обратной матрицы");
    Console.WriteLine("5. Транспонирование");
    Console.WriteLine("6. Нахождение корней системы уравнений, заданных матрицей");
    Console.WriteLine("7. Выход");
    int Funchoice = int.Parse(Console.ReadLine());

    switch (Funchoice){
        case 1:
            if (n1 == n2 && m1 == m2){
                var sum = AddMatr(matr1, matr2);
                PrintMatr(sum, "Сумма матриц");
            } else {
                Console.WriteLine("Сложение возможно только для матриц одинакового размера");
            }
            break;

        case 2:
            if (m1 == n2){
                var mul = MulMatr(matr1, matr2);
                PrintMatr(mul, "Умножение матриц");
            } else {
                Console.WriteLine("Умножение возможно только если количество столбцов первой матрицы равно количеству строк второй");
            }
            break;
    }
}

```

```

        case 3:
            if (n1 == m1){
                Console.WriteLine($"Детерминант первой матрицы:
{Deter(matr1)}");
            } else {
                Console.WriteLine("Детерминант можно вычислить только
для квадратных матриц");
            }
            if (n2 == m2){
                Console.WriteLine($"Детерминант второй матрицы:
{Deter(matr2)}");
            } else {
                Console.WriteLine("Детерминант можно вычислить только
для квадратных матриц");
            }
            break;

        case 4:
            if (n1 == m1){
                PrintMatr(InverMatr(matr1), "Обратная первая
матрица");
            } else {
                Console.WriteLine("Обратную матрицу можно найти
только для квадратных матриц");
            }
            if (n2 == m2){
                PrintMatr(InverMatr(matr2), "Обратная вторая
матрица");
            } else {
                Console.WriteLine("Обратную матрицу можно найти
только для квадратных матриц");
            }
            break;

        case 5:
            var trans1 = TransMatr(matr1);
            PrintMatr(trans1, "Транспонированная первая матрица");
            var trans2 = TransMatr(matr2);
            PrintMatr(trans2, "Транспонированная вторая матрица");
            break;

        case 6:
            if (n1 == n2) {
                double[] b = new double[n1];
                Console.WriteLine("Введите вектор свободных членов
(b):");

                for (int i = 0; i < n1; i++) {
                    Console.Write($"b[{i}]: ");
                    b[i] = double.Parse(Console.ReadLine());
                }
            }

```



```

        double[] roots = SolveSystem(matr1, b);
        if (roots != null) {
            Console.WriteLine("Корни системы уравнений:");
            for (int i = 0; i < roots.Length; i++) {
                Console.WriteLine($"x[{i}] = {roots[i]}");
            }
        }
        else {
            Console.WriteLine("Количество строк в матрице и
векторе свободных членов должно совпадать");
        }
        break;

    case 7:
        return;

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

static void ReadyMatr(double[,] matr, string name) {
    Console.WriteLine($"Заполнение {name}:");
    for (int i = 0; i < matr.GetLength(0); i++)
        for (int j = 0; j < matr.GetLength(1); j++) {
            Console.Write($"Введите элемент [{i}, {j}]: ");
            matr[i, j] = double.Parse(Console.ReadLine());
        }
}

static void ReadyMatrRandom(double[,] matr, double a, double b){
    Random rand = new Random();
    for (int i = 0; i < matr.GetLength(0); i++)
        for (int j = 0; j < matr.GetLength(1); j++)
            matr[i, j] = rand.NextDouble() * (b - a) + a;
}

static void PrintMatr(double[,] matr, string name)
{
    Console.WriteLine(name + ":");
    for (int i = 0; i < matr.GetLength(0); i++)
    {
        for (int j = 0; j < matr.GetLength(1); j++)
            Console.Write($"{matr[i, j]} ");
        Console.WriteLine();
    }
}

```

