Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

ОТЧЕТ  
к лабораторной работе по дисциплине

Технология программирования

Матрица

Наименование темы

Выполнил студент группы ИСТб-19–2 Комогорцева Ю.В.

шифр Фамилия И.О.

Проверила Бахвалова З.А.

Фамилия И.О.

Содержание отчета на 16 стр.

Иркутск 2020 г.

### Постановка задачи

**Задача №11(оранжевая)** Даны три квадратные матрицы A и B и С. Если разность между максимальным элементом и минимальным элементом матрицы B меньше разности между максимальным элементом A и минимальным элементом C то (B−C)T+2⋅A иначе A⋅3−CT⋅3+2−B

1. Находим maxB и minB
2. Находим maxA
3. Находим minC
4. ЕСЛИ (maxB – minB)<( maxA- minC) ТО
   1. F=(B−C)T+2⋅A
5. ИНАЧЕ
   1. F=A⋅3−CT⋅3+2−B

### Таблица описания классов,методов,конструкторов

### Класс Program

Main (string[] args)-точка входа в программу, взаимодействие с пользователем и вызов метода ReturnRezult () из класса Logic

**Класс Logic**

### ReturnRezult ()-метод для выполнения основного условия и выполнения задания

### Класс Matrix

Matrix(n), Matrix(int [,] inputData) – конструкторы класса

InputMatrix(int n)– ввод матрицы из консоли с проверкой данных

print() – вывод матрицы в консоль

this[I,j] – индексатор для обращения к объекту класса, как к двумерному массиву

Size – свойство для получения размера матрицы

CheckMax (Matrix A), CheckMin(Matrix A) –методы нахождения минимального и максимального элемента в матрице

Trans(Matrix A) – свойство для получения транспонированной матрицы

operator +, -( Matrix A, Matrix B ) – перегрузки для сложения, вычитания, произведения двух объектов класса

operator \*,+ (Matrix A, int c),– перегрузки для произведения и суммы матрицы и числа

operator +,\*(int c, Matrix A)- перегрузки для произведения и суммы числа и матрицы(перенаправляет на обычные операторы)

Equals – переопределение стандартного метода Equals, чтобы сравнивать 2 объекта класса в автоматизированном тестировании

### Класс Program

### Таблица спецификаций метода *Main* (string[] args)

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Входные величины* | | |
| n | Размер матрицы | Целое число от 1 до 10 |
| A,B,C | Матрицы для вычислений- объекты класса Matrix | Двумерные целочисленные массивы |
| *Выходные величины* | | |
| F | Итоговая матрица после вычислений | Класс Matrix  Двумерный целочисленный массив |

**Класс Logic**

### Таблица спецификаций метода *ReturnRezult* ()

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Входные величины* | | |
| A,B,C | Матрицы для вычислений- объекты класса Matrix | Двумерные целочисленные массивы |
| *Выходные величины* | | |
| F | Итоговая матрица после вычислений | Класс Matrix  Двумерный целочисленный массив |

### Класс Matrix

### Таблица спецификаций конструктора Matrix

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Входные величины* | | |
| size | Размер матрицы | Натуральное число |
| *Выходные величины* | | |
| size (поле класса) | Размер матрицы | Натуральное число |
| Matrix | Матрица размером size, size | Целочисленный двумерный массив |

### Таблица спецификаций конструктора Matrix

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Входные величины* | | |
| Matrix | Матрица размером size, size | Целочисленный двумерный массив |
| *Выходные величины* | | |
| Matrix (поле класса) | Матрица размером size, size | Двумерный массив целых чисел |

### Таблица спецификаций метода inputMatrix

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Входные величины* | | |
| size | Размер матрицы | Натуральное число |
| *Выходные величины* | | |
| Matrix (поле класса) | Матрица размером size, size | Двумерный массив целых чисел |

