**Лабораторна робота №2. Використання масивів і колекцій у мові C#.**

**Виконавець: Лазоренко М.Д.**

**Група виконавця: КН-922в**

**Мета роботи:** набути навички в написанні програм з використання масивів і колекцій.

**Індивідуальні завдання**

**Варіант №9**

**Завдання 1**

Використовуючи завдання власного варіанта до лабораторної роботи 8 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.

*Лабораторна робота 8:* *Треба оголосити масив цілих чисел і заповнити його випадковими значеннями. Розмір масиву і діапазон значень його елементів задані у виданому варіанті індивідуального завдання. В індивідуальних завданнях вказано також, яку обробку масиву слід провести.*

*Для всіх варіантів завдання слід мати на увазі наступне:*

* *нуль вважається додатним числом, якщо в завданні не обумовлений якийсь інший його статус;*
* *коли мова йде про якісь послідовності чисел, мається на увазі послідовність з довжиною, більшою одиниці;*
* *у тих випадках, коли завдання вимагає виконання якихось обчислень, дозволяється виконувати їх з тією точністю, яку забезпечують операції цілочисельної арифметики.*

|  |  |  |  |
| --- | --- | --- | --- |
| **№ варіанту** | **Розмірність масиву** | **Діапазон** | **Що потрібно зробити** |
| 9 | 200 | 0 — 100 | Знайти безперервну ділянку з 10 елементів, сума яких максимальна |

***Текст програми:***

using System;

class task\_01

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

int[] array = new int[200];

Random rand = new Random();

// Заповнення масиву випадковими значеннями від 0 до 100

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

{

array[i] = rand.Next(0, 101);

}

// Знаходження безперервної ділянки з 10 елементів з максимальною сумою

int maxSum = int.MinValue;

int startIndex = 0;

for (int i = 0; i <= array.Length - 10; i++)

{

int currentSum = 0;

for (int j = 0; j < 10; j++)

{

currentSum += array[i + j];

}

if (currentSum > maxSum)

{

maxSum = currentSum;

startIndex = i;

}

}

// Виведення результатів

Console.WriteLine("Максимальна сума: " + maxSum);

Console.WriteLine("Початковий індекс ділянки: " + startIndex);

Console.WriteLine("Елементи ділянки: ");

for (int i = startIndex; i < startIndex + 10; i++)

{

Console.Write(array[i] + " ");

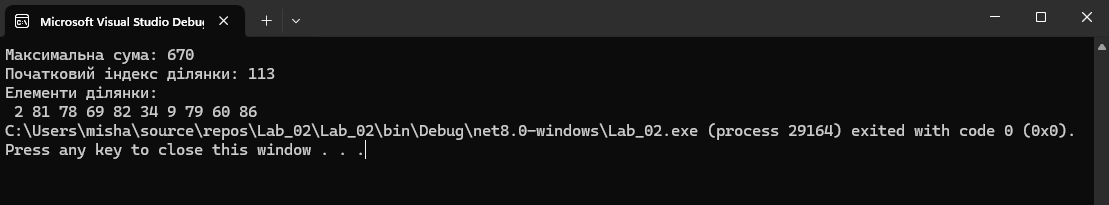
}

Console.ReadKey();

}

}

***Результат роботи програми:***



***Опис програми:***

Ця програма на C# генерує масив з 200 випадкових цілих чисел в діапазоні від 0 до 100, а потім знаходить безперервну ділянку з 10 елементів, сума яких є максимальною серед усіх можливих ділянок. Після обчислення максимального значення, програма виводить на екран:

* Максимальну суму елементів.
* Початковий індекс цієї ділянки в масиві.
* Саму ділянку (10 елементів).

***Ієрархія та структура класів:***

У програмі використовується лише один клас task\_01 і один метод Main, що є точкою входу в програму.

**Клас task\_01**

* **Призначення**: Основний клас, в якому відбувається вся логіка програми. У ньому оголошується масив, генерується випадковий набір чисел, виконується обчислення максимальної суми ділянки, і виводяться результати.
* **Методи**:
  + Main(): Головний метод, який містить основну програмну логіку.

