

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №6
з курсу
«Об'єктно-орієнтоване програмування»**

**Студента 2 курсу
групи ПП-22
спеціальності 122
«Комп'ютерні науки»
ОП «Прикладне програмування»
Шевлюк В.В.**

**Викладач:
к.ф.-м.н., доц. Шолохов О.В.**

Київ – 2022

Мета: набуття навичок розробки програм мовами C++ та C# з використанням базового принципу об'єктно-орієнтованого програмування – (одиначного) успадкування.

Завдання:

Створити консольну програму, що реалізує основні операції з одно-, дво- та n-вимірними

масивами: пошук, вставку, видалення, сортування заданим алгоритмом, заповнення за заданим алгоритмом, трансформацію за заданим алгоритмом, копіювання в інший масив елементів, що задовольняють певному критерію.

Хід роботи:

Для одно-, дво-, та багатовимірних масивів створимо спеціальні класи з полями та методами:

► Одновимірний масив:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Text;

namespace Lab2_6_OOP
{
    class ONEDArray
    {
        public List<int> arr;

        public ONEDArray()...
        public ONEDArray(List<int> arr)...
        public ONEDArray(ONEDArray a)...
        public void InputArray()...
        public void OutputArray()...
        public void MinMax()...
        public void SortArr()...
    }
}
```

► Двовимірний масив:

```
namespace Lab2_6_OOP
{
    Ссылка: 4
    class TWODArray
    {
        public int[,] arr;
        public int columns;
        public int rows;
        ссылка: 1
        public TWODArray()...
        ссылка: 1
        public void InpurArray()...
        Ссылка: 4
        public void OutputArray()...
        ссылка: 1
        public void SortArr(int index)...
        Ссылка: 2
        public void CopyArr(TWODArray a)...
        ссылка: 1
        private void Sort3()...
        ссылка: 1
        private void Sort2()...
        ссылка: 1
        private void Sort1()...
    }
}
```

► Багатовимірний масив:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab2_6_OOP
{
    Ссылка: 4
    class MULTIArray
    {
        public int[, ,] arr;
        public int columns;
        public int rows;
        public int width;
        ссылка: 1
        public MULTIArray()...
        ссылка: 1
        public void InpurArray()...
        Ссылка: 4
        public void OutputArray()...
        Ссылка: 2
        public void CopyArr(MULTIArray a)...
        ссылка: 1
        public void SortArr()...
    }
}
```

Далі у кожному класі реалізовуємо алгоритм заповнення масиву:

```
public void InpurArray()
{
    string a;
    Random rnd = new Random();
    Console.Write("Введіть розмір масиву: ");
    a = Console.ReadLine();

    int length = Convert.ToInt32(a);
    arr = new List<int>(length);
    for (int i = 0; i < length; i++)
    {
        arr.Add(rnd.Next(0, 10));
    }
}
```

Знаходження мінімального та максимального значення у масиві:

```
public void MinMax()
{
    int max = arr.Max<int>();
    int min = arr.Min<int>();
    Console.WriteLine("\nНайбільший елемент масиву: " + max);
    Console.WriteLine("\nНайменший елемент масиву " + min);
}
```

Копіювання одного масиву у другий:

```
public void CopyArr(TWODArrayu a)
{
    this.arr = a.arr;
}
```

Також різні методи сортування:

► Сортування за спаданням:

```
private void Sort3()

    int minVal = arr[0, 0];
    for (int i = 0; i < arr.GetLength(0); i++)
    {
        for (int j = 0; j < arr.GetLength(1); j++)
        {
            for (int q = i; q < arr.GetLength(0); q++)
            {
                for (int w = (q == i) ? j : 0; w < arr.GetLength(1); w++)
                {
                    if (arr[i, j] < arr[q, w])
                    {
                        minVal = arr[q, w];
                        arr[q, w] = arr[i, j];
                        arr[i, j] = minVal;
                    }
                }
            }
        }
    }
}
```

► За зростанням:

```
private void Sort2()
{
    int minVal = arr[0, 0];
    for (int i = 0; i < arr.GetLength(0); i++)
    {
        for (int j = 0; j < arr.GetLength(1); j++)
        {
            for (int q = i; q < arr.GetLength(0); q++)
            {
                for (int w = (q == i) ? j : 0; w < arr.GetLength(1); w++)
                {
                    if (arr[i, j] > arr[q, w])
                    {
                        minVal = arr[q, w];
                        arr[q, w] = arr[i, j];
                        arr[i, j] = minVal;
                    }
                }
            }
        }
    }
}
```

► Сортуння по рядкам:

```
private void Sort1()
{
    int x, b;
    for (int i = 0; i < arr.GetLength(0); i++)
    {
        for (int j = 0; j < arr.GetLength(1); j++)
        {
            x = arr[i, j];
            for (b = j - 1; b >= 0 && arr[i, b] > x; b--)
            {
                arr[i, b + 1] = arr[i, b];
            }
            arr[i, b + 1] = x;
        }
    }
}
```

► Трансформація матриці:

```
ссылка: 1
public void Transform() {
    Console.WriteLine("Трансофрмований масив: ");
    for (int i = 0; i < columns; i++)
    {
        for (int j = 0; j < rows; j++)
        {
            Console.Write($"{arr[j, i],3}");
        }
        Console.WriteLine();
    }
    Console.WriteLine();
}
```