```

        static double[,] AddMatr(double[,] matr1, double[,] matr2){
            if (matr1.GetLength(0) != matr2.GetLength(0) || matr1.GetLength(1) !=
matr2.GetLength(1))
                throw new InvalidOperationException("Матрицы имеют разные
размеры");

            double[,] result = new double[matr1.GetLength(0),
matr1.GetLength(1)];
            for (int i = 0; i < result.GetLength(0); i++)
                for (int j = 0; j < result.GetLength(1); j++)
                    result[i, j] = matr1[i, j] + matr2[i, j];

            return result;
        }

        static double[,] MulMatr(double[,] matr1, double[,] matr2){
            if (matr1.GetLength(1) != matr2.GetLength(0))
                throw new InvalidOperationException("Кол-во столбцов первой
матрицы не равно кол-ву строк второй");

            double[,] result = new double[matr1.GetLength(0),
matr2.GetLength(1)];
            for (int i = 0; i < result.GetLength(0); i++)
                for (int j = 0; j < result.GetLength(1); j++)
                    for (int k = 0; k < matr1.GetLength(1); k++)
                        result[i, j] += matr1[i, k] * matr2[k, j];

            return result;
        }

        static double[,] TransMatr(double[,] matr)
        {
            double[,] result = new double[matr.GetLength(1), matr.GetLength(0)];
            for (int i = 0; i < matr.GetLength(0); i++)
                for (int j = 0; j < matr.GetLength(1); j++)
                    result[j, i] = matr[i, j];

            return result;
        }

        static double Deter(double[,] matr){
            if (matr.GetLength(0) != matr.GetLength(1))
                throw new InvalidOperationException("Детерминант можно вычислить
только для квадратных матриц");

            if (matr.GetLength(0) == 2)
                return matr[0, 0] * matr[1, 1] - matr[0, 1] * matr[1, 0];

            if (matr.GetLength(0) == 3)

```

```

        return matr[0, 0] * (matr[1, 1] * matr[2, 2] - matr[1, 2] *
matr[2, 1]) - matr[0, 1] * (matr[1, 0] * matr[2, 2] - matr[1, 2] * matr[2,
0]) + matr[0, 2] * (matr[1, 0] * matr[2, 1] - matr[1, 1] * matr[2, 0]);
        throw new NotSupportedException("Поддерживаются только матрицы
размерности 2x2 и 3x3.");
    }

    static double[,] InverMatr(double[,] matr){
        if (matr.GetLength(0) != matr.GetLength(1))
            throw new InvalidOperationException("Обратную матрицу можно найти
только для квадратной матрицы!");

        if (matr.GetLength(0) == 2){
            double det = Deter(matr);
            if (det == 0)
                throw new InvalidOperationException("Обратная матрица не
существует!");

            return new double[,]
            {
                {matr[1, 1] / det, -matr[0, 1] / det},
                {-matr[1, 0] / det, matr[0, 0] / det}
            };
        } else {
            double det = Deter(matr);
            if (det == 0)
                throw new InvalidOperationException("Обратная матрица не
существует!");

            double[,] result = new double[3, 3];

            result[0, 0] = (matr[1, 1] * matr[2, 2] - matr[1, 2] * matr[2,
1]) / det;
            result[0, 1] = (matr[0, 2] * matr[2, 1] - matr[0, 1] * matr[2,
2]) / det;
            result[0, 2] = (matr[0, 1] * matr[1, 2] - matr[0, 2] * matr[1,
1]) / det;

            result[1, 0] = (matr[1, 2] * matr[2, 0] - matr[1, 0] * matr[2,
2]) / det;
            result[1, 1] = (matr[0, 0] * matr[2, 2] - matr[0, 2] * matr[2,
0]) / det;
            result[1, 2] = (matr[0, 2] * matr[1, 0] - matr[0, 0] * matr[1,
2]) / det;

            result[2, 0] = (matr[1, 0] * matr[2, 1] - matr[1, 1] * matr[2,
0]) / det;
            result[2, 1] = (matr[0, 1] * matr[2, 0] - matr[0, 0] * matr[2,
1]) / det;
            result[2, 2] = (matr[0, 0] * matr[1, 1] - matr[0, 1] * matr[1,
0]) / det;

```

