**Лабораторна робота №1**

**Виконала студентка КС-19**

**Ковтун Єлизавета**

*Завдання 1.*

Варіант 3:

Кожен рядок матриці A на першій і останній позиції містить індекси. відповідно початку і кінця діапазону елементів рядка, які необхідно переписати у відповідний рядок матриці B. Створити матрицю B з необхідною кількістю стовпчиків у кожному рядку та переписати до неї вказані елементи з матриці A.

Лістинг програми:

using System;

namespace Лаба1

{

class MainClass

{

public static void Main(string[] args)

{

Console.Write("Write how many rows: ");

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

Console.Write("Columns: ");

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

int[,] matrixA = new int[row, column];

int[][] matrixB = new int[row][];

Console.WriteLine("Your random matrix A: ");

RandMatrix(row, column, matrixA);

FindNumbInRow(matrixA, matrixB, row, column);

NullMatrixB(matrixB, row);

Console.WriteLine("Matrix B: ");

MatrixB(matrixA, matrixB, row, column);

}

public static void RandMatrix(int row, int column, int[,] matrixA)

{

Random rand = new Random();

for (int i = 0; i < row; i++)

{

for (int j = 0; j < column; j++)

{

matrixA[i, j] = rand.Next(-20, 20);

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

}

Console.WriteLine();

}

}

public static int Find\_Min(int[,] matrixA, int numbrow, int column)

{

int min = 0;

if (matrixA[numbrow, column - 1] < matrixA[numbrow, 0])

{

min = matrixA[numbrow, column - 1];

}

else if (matrixA[numbrow, column - 1] > matrixA[numbrow, 0])

{

min = matrixA[numbrow, 0];

}

return min;

}

public static int Find\_Max(int[,] matrixA, int numbrow, int column)

{

int max = 0;

if (matrixA[numbrow, column - 1] >= matrixA[numbrow, 0])

{

max = matrixA[numbrow, column - 1];

}

else if (matrixA[numbrow, column - 1] <= matrixA[numbrow, 0])

{

max = matrixA[numbrow, 0];

}

return max;

}

public static void FindNumbInRow(int[,] matrixA, int[][] matrixB, int row, int column)

{

for (int i = 0; i < row; i++)

{

int max = Find\_Max(matrixA, i, column);

int min = Find\_Min(matrixA, i, column);

int count = 0;

for (int j = 0; j < column; j++)

{

if ((matrixA[i, j] >= min && matrixA[i, j] <= max))

{

count++;

}

}

matrixB[i] = new int[count];

}

}

public static void NullMatrixB(int[][] matrix, int row)

{

for (int i = 0; i < row; i++)

{

for (int j = 0; j < matrix[i].Length; j++)

{

matrix[i][j] = 0;

}

}

}

public static void MatrixB(int[,] matrixA, int[][] matrixB, int row, int column)

{

for (int i = 0; i < row; i++)

{

int max = Find\_Max(matrixA, i, column);

int min = Find\_Min(matrixA, i, column);

int ij = 0;

for (int j = 0; j < column; j++)

{

if (matrixA[i, j] >= min && matrixA[i, j] <= max)

{

matrixB[i][ij] = matrixA[i, j];

Console.Write(matrixB[i][ij] + " ");

ij++;

}

}

Console.WriteLine();

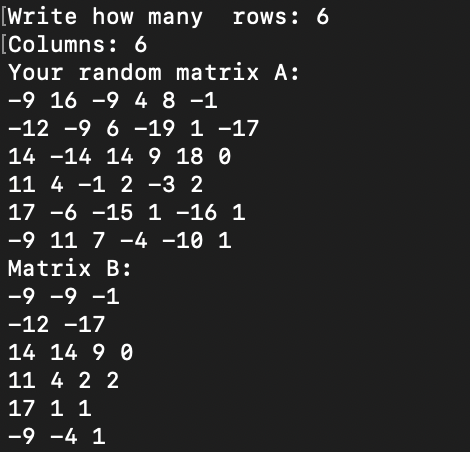
}

}

}

}

Результати роботи:



Git repository: https://github.com/eliz38781/OOP-Lab

Задача 2:

Лістинг програми:

using System;

namespace Лаба\_1\_часть\_2

{

class TTriangle

{

protected double a;

protected double b;

protected double c;

public TTriangle()

