**МІНІСТЕРСТВО**

**ОСВІТИ ТА НАУКИ УКРАЇНИ**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Кафедра інформаційних систем та технологій**

**Звіт**

**з лабораторної  роботи № 1**

**«Класи та об’єкти. Конструктори та деструктори. Модифікатори.»**

**з дисципліни**

**«Програмування – 2. ООП»**

**Варіант №21**

|  |  |  |
| --- | --- | --- |
| **Перевірив:**  **доц. Корнага Ярослав Ігорович** |  | **Виконала:** |
|  |  | **Студентка гр. ІС-12 ФІОТ**  **Мельникова Катерина** |
|  |  | **1 курс,**  **залікова книжка № IC-1222** |

**Київ 2021**

**Тема:** Класи та об’єкти. Конструктори та деструктори. Модифікатори.

**Завдання 1**

Створити клас с атрибутами та конструктором. У методі main() ініціалізувати створення екземплярів класу та продемонструвати роботу його методів згідно умов завдання.

**21.** Скласти опис класу для послідовності. Зберігає послідовність цілих чисел. Методи: тип (спадна, зростаюча, неспадна, незростаюча, геометрична, арифметична прогресія), належність елемента, чи рівні дві послідовності, максимум, мінімум, встановити роздільники підпослідовностей (лок. максимуми, лок. мінімуми, лок. екстремуми), найбільша (найменша) підпослідовність.

**Завдання 2**

Створити у попередньому завданні два методи з використанням серіалізації та десеріалізації JSON.

* **Метод 1**. Зберігає створений об’єкт класу з Завдання 1 у JSON файл
* **Метод 2.** Відкриває JSON файл з даними та створює об’єкт класу з цими даними для виконання Завдання 1.

**Лістинг програми**

*Sequence.cs*

using System;

using Newtonsoft.Json;

using System.IO;

public class Sequence

{

public int[] array;

public Sequence(int[] numbers)

{

array = numbers;

}

public bool Is\_in(int n)

{

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

{

if (array[i] == n)

{

return true;

}

}

return false;

}

public bool Is\_equal(int[] list)

{

if (array.Length != list.Length) { return false; }

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

{

if (array[i] != list[i])

{

return false;

}

}

return true;

}

public int Max()

{

int max = array[0];

for (int i = 1; i < array.Length; i++)

{

if (max < array[i])

{

max = array[i];

}

}

return max;

}

public int Min()

{

int min = array[0];

for (int i = 1; i < array.Length; i++)

{

if (min > array[i])

{

min = array[i];

}

}

return min;

}

public void My\_Type()

{

bool is\_descending = true; // спадна послідовність

bool is\_ascending = true; // зростаюча послідовність

bool is\_notdes = true; // неспадна послідовність

bool is\_notasc = true; // незростаюча послідовність

for (int i = 0; i < array.Length-1; i++)

{

if (array[i] <= array[i+1])

{

is\_descending = false;

}

if(array[i] >= array[i + 1])

{

is\_ascending = false;

}

if (array[i] < array[i + 1])

{

is\_notdes = false;

}

if (array[i] > array[i + 1])

{

is\_notasc = false;

}

}

if (is\_descending) { Console.WriteLine("Спадна послiдовнiсть"); }

if (is\_ascending) { Console.WriteLine("Зростаюча послiдовнiсть"); }

if (is\_notasc & is\_ascending != true) { Console.WriteLine("Неспадна послiдовнiсть"); }

if (is\_notdes & is\_descending != true) { Console.WriteLine("Незростаюча послiдовнiсть"); }

bool is\_arithmetic = true; // неспадна послідовність

bool is\_geometric = true; // незростаюча послідовність

if (array.Length >= 2)

{

int d = array[1] - array[0];

int q = array[1] / array[0];

for (int i = 1; i < array.Length - 1; i++)

{

if (array[i+1] - array[i] != d)

{

is\_arithmetic = false;

}

if (array[i] != 0)

{

if (array[i + 1] / array[i] != q)

{

is\_geometric = false;

}

}

else { is\_geometric = false; }

}

}

if (is\_arithmetic) { Console.WriteLine("Арифметична прогресiя"); }

if (is\_geometric) { Console.WriteLine("Геометрична прогресiя"); }

}

public int[] loc\_max()

{

string numbers = "";

if (array.Length == 1) { numbers += array[0] + " "; }

if (array.Length >= 2)

{

if (array[0] > array[1]) { numbers += array[0] + " "; }

if (array[^1] > array[^2]) { numbers += array[^1] + " "; }

}

if (array.Length >= 3)

{

for (int i = 1; i < array.Length - 1; i++)

{

if (array[i] > array[i - 1] & array[i] > array[i + 1])

{

numbers += array[i] + " ";

}

}

}

string[] s = numbers.Split(' ');

int[] a = new int[s.Length-1];

for (int i = 0; i < s.Length-1; i++) { a[i] = int.Parse(s[i]); }

return a;

}

public int[] loc\_min()

{

string numbers = "";

if (array.Length == 1) { numbers += array[0] + " "; }

if (array.Length >= 2)

{

if (array[0] < array[1]) { numbers += array[0] + " "; }

if (array[^1] < array[^2]) { numbers += array[^1] + " "; }

}

if (array.Length >= 3)

{

for (int i = 1; i < array.Length - 1; i++)

{

if (array[i] < array[i - 1] & array[i] < array[i + 1])

{

numbers += array[i] + " ";

}

}

}

string[] s = numbers.Split(' ');

int[] a = new int[s.Length - 1];

for (int i = 0; i < s.Length - 1; i++) { a[i] = int.Parse(s[i]); }

return a;