### Таблица спецификаций метода CheckMax ()

|  |  |  |
| --- | --- | --- |
| **Имя** | **Назначение** | **Тип/Диапазон** |
| *Входные величины* | | |
| A | Матрица размером size, size | Целочисленный двумерный массив |
| *Выходные величины* | | |
| Max | Максимальный элемент матрицы A | Целое число |

### Таблица спецификаций метода CheckMin ()

|  |  |  |
| --- | --- | --- |
| **Имя** | **Назначение** | **Тип/Диапазон** |
| *Входные величины* | | |
| A | Матрица размером size, size | Целочисленный двумерный массив |
| *Выходные величины* | | |
| Min | Минимальный элемент матрицы A | Целое число |

### Таблица спецификаций перегрузки оператора +

|  |  |  |
| --- | --- | --- |
| **Имя** | **Назначение** | **Тип/Диапазон** |
| *Входные величины* | | |
| A | Матрица размером size, size | Целочисленный двумерный массив |
| B | Матрица размером size, size | Целочисленный двумерный массив |
| *Выходные величины* | | |
| resultMatix | Матрица размером size, size  Сумма элементов матриц A и B | Целочисленный двумерный массив |

### Таблица спецификаций перегрузки оператора -

|  |  |  |
| --- | --- | --- |
| **Имя** | **Назначение** | **Тип/Диапазон** |
| *Входные величины* | | |
| A | Матрица размером size, size | Целочисленный двумерный массив |
| B | Матрица размером size, size | Целочисленный двумерный массив |
| *Выходные величины* | | |
| resultMatix | Матрица размером size, size  Разность элементов матриц A и B | Целочисленный двумерный массив |

### Таблица спецификаций перегрузки оператора +

|  |  |  |
| --- | --- | --- |
| **Имя** | **Назначение** | **Тип/Диапазон** |
| *Входные величины* | | |
| A | Матрица размером size, size | Целочисленный двумерный массив |
| c | Число для сложения | Целое число |
| *Выходные величины* | | |
| resultMatix | Матрица размером size, size  Сумма элементов лежащих на диагонали матрицы A и числа c | Целочисленный двумерный массив |

### Таблица спецификаций перегрузки оператора \*

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Входные величины* | | |
| A | Матрица размером size, size | Целочисленный двумерный массив |
| c | Число для умножения | Целое число |
| *Выходные величины* | | |
| resultMatix | Матрица размером size, size  Произведение элементов матрицы A и числа c | Целочисленный двумерный массив |

### Таблица спецификаций метода print()

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| size (поле класса) | Размер матрицы | Целое число от 1 до 10 |
| Matrix (поле класса) | Матрица размером size, size | Целочисленный двумерный массив |
| Выходные величины | | |
| - | Элементы матрицы matrix | Строка |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| **Проверяется автоматизированным тестированием** | | | |
| 1 | (maxB – minB)<( maxA- minC) | matrixA =  1 2  3 4  matrixB =  1 5  0 3  matrixC =  -4 10  2 7 | F =  7 2  1 4 |
| 2 | Проверка условия (maxB – minB)>( maxA- minC)  не выполняется | matrixA =  1 2  3 4  matrixB =  10 5  1 0  matrixC =  2 10  15 4 | F =  -11 -44  -22 2 |
| 3 | Проверка условия (maxB – minB)=( maxA- minC) | matrixA =  5 0  1 2  matrixB =  6 3  4 5  matrixC =  2 3  4 6 | F =  5 -15  -10 -15 |
| **Не проверяется автоматизированным тестированием** | | | |
| 3 | Проверка корректности ввода размерности матрицы | n=w | "Некорректный ввод. Введите размерность матрицы не меньше 1" |
| 4 | Проверка корректности ввода элемента массива | A[ i , j ]=w | "Некорректный ввод. Элемент матрицы может быть только числом " |

### Все автотесты логичнее проверять в методе ReturnRezult, так как он автоматически обращается к остальным методам.

### 

### Код программы

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите размерность матриц A,B,C(от 1 до 10): ");

int n;

while (!(Int32.TryParse(Console.ReadLine(), out n)) || (n < 1) || (n > 10))