{

Random rand = new Random();

while (a + b <= c || a + c <= b || b + c <= a)

{

this.a = rand.Next(1, 100);

this.b = rand.Next(1, 100);

this.c = rand.Next(1, 100);

}

}

public double A

{

get

{

return a;

}

set

{

value = double.Parse(Console.ReadLine());

if (value + b <= c || value + c <= b || value >= b + c)

{

while (value+b <=c || value +c<= b || value>=b+c )

{

value = double.Parse(Console.ReadLine());

}

}

a = value;

}

}

public double B

{

get

{

return b;

}

set

{

value = double.Parse(Console.ReadLine());

if (value + a < c || a + c < value || a > value + c)

{

while (value + a < c || a + c < value || a > value + c)

{

value = double.Parse(Console.ReadLine());

}

}

b = value;

}

}

public double C

{

get

{

return c;

}

set

{

value = double.Parse(Console.ReadLine());

if (a + b < value || a + value < b || value > b + a)

{

while (a + b < value || a + value < b || value > b + a)

{

value = double.Parse(Console.ReadLine());

}

}

c = value;

}

}

static double Perimetr(double a, double b, double c)

{

double perimetr = a + b + c;

return perimetr;

}

static double Square(double a, double b, double c)

{

double square;

double semiper = (a + b + c) / 2;

square = Math.Sqrt(semiper \* (semiper - a) \* (semiper - b) \* (semiper - c));

return square;

}

public static void Tr(double a, double b, double c)

{

double perimetr = Perimetr(a,b,c);

double square = Square(a,b,c);

Console.WriteLine("Side A: {0} Side B: {1} Side B: {2}", a, b, c);

Console.Write("Perimetr: {0} ",perimetr);

Console.WriteLine("Square: {0} ",square);

}

}

class MainClass

{

public static void Main(string[] args)

{

TTriangle tr = new TTriangle();

TTriangle.Tr(tr.A, tr.B, tr.C);

Console.Write("Changed side A: ");

tr.A = double.Parse(Console.ReadLine());

TTriangle.Tr(tr.A, tr.B, tr.C);

Console.Write("Changed side B: ");

tr.B = double.Parse(Console.ReadLine());

TTriangle.Tr(tr.A, tr.B, tr.C);

Console.Write("Changed side C: ");

tr.C = double.Parse(Console.ReadLine());

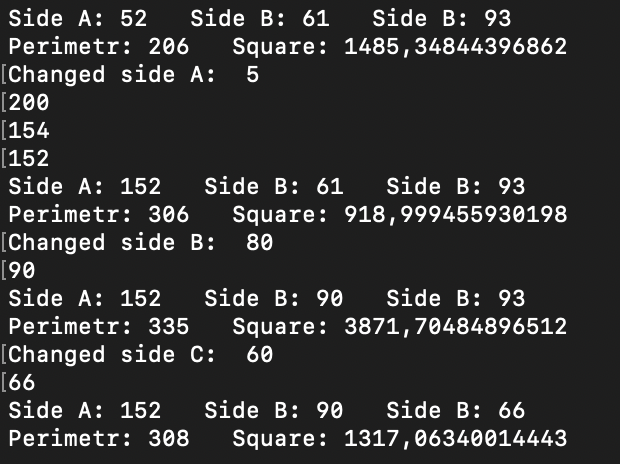
TTriangle.Tr(tr.A, tr.B, tr.C);

}

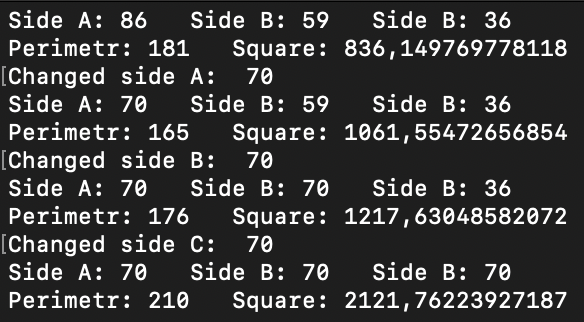
}

}

Результат роботи програми:

а) в разі, якщо трикутник не існує 

б) коли трикутник існує



Git repository:

https://github.com/eliz38781/OOP-Lab/blob/master/Lab\_1\_2.cs