}

public int[] extremes()

{

string numbers = "";

if (array.Length == 1) { numbers += array[0] + " "; }

if (array.Length >= 2)

{

if (array[0] != array[1]) { numbers += array[0] + " "; }

for (int i = 1; i < array.Length - 1; i++)

{

if ((array[i] < array[i - 1] & array[i] < array[i + 1]) || (array[i] > array[i - 1] & array[i] > array[i + 1]))

{

numbers += array[i] + " ";

}

}

if (array[^1] != array[^2]) { numbers += array[^1] + " "; }

}

string[] s = numbers.Split(' ');

int[] a = new int[s.Length - 1];

for (int i = 0; i < s.Length - 1; i++) { a[i] = int.Parse(s[i]); }

return a;

}

public int[] max\_subsequence()

{

string sub = array[0] + " ";

string max\_sub = "";

int len = 1;

int max\_len = 0;

for (int i = 1; i < array.Length; i++)

{

sub += array[i] + " ";

len += 1;

if (i == array.Length - 1)

{

if (len > max\_len)

{

max\_len = len;

max\_sub = sub;

}

}

else

{

if ((array[i] > array[i - 1] && array[i] > array[i + 1]) || (array[i] < array[i - 1] && array[i] < array[i + 1]))

{

if (len > max\_len)

{

max\_len = len;

max\_sub = sub;

}

sub = array[i] + " ";

len = 0;

}

}

}

string[] s = max\_sub.Split(' ');

int[] a = new int[s.Length - 1];

for (int i = 0; i < s.Length - 1; i++) { a[i] = int.Parse(s[i]); }

return a;

}

public int[] min\_subsequence()

{

string sub = array[0] + " ";

string min\_sub = "";

int len = 1;

int min\_len = array.Length;

for (int i = 1; i < array.Length; i++)

{

sub += array[i] + " ";

len += 1;

if (i == array.Length - 1)

{

if (len <= min\_len)

{

min\_len = len;

min\_sub = sub;

}

}

else

{

if ((array[i] > array[i - 1] && array[i + 1] < array[i]) || (array[i] < array[i - 1] && array[i + 1] > array[i]))

{

if (len <= min\_len)

{

min\_len = len;

min\_sub = sub;

}

sub = array[i] + " ";

len = 0;

}

}

}

string[] s = min\_sub.Split(' ');

int[] a = new int[s.Length - 1];

for (int i = 0; i < s.Length - 1; i++) { a[i] = int.Parse(s[i]); }

return a;

}

public void ToJson(string filePath)

{

string j = JsonConvert.SerializeObject(this);

File.WriteAllText(filePath, j);

}

public static Sequence FromJson(string filePath)

{

return JsonConvert.DeserializeObject<Sequence>(File.ReadAllText(filePath));

}

}

*Program.cs*

using System;

namespace Lab\_1

{

class Program

{

static void Main(string[] args)

{

string[] input = Console.ReadLine().Split(' ');

int[] array = new int[input.Length];

for (int i = 0; i < input.Length; i++) { array[i] = int.Parse(input[i]); }

Sequence s = new Sequence(array);

Console.WriteLine(s.Is\_in(5));

int[] list = new int[] {1, 2, 3, 4, 5 };

Console.WriteLine(s.Is\_equal(list));

Console.WriteLine(s.Max());

Console.WriteLine(s.Min());

s.My\_Type();

Console.WriteLine("Локальнi максимуми: ");

for (int i = 0; i < s.loc\_max().Length; i++) { Console.WriteLine(s.loc\_max()[i] + " "); }

Console.WriteLine("Локальнi мiнiмуми: ");

for (int i = 0; i < s.loc\_min().Length; i++) { Console.WriteLine(s.loc\_min()[i] + " "); }

Console.WriteLine("Екстремуми: ");

for (int i = 0; i < s.extremes().Length; i++) { Console.WriteLine(s.extremes()[i] + " "); }

Console.WriteLine("Найбiльша пiдпослiдовнiсть: ");

for (int i = 0; i < s.max\_subsequence().Length; i++) { Console.WriteLine(s.max\_subsequence()[i] + " "); }

Console.WriteLine("Найменша пiдпослiдовнiсть: ");

for (int i = 0; i < s.min\_subsequence().Length; i++) { Console.WriteLine(s.min\_subsequence()[i] + " "); }

s.ToJson("new.json");

var json\_array = Sequence.FromJson("test.json");

Console.WriteLine("Масив з json-файлу: ");

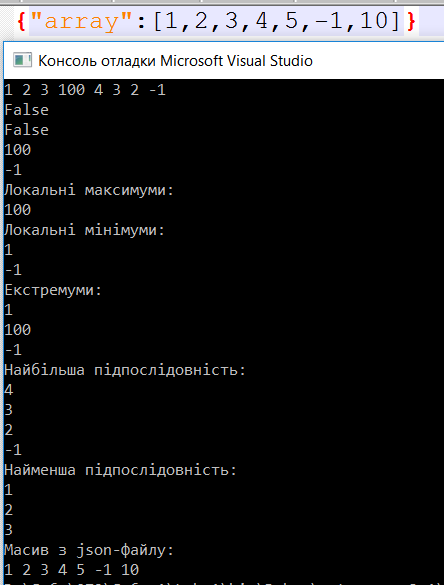
for (int i = 0; i < json\_array.array.Length; i++) { Console.Write(json\_array.array[i] + " "); }

}

}

}

**Результат виконання програми**



**Висновок**

Під час виконання цієї лабораторної роботи, я навчилася працювати з класами, об’єктами, конструкторами, деструкторами та модифікаторами у мові С#, а також з json-файлами.