

Дисциплина «Программирование корпоративных систем»
Рабочая тетрадь 2.2
Массивы

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]);
```


1) Создание двух матриц размерности $n \times m$ (начинать с 0 и не выводить с клавиатуры);

2) Заполнение матриц значениями с клавиатуры (по выбору пользователя, с последующим выводом результата на экран);

3) Заполнение матриц случайными числами с диапазоном [a, b] (начинать с 0 и не выводить 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 determinante (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 матрицы (с последующим выводом результата на экран);

9) Нахождение первой системы уравнений, заданных матрицей (с последующим выводом результата на экран).

При выполнении предусмотреть условие выполнения цикла чтения (заключительный символ), с точки удобства создания ситуаций (заключительный символ)

1) Возможность сложения матриц по критериям совместности их размерностей;

2) Возможность умножения матрицы на скаляр и на комплексное число;

3) Возможность нахождения определителя у не квадратных матриц ($n^2 \times m$);

4) Возможность нахождения обратной матрицы в случае, если определитель равен нулю (0*0);

5) Возможность нахождения первой системы уравнений, если она не имеет решения или не имеет единственного решения.

Весь функционал должен быть реализован на чистом, структурированном и исполняемом уровне (абстрактно, функционально и т.д.) реализован, функционально, применительно не будет.

2) Неполнота размещения товара и/или его составляющих;

3) Неполнота размещения информации на внешних носителях (CD/DVD);

4) Неполнота размещения обложки носителя и/или, если допускается размещение (CD/DVD);

5) Неполнота размещения ярлыка системы управления, если она не имеет значения или не имеет правильного размещения.

Весь функционал должен быть доступен пользователям, прошедшим регистрацию с использованием стандартной панели (таблицы), форматы и т.д.) размещения функционала, применяться не будут.

[illegible]

4) Полнота, полнота обфайл системы, если документ равен нулю (0-0);

5) Полнота, полнота обфайл системы, если она не имеет равных или не имеет определенных равных.

Важно отметить, что в настоящее время, программы, разработанные с использованием страниц равных (данных, функций и т.д.) равных функций, применяются на базе

5) Полнота, тождество и свойство транзитивности, если они не имеют отношения к нечетким отношениям.

Весь функционал должен быть реализован или, программно, разработаны с использованием сторонних библиотек (Библиотека Фрагментов v.1.2) реализующих функционал, описанный на рис. 1.

Все функции должны быть реализованы или, при необходимости, разработаны с использованием страниц шаблонов (файлов, форматов и т.д.) реализации функций, описанных на рис. 1.

```
using System;
namespace MatrixCalculator
{
    class Matrix
    {
        private double[,] data;
        private int rows, cols;

        public Matrix(int rows, int cols)
        {
            this.rows = rows;
            this.cols = cols;
            this.data = new double[rows, cols];
        }

        public void FillRandom()
        {
            Console.WriteLine("Заполнение матрицы случайными значениями");
            for (int i = 0; i < this.rows; i++)
            {
                for (int j = 0; j < this.cols; j++)
                {
                    Console.WriteLine("Введите элемент [{0},{1}]".Format(i, j));
                    double val = Console.ReadLine();
                    this.data[i, j] = double.Parse(val);
                }
            }
        }

        public void FillRandomForTestValues(int rows, int cols)
        {
            Random random = new Random();
            for (int i = 0; i < rows; i++)
            {
                for (int j = 0; j < cols; j++)
                {
                    this.data[i, j] = random.Next(0, 100);
                }
            }
        }

        public void Print()
        {
            Console.WriteLine("Матрица:");
            for (int i = 0; i < this.rows; i++)
            {
                for (int j = 0; j < this.cols; j++)
                {
                    Console.WriteLine("{0} ", this.data[i, j]);
                }
                Console.WriteLine();
            }
        }

        public int GetRows() { return rows; }
        public int GetCols() { return cols; }

        public static Matrix Add(Matrix a, Matrix b)
        {
            if (a.rows != b.rows || a.cols != b.cols)
            {
                Console.WriteLine("Матрицы несовместимы для сложения. Проверьте размеры матриц.");
                return null;
            }
            Matrix result = new Matrix(a.rows, a.cols);
            for (int i = 0; i < a.rows; i++)
            {
                for (int j = 0; j < a.cols; j++)
                {
                    result.data[i, j] = a.data[i, j] + b.data[i, j];
                }
            }
            return result;
        }

