# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №3**

# з курсу

**«Об’єктно-орієнтоване програмування»**

*Студента 2 курсу*

*групи ПП-11 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

к.ф.-м.н., доц. Шолохов О.В.

## Київ – 202

**1.Назва роботи**

Використання інтерфейсів у С#.

1. **Тема роботи**

Використання інтерфейсів у С#.

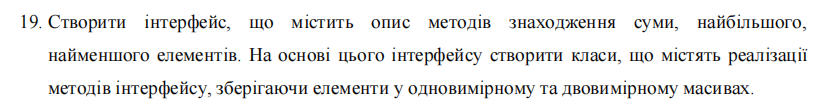
1. **Мета роботи**

Набуття навичок розробки класів з використанням інтерфейсів та базового

принципу об’єктно-орієнтованого програмування – успадкування.

1. **Умова завдання**

(19 варіант)

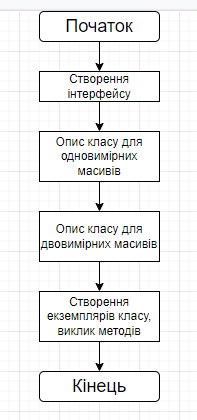


1. **Рішення**

Описуємо інтерфейс, що має потрібні для обох класів абстрактні методи. Описуємо перший клас для звичайного масиву: метод заповнення масиву випадковими числами, метод виведення масиву на екран шляхом перебору в циклі, метод пошуку найбільшого та найменшого елементів, метод пошуку суми елементів.

Після цього описуємо другий клас, який імплементує вищевказаний інтерфейс для двовимірного масиву: метод заповнення масиву випадковими числами, метод виведення масиву на екран шляхом перебору в двох циклах, метод пошуку найбільшого та найменшого елементів, метод пошуку суми елементів. Після цього у функції Main створюємо інстанції класів та викликаємо потрібні методи.

Блок-схема:



Код С#:

using System;

using System.Collections.Generic;

using System.Linq;

namespace U\_236

{

class Program

{

interface Array

{

int[,] fillJaggedArray();

void printJaggedArray(int[,] array);

void findJaggedExtremes(int[,] array);

void findJaggedSum(int[,] array);

}

class RegularArray : Array

{

/\*

public int[] fillArray()

{

Console.WriteLine("Please enter the array's dimension:");

int arrayDimension = Convert.ToInt32(Console.ReadLine());

int[] array = new int[arrayDimension];

Random randNum = new Random();

for (int i = 0; i < array.Length; i++)

{

array[i] = randNum.Next(-10, 10);

}

return array;

}

public int[,] fillJaggedArray()

{

Console.WriteLine("This is intended for a two-dimensional array.");

return null;

}

public void printArray(int[] array)

{

for (int i = 0; i < array.Length; i++)

{

Console.WriteLine(array[i]);

}

}

public void printJaggedArray(int[,] array)

{

}

public void findExtremes(int[] array)

{

int minint = array[0];

int maxint = array[0];

foreach (int value in array)

{

if (value < minint) minint = value;

if (value > maxint) maxint = value;

}

Console.WriteLine("Biggest number is: " + maxint + ". Smallest number is: " + minint + ".");

}

public void findJaggedExtremes(int[,] array)

{

}

public void findSum(int[] array)

{

int sum = 0;

foreach (int value in array)

{

sum += value;

}

Console.WriteLine("Sum of the elements is: " + sum + ".");

}

public void findJaggedSum(int[,] array)

{

}

\*/

public int[,] fillJaggedArray()

{

Console.WriteLine("Please enter the array's dimensions:");

int arrayDimension1 = Convert.ToInt32(Console.ReadLine());

int arrayDimension2 = 1;

int[,] array = new int[arrayDimension1, arrayDimension2];

Random randNum = new Random();

for (int i = 0; i < array.GetLength(0); i++)

{

array[i, 0] = randNum.Next(-10, 10);

}

return array;

}

public void printJaggedArray(int[,] array)

{

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

Console.Write(array[i, j] + "\t");

}

Console.WriteLine();

}

}

public void findJaggedExtremes(int[,] array)

{

IEnumerable<int> allValues = array.Cast<int>();

Console.WriteLine("Biggest value is: " + allValues.Max() + ".");

Console.WriteLine("Smallest value is: " + allValues.Min() + ".");

}

public void findJaggedSum(int[,] array)

{

int sum = 0;

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

sum += array[i, j];

}

}

Console.WriteLine("Sum of all elements is: " + sum + ".");

}

}

class TwoDimensionalArray : Array

{

public int[,] fillJaggedArray()