```

        return result;
    }
}

static double[] SolveSystem(double[,] A, double[] b) {
    int n = b.Length;
    int m = A.GetLength(1);

    if (m != n+1){
        throw new InvalidOperationException("Система не имеет решения");
    }

    for (int i = 0; i < n; i++) {
        for (int k = i + 1; k < n; k++) {
            if (Math.Abs(A[k, i]) > Math.Abs(A[i, i])) {
                for (int j = 0; j < n; j++) {
                    double temp = A[i, j];
                    A[i, j] = A[k, j];
                    A[k, j] = temp;
                }
                double tempB = b[i];
                b[i] = b[k];
                b[k] = tempB;
            }
        }

        if (Math.Abs(A[i, i]) < 1e-10) {
            throw new InvalidOperationException("Определитель равен нулю.  
Система не имеет единственного решения.");
        }

        for (int k = i + 1; k < n; k++) {
            double factor = A[k, i] / A[i, i];
            for (int j = i; j < n; j++) {
                A[k, j] -= factor * A[i, j];
            }
            b[k] -= factor * b[i];
        }
    }

    double[] x = new double[n];
    for (int i = n - 1; i >= 0; i--) {
        x[i] = b[i];
        for (int j = i + 1; j < n; j++) {
            x[i] -= A[i, j] * x[j];
        }
        x[i] /= A[i, i];
    }

    return x;
}

```

	<pre>         }     } </pre>
Ответ:	
	<p><b>Положительный сценарий для матрицы со значениями с клавиатуры:</b></p> <pre> Введи кол-во строк: 2 Введи кол-во столбцов: 2 Выберите способ заполнения: 1. Свои значения 2. Рандомные числа 1 Заполнение Первая матрица: Введите элемент [0, 0]: 1 Введите элемент [0, 1]: 2 Введите элемент [1, 0]: 0 Введите элемент [1, 1]: 1 Заполнение Вторая матрица: Введите элемент [0, 0]: 1 Введите элемент [0, 1]: 4 Введите элемент [1, 0]: 5 Введите элемент [1, 1]: 2 Первая матрица: 1 2 0 1 Вторая матрица: 1 4 5 2 Выберите действие: 1. Сложение 2. Умножение 3. Нахождение детерминанта 4. Нахождение обратной матрицы 5. Транспонирование 6. Нахождение корней системы уравнений, заданных матрицей 7. Выход 1 Сумма матриц: 2 6 5 3 </pre>

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

2

Умножение матриц:

11 8

5 2

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

3

Детерминант первой матрицы: 1

Детерминант второй матрицы: -18

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

4

Обратная первая матрица:

1 -2

-0 1

Обратная вторая матрица:

-0,1111111111111111 0,2222222222222222

0,2777777777777778 -0,05555555555555555

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

5

Транспонированная первая матрица:

1 0

2 1

Транспонированная вторая матрица:

1 5

4 2

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

6

Введите вектор свободных членов (b):

b[0]: 5

b[1]: 11

Корни системы уравнений:

x[0] = -17

x[1] = 11

*Положительный сценарий для матрицы с случайными значениями:*

Введи кол-во строк:

3

Введи кол-во столбцов:

3

Выберите способ заполнения:

1. Свои значения

2. Рандомные числа

2

Введите диапазон [a; b] для рандомных чисел:

Введите a:

0

Введите b:

5

Первая матрица:

1,3156687311344655 4,135240609294072 0,27348674640730153

0,056080238527135284 1,71911479222898 4,243177225349093

2,2661475621684977 3,0976337949834623 1,2282696595857256

Вторая матрица:

0,5199515503782509 4,717946452710601 0,10011101423380242

3,683430566090373 4,3313646457524655 2,0296345427844233

0,6120112413417667 3,1992190627064327 1,9623802911050703

Выберите действие:

1. Сложение

2. Умножение

3. Нахождение детерминанта

4. Нахождение обратной матрицы

5. Транспонирование

6. Нахождение корней системы уравнений, заданных матрицей

7. Выход

1

Сумма матриц:

1,8356202815127163 8,853187062004672 0,37359776064110395

3,739510804617508 6,050479437981446 6,272811768133517

2,8781588035102645 6,296852857689895 3,190649950690796



Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

2

Умножение матриц:

16,083332618108766 24,993433612278135 9,061425115445864  
8,958271140201102 21,28555006125557 11,821516373567988  
13,339920780397497 28,03804806642164 8,924263054180345

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

3

Детерминант первой матрицы: 23,94550121515587

Детерминант второй матрицы: -26,28460929967085

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

4

Обратная первая матрица:

-0,4607242308932318 -0,17673586154265938 0,7131361966505737  
0,3986880033877121 0,04160425175110824 -0,23249788526278434  
-0,15543845668385148 0,2211524183012317 0,08477084189598456

Обратная вторая матрица:

-0,07633893553196845 0,34005177578487483 -0,34781151507701663  
0,22774287121038078 -0,036488029849110294 0,026120215429624612  
-0,3474754874806311 -0,046567074271439246 0,5754747288737047

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

5

Транспонированная первая матрица:

1,3156687311344655 0,056080238527135284 2,2661475621684977  
4,135240609294072 1,71911479222898 3,0976337949834623  
0,27348674640730153 4,243177225349093 1,2282696595857256

Транспонированная вторая матрица:

0,5199515503782509 3,683430566090373 0,6120112413417667  
4,717946452710601 4,3313646457524655 3,1992190627064327  
0,10011101423380242 2,0296345427844233 1,9623802911050703

Выберите действие:

1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход

6

Введите вектор свободных членов (b):

b[0]: 5

b[1]: 11

b[2]: 4

Корни системы уравнений:

x[0] = -1,3951708448331173

x[1] = 1,5210952451496138

x[2] = 1,9945676854782295

## Негативный сценарий

1)