        public static Matrix Multiply(Matrix a, Matrix b)
        {
            if (a.cols != b.rows)
            {
                Console.WriteLine("Матрицы несовместимы для умножения. Проверьте количество столбцов первой матрицы и количество строк второй матрицы.");
                return null;
            }
            Matrix result = new Matrix(a.rows, b.cols);
            for (int i = 0; i < a.rows; i++)
            {
                for (int j = 0; j < b.cols; j++)
                {
                    double sum = 0;
                    for (int k = 0; k < a.cols; k++)
                    {
                        sum += a.data[i, k] * b.data[k, j];
                    }
                    result.data[i, j] = sum;
                }
            }
            return result;
        }

        public static Matrix Transpose()
        {
            Matrix result = new Matrix(cols, rows);
            for (int i = 0; i < cols; i++)
            {
                for (int j = 0; j < rows; j++)
                {
                    result.data[j, i] = this.data[i, j];
                }
            }
            return result;
        }

        public double Determinant()
        {
            if (rows != cols)
            {
                Console.WriteLine("Матрица не является квадратной. Детерминант не определен.");
                return null;
            }
            return CalculateDeterminant(this.data, 0, 0);
        }

        private double CalculateDeterminant(double[,] data, int row, int col)
        {
            double det = 0;
            if (data.GetLength(0) == 1)
            {
                return data[row, col];
            }
            if (data.GetLength(0) == 2)
            {
                return data[row, col] * data[col, col + 1] - data[row, col + 1] * data[col, col];
            }
            for (int k = 0; k < data.GetLength(0) - 1; k++)
            {
                for (int i = 0; i < data.GetLength(0) - 1; i++)
                {
                    for (int j = 0; j < data.GetLength(0) - 1; j++)
                    {
                        double sign = (i + j + k) % 2 == 0 ? 1 : -1;
                        det += sign * data[row, k] * CalculateDeterminant(data, i + 1, j + 1);
                    }
                }
            }
            return det;
        }

        public Matrix Inverse()
        {
            double det = Determinant();
            if (det == 0)
            {
                Console.WriteLine("Матрица не является невырожденной. Обратная матрица не существует.");
                return null;
            }
            return null;
        }

        public void SubMatrix()
        {
            // ...
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Программа для работы с матрицами");
            Console.WriteLine("1. Заполнение матрицы случайными значениями");
            Console.WriteLine("2. Ввод значений вручную");
            Console.WriteLine("3. Сложение матриц");
            Console.WriteLine("4. Умножение матриц");
            Console.WriteLine("5. Транспонирование");
            Console.WriteLine("6. Детерминант");
            Console.WriteLine("7. Обратная матрица");
            Console.WriteLine("8. Выход");
            Console.WriteLine("Выберите действие:");
            int choice = 0;
            while (choice < 8)
            {
                choice = int.Parse(Console.ReadLine());
                switch (choice)
                {
                    case 1:
                        Matrix m1 = new Matrix(4, 4);
                        m1.FillRandom();
                        Console.WriteLine("Матрица 1:");
                        m1.Print();
                        break;
                    case 2:
                        Matrix m2 = new Matrix(4, 4);
                        m2.FillRandomForTestValues(4, 4);
                        Console.WriteLine("Матрица 2:");
                        m2.Print();
                        break;
                    case 3:
                        Matrix m3 = m1.Add(m2);
                        Console.WriteLine("Сумма матриц:");
                        m3.Print();
                        break;
                    case 4:
                        Matrix m4 = m1.Multiply(m2);
                        Console.WriteLine("Произведение матриц:");
                        m4.Print();
                        break;
                    case 5:
                        Matrix m5 = m1.Transpose();
                        Console.WriteLine("Транспонированная матрица:");
                        m5.Print();
                        break;
                    case 6:
                        double det = m1.Determinant();
                        Console.WriteLine("Детерминант матрицы 1: {0}", det);
                        break;
                    case 7:
                        Matrix m6 = m1.Inverse();
                        Console.WriteLine("Обратная матрица:");
                        m6.Print();
                        break;
                    case 8:
                        return;
                }
            }
        }
    }
}
```

Введите размерность матриц (n и m):

2

2

Как заполнить матрицы? 1 – вручную, 2 – случайно

2

Введите диапазон для случайных чисел (a и b):

1

5

Матрица 1:

Матрица:

1 1

5 4

Матрица 2:

Матрица:

5 2

4 1

Сумма матриц:

Матрица:

6 3

9 5

Произведение матриц:

Матрица:

9 3

41 14

...Program finished with exit code 0

Press ENTER to exit console.