Console.WriteLine("Некорректный ввод.Введите размерность матрицы не меньше 1 и не больше 10");

Console.WriteLine("\nВведите матрицу А");

var matrixA = Matrix.InputMatrix(n);

Console.WriteLine("\nВведите матрицу B");

var matrixB = Matrix.InputMatrix(n);

Console.WriteLine("\nВведите матрицу C");

var matrixC = Matrix.InputMatrix(n);

Console.WriteLine("\nВведите матрицу А");

Console.WriteLine("\nМатрица А");

matrixA.Print();

Console.WriteLine("\nМатрица B");

matrixB.Print();

Console.WriteLine("\nМатрица C");

matrixC.Print();

var F = Logic.ReturnRezult(matrixA, matrixB, matrixC);

Console.WriteLine("\nМатрица F");

F.Print(); //вывод результата в консоль

Console.ReadKey();

}

}

public class Logic

{

public static Matrix ReturnRezult(Matrix A, Matrix B, Matrix C)

{

Matrix F;

if ((Matrix.CheckMax(B)) - (Matrix.CheckMin(B)) < ((Matrix.CheckMax(A)) - (Matrix.CheckMin(C))))

{

F = Matrix.Trans(B - C) + 2 \* A;

}

else

{

F = A \* 3 - Matrix.Trans(C) \* 3 + 2 - B;

}

return F;

}

}

public class Matrix

{

public int[,] data; // поле для хранения данных

private int n; // поле для хранения размера матрицы

public Matrix(int n) //конструкторы инициализация массива

{

this.data = new int[n, n]; // инициализируем матрицу

this.n = n;

}

public int this[int i, int j] //индексатор для обращения к обьекту класса как к двумерному массиву

{

get // обращение, например если кто-то пишет var k = matrix[1,1];

{

return data[i, j]; // зеркалим данные из поля data

}

set // присваивание, вызывается если кто-то пишет matrix[1,1] = 2;

{

data[i, j] = value; // зеркалим присвоенное значение в data[i, j]

}

}

public Matrix(int[,] inputData) //новый конструктор с параметром двумерным массивом

{

// мы надеемся, что передали квадратную матрицу

// и берем за размер количество строк

this.n = inputData.GetLength(0);

// инициализируем внутренний двумерный массив

this.data = new int[this.n, this.n];

// копируем элементы, поэлементно

for (var i = 0; i < this.n; ++i)

{

for (var j = 0; j < this.n; ++j)

{

this.data[i, j] = inputData[i, j];

}

}

}

public void Print() //вывод матрицы

{

for (var i = 0; i < this.Size; ++i)

{

for (var j = 0; j < this.Size; ++j)

{

Console.Write("{0}\t", this[i, j]);

}

Console.WriteLine();

}

}

public static Matrix InputMatrix(int n) //Ввод матрицы

{

Matrix A = new Matrix(n); // объявил матрицу

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

int t;

Console.Write("Элемент[" + i + "," + j + "] : ");

while (!(Int32.TryParse(Console.ReadLine(), out t)))

Console.WriteLine("Некорректный ввод.Элемент матрицы может быть только числом ");// заполнили

A[i, j] = t;

}

}

return A;

}

public static Matrix operator +(Matrix A, Matrix B) //Оператор сложения двух матриц

{

var resultMatix = new Matrix(A.Size);

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

resultMatix[i, j] = A[i, j] + B[i, j];

}

}

return resultMatix;

}

public static Matrix operator +(Matrix A, int c) //Оператор сложения матрицы и числа

{

var resultMatix = new Matrix(A.Size);

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

if (i == j) // если элемент на диагонали, то прибавляем c

{

resultMatix[i, j] = A[i, j] + c;

}

else // а если нет, то просто присваиваем значение исходной матрицы

{

resultMatix[i, j] = A[i, j];

}

}

}

return resultMatix;

}

public static Matrix operator +(int c, Matrix A) // функция сложения матрицы и числа //

{

return A + c;