{

Console.WriteLine("Please enter the array's dimensions:");

int arrayDimension1 = Convert.ToInt32(Console.ReadLine());

int arrayDimension2 = Convert.ToInt32(Console.ReadLine());

int[,] array = new int[arrayDimension1, arrayDimension2];

Random randNum = new Random();

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

array[i, j] = randNum.Next(-10, 10);

}

}

return array;

}

public void printJaggedArray(int[,] array)

{

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

Console.Write(array[i, j] + "\t");

}

Console.WriteLine();

}

}

public void findJaggedExtremes(int[,] array)

{

IEnumerable<int> allValues = array.Cast<int>();

Console.WriteLine("Biggest value is: " + allValues.Max() + ".");

Console.WriteLine("Smallest value is: " + allValues.Min() + ".");

}

public void findJaggedSum(int[,] array)

{

int sum = 0;

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

sum += array[i, j];

}

}

Console.WriteLine("Sum of all elements is: " + sum + ".");

}

}

static void Main(string[] args)

{

RegularArray exampleRegularArray = new RegularArray();

int[,] exampleArray = exampleRegularArray.fillJaggedArray();

exampleRegularArray.printJaggedArray(exampleArray);

exampleRegularArray.findJaggedExtremes(exampleArray);

exampleRegularArray.findJaggedSum(exampleArray);

TwoDimensionalArray exampleTwoDimensionalArray = new TwoDimensionalArray();

int[,] exampleMatrixArray = exampleTwoDimensionalArray.fillJaggedArray();

exampleTwoDimensionalArray.printJaggedArray(exampleMatrixArray);

exampleTwoDimensionalArray.findJaggedExtremes(exampleMatrixArray);

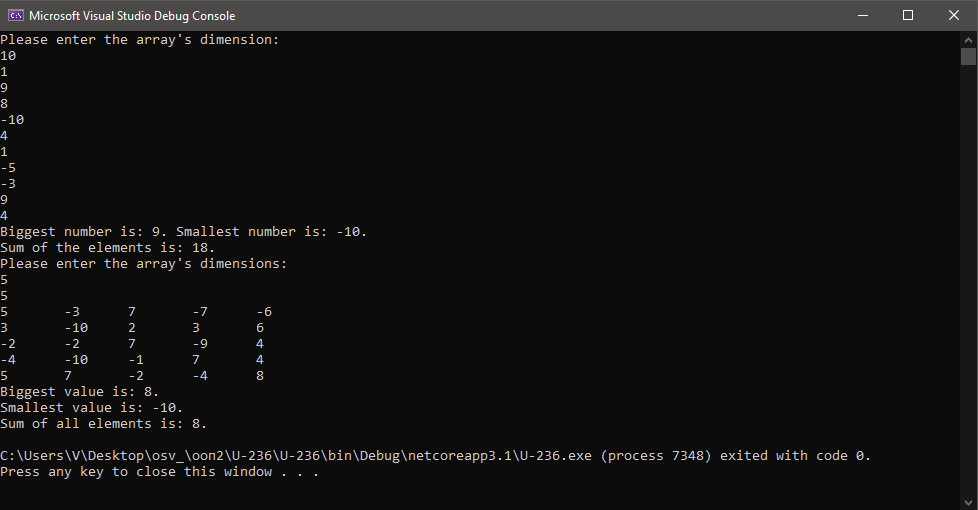
exampleTwoDimensionalArray.findJaggedSum(exampleMatrixArray);

}

}

}

Результат роботи програми C#:



**Контрольні питання**

1. Що таке інтерфейс?

Це свого роду полегшений клас, де всі функції є віртуальними і

відсутні поля.

1. Що спільного між інтерфейсами та абстрактними класами?

І там, і там можуть бути нереалізовані методи.

1. Чим відрізняються інтерфейси від абстрактних класів?

В інтерфейсі не може бути реалізованих методів, непублічних методів та полів.

1. Що повинен містити клас, який реалізує інтерфейс?

Усі методи інтерфейсу (реалізовані).

1. Чи обов’язково клас, який реалізує інтерфейс має містити реалізації всіх його методів?

Так, інакше станется помилка компіляції.

1. Чи можна змінній типу інтерфейс надавати значення об’єкта типу класу, в якому реалізовано цей інтерфейс?

Інтерфейс - це синтаксична конструкція, а не тип даних, тож їй не можна надати значення певного типу.

1. Чи може інтерфейс мати багато реалізацій?

Так, може.

1. **Висновки**

В результаті виконання даної лабораторної роботи я оволодів практичними навичками використання успадкування в С#, навчився складати програми для виконання операцій з використанням успадкування, перевантаження операторів. Вважаю дану лабораторну роботу виконаною в повному обсязі.