```
Введите кол-во строк для первой матрицы:
2
Введите кол-во столбцов для первой матрицы:
2
Введите кол-во строк для второй матрицы:
3
Введите кол-во столбцов для второй матрицы:
1
Выберите способ заполнения:
1. Свои значения
2. Рандомные числа
1
Заполнение Первая матрица:
Введите элемент [0, 0]: 1
Введите элемент [0, 1]: 0
Введите элемент [1, 0]: 2
Введите элемент [1, 1]: 4
Заполнение Вторая матрица:
Введите элемент [0, 0]: 1
Введите элемент [1, 0]: 5
Введите элемент [2, 0]: 3
Первая матрица:
1 0
2 4
Вторая матрица:
1
5
3
Выберите действие:
1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход
1
Сложение возможно только для матриц одинакового размера
```

2)

Введите кол-во строк для первой матрицы:

2

Введите кол-во столбцов для первой матрицы:

2

Введите кол-во строк для второй матрицы:

3

Введите кол-во столбцов для второй матрицы:

1

Выберите способ заполнения:

1. Свои значения

2. Рандомные числа

1

Заполнение Первая матрица:

Введите элемент [0, 0]: 1

Введите элемент [0, 1]: 0

Введите элемент [1, 0]: 2

Введите элемент [1, 1]: 4

Заполнение Вторая матрица:

Введите элемент [0, 0]: 1

Введите элемент [1, 0]: 5

Введите элемент [2, 0]: 3

Первая матрица:

1 0

2 4

Вторая матрица:

1

5

3

Выберите действие:

1. Сложение

2. Умножение

3. Нахождение детерминанта

4. Нахождение обратной матрицы

5. Транспонирование

6. Нахождение корней системы уравнений, заданных матрицей

7. Выход

2

Умножение возможно только если количество столбцов первой матрицы равно количеству строк второй

3)

Введите кол-во строк для первой матрицы:  
2  
Введите кол-во столбцов для первой матрицы:  
2  
Введите кол-во строк для второй матрицы:  
3  
Введите кол-во столбцов для второй матрицы:  
1  
Выберите способ заполнения:  
1. Свои значения  
2. Рандомные числа  
1  
Заполнение Первая матрица:  
Введите элемент [0, 0]: 1  
Введите элемент [0, 1]: 0  
Введите элемент [1, 0]: 2  
Введите элемент [1, 1]: 4  
Заполнение Вторая матрица:  
Введите элемент [0, 0]: 1  
Введите элемент [1, 0]: 5  
Введите элемент [2, 0]: 3  
Первая матрица:  
1 0  
2 4  
Вторая матрица:  
1  
5  
3  
Выберите действие:  
1. Сложение  
2. Умножение  
3. Нахождение детерминанта  
4. Нахождение обратной матрицы  
5. Транспонирование  
6. Нахождение корней системы уравнений, заданных матрицей  
7. Выход  
3  
Детерминант первой матрицы: 4  
Детерминант можно вычислить только для квадратных матриц

4)