Результат роботи програми:

```
      Array 1
Введіть розмір масиву: 2
      Array 2
Введіть розмір масиву: 3
      Array 1
9 4
Найбільший елемент масиву: 9

Найменший елемент масиву 4
Ось ваш відсортований масив:
4 9

      Array 2
7 3 2
Найбільший елемент масиву: 7

Найменший елемент масиву 2
Ось ваш відсортований масив:
2 3 7
```

```
1.Додати число до масиву
2.Видалити число з масиву
3. Скопіювати Array 1 в Array 2
4. Скопіювати Array 2 в Array 1
5.Exit
Ваш вибір: 1
1.Додати до масиву Array 1
2.Додати до масиву Array 2
Ваш вибір: 2
Ваш вибір: 4
2 3 7 4
1.Додати число до масиву
2.Видалити число з масиву
3. Скопіювати Array 1 в Array 2
4. Скопіювати Array 2 в Array 1
5.Exit
Ваш вибір: 3
      Array 2
2 3 7 4 4 9
```

Обер?ть, з яким масивом ви хочете працювати:

- 1.Одновим?рний
- 2.Двовим?рний
- 3.Багатовим?рний
- 4.Завершити роботу програми

Ваш виб?р: 2

Array 1

Введ?ть к?льк?сть рядк?в: 2

Введ?ть к?лк?сть стовбц?в: 3

Array 2

Введ?ть к?льк?сть рядк?в: 2

Введ?ть к?лк?сть стовбц?в: 1

Array 1

59 99 46

12 90 17

- 1.В?дсортувати за рядками
- 2.В?дсортувати за зб?льшенням
- 3.В?дсортувати за зменшенням
4. Трансофрмувати

Ваш виб?р: 1

Ось ваш в?дсортований масив:

46 59 99

12 17 90

Array 2

96

71

- 1.В?дсортувати за рядками
- 2.В?дсортувати за зб?льшенням
- 3.В?дсортувати за зменшенням
4. Трансофрмувати

Ваш виб?р: 2

Ось ваш в?дсортований масив:

71

96


```

Array 1
57 39 82
91 36 72
1.В?дсортувати за рядками
2.В?дсортувати за зб?льшенням
3.В?дсортувати за зменшенням
4. Трансофрмувати
Ваш виб?р: 4
Трансофрмований масив:
57 91
39 36
82 72

```

```

Array 2
77 65
74 94
21 84
1.В?дсортувати за рядками
2.В?дсортувати за зб?льшенням
3.В?дсортувати за зменшенням
4. Трансофрмувати
Ваш виб?р: 3
Ось ваш в?дсортований масив:
94 84
77 74
65 21

```

Висновок: у ході цієї лабораторної роботи я набула навичок розробки програм мовами C++ та C# з використанням базового принципу об'єктно-орієнтованого програмування – (одиначного) успадкування.

Контрольні питання:

1. Які види масивів можна створювати в C#?

Масив – іменована структура даних, фіксованого розміру, яка дозволяє зберігати, послідовність однотипних елементів, до яких можна звертатися за допомогою індексу. C# підтримує наступні типи масивів:

- одновимірні;
- ступінчасті;
- багатовимірні.

Одновимірний, або лінійний масив – це набір елементів фіксованої довжини та наперед заданого типу, доступ до яких здійснюється з використанням одного індексу.

Багатовимірний масив це масив кількості вимірів (тобто ранг) якого більше 1.

Ступінчастий масив- масив масивів, різної розмірності.

2. Як оголошуються різні види масивів?

Одновимірний

тип[] ім'я_масиву = new тип[довжина_масиву];

Багатовимірний масив

тип[,] ім'я_масиву = new тип[кількість рядків,кількість стовпців]; //для двовимірних масивів

Ступінчастий масив

тип[][] ім'я_масиву = new тип[кількість_масивів][];

3. Оператор *foreach* – його синтаксис та приклади роботи з масивами.

Оператор `foreach` забезпечує простий і зрозумілий спосіб перебору елементів масиву.

Для одновимірних масивів оператор `foreach` обробляє елементи в порядку збільшення індексу, починаючи з індексу 0 і закінчуючи індексом `Length - 1`

Для багатовимірних масивів елементи обходять таким чином, що спочатку збільшуються індекси крайнього правого виміру, потім наступного лівого виміру і так далі вліво.

```
foreach (var b in bytes){  
    Console.WriteLine(b);  
}
```

4. Елементи класу : *Rank, Length, GetLength, Clear, Sort, Clone, IndexOf, Reverse, GetLowerBound, GetUpperBound, GetType*. Їх онис (що вони виконують)

- `Rank` – повертає розмірність масиву;
- `GetLength` – повертає кількість елементів у вказаному вимірі масиву;
- `Length` – повертає загальну кількість елементів у масиві;
- `GetLowerBound` – повертає нижню границю для зазначеного виміру;

- `GetUpperBound` – повертає верхню границю для зазначеного виміру;
- `Sort` – (статичний метод) сортує елементи в одному або кількох масивах. Тип елемента масиву повинен мати реалізацію інтерфейсу `IComparer` чи передавати об'єкт, тип якого має реалізацію інтерфейсу `IComparer`;
- `IndexOf` – (статичний метод) повертає індекс першого входження значення в одновимірному масиві або його частині;
- `Clone` – створює новий масив, який є обмеженою (shallow) копією вихідного масиву;
- `CopyTo` – копіює елементи з одного масиву до іншого;
- `Clear` – (статичний метод) встановлює діапазону елементів масиву значення 0 чи null;
- `Reverse` - змінює порядок елементів в одновимірному масиві або в його частині.
- `Get Type` - отримує тип поточного екземпляра.