}

public static Matrix operator -(Matrix A, Matrix B) //Оператор вычитания двух матриц

{

var resultMatix = new Matrix(A.Size);

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

resultMatix[i, j] = A[i, j] - B[i, j];

}

}

return resultMatix;

}

public static Matrix operator \*(Matrix A, int c) //Оператор умножения матрицы на число

{

var resultMatix = new Matrix(A.Size);

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

resultMatix[i, j] = A[i, j] \* c;

}

}

return resultMatix;

}

public static Matrix operator \*(int c, Matrix A) // функция умножения числа на матрицу //

{

return A \* c;

}

public override bool Equals(object obj)

{

// тут мы пытаемся преобразовать переданный объект в матрицу

var B = obj as Matrix;

// если преобразование не удалось, то есть если B не матрица, то в B окажется null

if (B == null)

return false; // а если не матрица, то значит точно не совпадает, возвращаем false

// а если B оказалось матрицей поэлементно сравниваем элементы

for (var i = 0; i < this.Size; ++i)

{

for (var j = 0; j < this.Size; ++j)

{

// ищем первый несовпавший элемент

if (this[i, j] != B[i, j])

return false; // если найдем, значит не совпадают

}

}

return true; // ну а если такого не найдем, значит совпадает, возвращает true

}

public static int CheckMax(Matrix A) //Нахождение максимального элемента матрицы

{

int Max = A[0, 0];

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

if (A[i, j] > Max)

{

Max = A[i, j];

}

}

}

return Max;

}

public static int CheckMin(Matrix A) //Нахождение минимального элемента матрицы

{

int Min = A[0, 0];

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

if (A[i, j] < Min)

{

Min = A[i, j];

}

}

}

return Min;

}

public static Matrix Trans(Matrix A) //транспонирование матрицы

{

var resultMatix = new Matrix(A.Size);

for (var i = 0; i < A.Size; ++i)

{

for (var j = 0; j < A.Size; ++j)

{

resultMatix[i, j] = A[j, i];

}

}

return resultMatix;

}

public int Size // свойство которое возвращает количество элементов в матрице

{

get // тут только метод для обращения

{

return n; // возвращаем просто размер матрицы

}

}

}

}

**Код автоматизированного тестирования**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using ConsoleApp1;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1.Tests

{

[TestClass()]

public class LogicTests

{

[TestMethod()]

public void ReturnRezultTest()

{

var matrixA = new Matrix(new int[,] { { 1, 2 }, { 3, 4 } });

var matrixB = new Matrix(new int[,] { { 1, 5 }, { 0, 3 } });

var matrixC = new Matrix(new int[,] { { -4, 10 }, { 2, 7 } });

var F = new Matrix(new int[,] { { 7, 2 }, { 1, 4 } });

var rezult = Logic.ReturnRezult(matrixA, matrixB, matrixC);

Assert.AreEqual(F, rezult);

}

[TestMethod()]

public void ReturnRezultTest1()

{

var matrixA = new Matrix(new int[,] { { 1, 2 }, { 3, 4 } });

var matrixB = new Matrix(new int[,] { { 10, 5 }, { 1, 0 } });

var matrixC = new Matrix(new int[,] { { 2, 10 }, { 15, 4 } });

var rezult = Logic.ReturnRezult(matrixA, matrixB, matrixC);

var F = new Matrix(new int[,] { { -11, -44 }, { -22, 2 } });

Assert.AreEqual(F, rezult);

}

[TestMethod()]

public void ReturnRezultTest2()

{

var matrixA = new Matrix(new int[,] { { 5, 0 }, { 1, 2 } });

var matrixB = new Matrix(new int[,] { { 6, 3 }, { 4, 5 } });

var matrixC = new Matrix(new int[,] { { 2, 3 }, { 4, 6 } });

var F = new Matrix(new int[,] { { 5, -15 }, { -10, -15 } });

var rezult = Logic.ReturnRezult(matrixA, matrixB, matrixC);

Assert.AreEqual(F, rezult);

}

}

}