```
Введите кол-во строк для первой матрицы:
3
Введите кол-во столбцов для первой матрицы:
3
Введите кол-во строк для второй матрицы:
3
Введите кол-во столбцов для второй матрицы:
3
Выберите способ заполнения:
1. Свои значения
2. Рандомные числа
1
Заполнение Первая матрица:
Введите элемент [0, 0]: 1
Введите элемент [0, 1]: 2
Введите элемент [0, 2]: 3
Введите элемент [1, 0]: 1
Введите элемент [1, 1]: 2
Введите элемент [1, 2]: 3
Введите элемент [2, 0]: 10
Введите элемент [2, 1]: 20
Введите элемент [2, 2]: 30
Заполнение Вторая матрица:
Введите элемент [0, 0]: 2
Введите элемент [0, 1]: 4
Введите элемент [0, 2]: 6
Введите элемент [1, 0]: 4
Введите элемент [1, 1]: 8
Введите элемент [1, 2]: 3
Введите элемент [2, 0]: -1
Введите элемент [2, 1]: 10
Введите элемент [2, 2]: 0
Первая матрица:
1 2 3
1 2 3
10 20 30
Вторая матрица:
2 4 6
4 8 3
```

```
-1 10 0
Выберите действие:
1. Сложение
2. Умножение
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход
4
Unhandled exception. System.InvalidOperationException: Обратная матрица не существует!
```

5)

```
Введите кол-во строк для первой матрицы:
2
Введите кол-во столбцов для первой матрицы:
2
Введите кол-во строк для второй матрицы:
2
Введите кол-во столбцов для второй матрицы:
2
Выберите способ заполнения:
1. Свои значения
2. Рандомные числа
1
Заполнение Первая матрица:
Введите элемент [0, 0]: 1
Введите элемент [0, 1]: 1
Введите элемент [1, 0]: 1
Введите элемент [1, 1]: 1
Заполнение Вторая матрица:
Введите элемент [0, 0]: 1
Введите элемент [0, 1]: 4
Введите элемент [1, 0]: 7
Введите элемент [1, 1]: 8
Первая матрица:
1 1
1 1
Вторая матрица:
1 4
7 8
Выберите действие:
3. Нахождение детерминанта
4. Нахождение обратной матрицы
5. Транспонирование
6. Нахождение корней системы уравнений, заданных матрицей
7. Выход
6
Введите вектор свободных членов (b):
b[0]: 8
b[1]: 18
Unhandled exception. System.InvalidOperationException: Система не имеет решения
```