**Завдання 2\***

Виконати минуле завдання, зробивши похідний масив динамічним, задавши його розмір з консолі.  
***Текст програми:***

using System;

class task\_02

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.Write("Введіть розмір масиву: ");

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

int[] array = new int[size];

Random rand = new Random();

// Заповнення масиву випадковими значеннями від 0 до 100

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

{

array[i] = rand.Next(0, 101);

}

// Знаходження безперервної ділянки з 10 елементів з максимальною сумою

int maxSum = int.MinValue;

int startIndex = 0;

for (int i = 0; i <= array.Length - 10; i++)

{

int currentSum = 0;

for (int j = 0; j < 10; j++)

{

currentSum += array[i + j];

}

if (currentSum > maxSum)

{

maxSum = currentSum;

startIndex = i;

}

}

// Виведення результатів

Console.WriteLine("Максимальна сума: " + maxSum);

Console.WriteLine("Початковий індекс ділянки: " + startIndex);

Console.WriteLine("Елементи ділянки: ");

for (int i = startIndex; i < startIndex + 10; i++)

{

Console.Write(array[i] + " ");

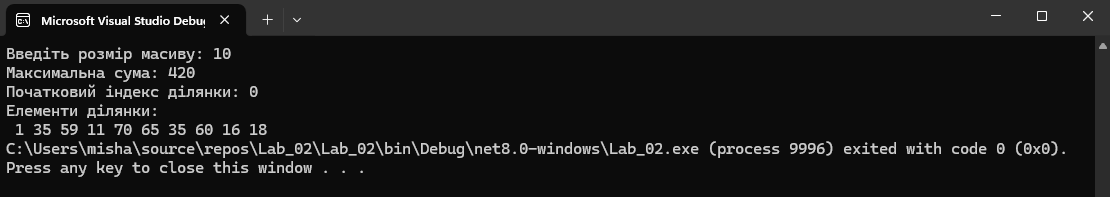
}

Console.ReadKey();

}

}

***Результат роботи програми:***



**Завдання 3\***

Виконати минуле завдання, використовуючи кольорові можливості консолі для покращення інтерфейсу роботи програми. Наприклад, виділяти різними кольорами знайдені в масивах за завданням елементи чи їхні послідовності, розраховані параметри, тощо.

***Текст програми:***

using System;

class task\_03

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.Write("Введіть розмір масиву: ");

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

int[] array = new int[size];

Random rand = new Random();

// Заповнення масиву випадковими значеннями від 0 до 100

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

{

array[i] = rand.Next(0, 101);

}

// Знаходження безперервної ділянки з 10 елементів з максимальною сумою

int maxSum = int.MinValue;

int startIndex = 0;

for (int i = 0; i <= array.Length - 10; i++)

{

int currentSum = 0;

for (int j = 0; j < 10; j++)

{

currentSum += array[i + j];

}

if (currentSum > maxSum)

{

maxSum = currentSum;

startIndex = i;

}

}

// Виведення результатів з використанням кольорів

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Максимальна сума: " + maxSum);

Console.ResetColor();

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("Початковий індекс ділянки: " + startIndex);

Console.ResetColor();

Console.WriteLine("Елементи ділянки: ");

for (int i = startIndex; i < startIndex + 10; i++)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write(array[i] + " ");

Console.ResetColor();

}

Console.ReadKey();

}

}

***Результат роботи програми:***



**Завдання 4\***

Виконати минуле завдання, використовуючи замість масивів одну з колекцій мови C# (List, LinkedList, ArrayList, чи ін.).

***Текст програми:***

using System;

using System.Collections.Generic;

class task\_04

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.Write("Введіть розмір колекції: ");

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

List<int> list = new List<int>();

Random rand = new Random();

// Заповнення колекції випадковими значеннями від 0 до 100

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

{

list.Add(rand.Next(0, 101));

}

// Знаходження безперервної ділянки з 10 елементів з максимальною сумою

int maxSum = int.MinValue;

int startIndex = 0;

for (int i = 0; i <= list.Count - 10; i++)

{

int currentSum = 0;

for (int j = 0; j < 10; j++)

{

currentSum += list[i + j];

}

if (currentSum > maxSum)

{

maxSum = currentSum;

startIndex = i;

}

}

// Виведення результатів з використанням кольорів

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Максимальна сума: " + maxSum);

Console.ResetColor();

Console.ForegroundColor = ConsoleColor.Yellow;

Console.WriteLine("Початковий індекс ділянки: " + startIndex);

Console.ResetColor();

Console.WriteLine("Елементи ділянки: ");

for (int i = startIndex; i < startIndex + 10; i++)

{

Console.ForegroundColor = ConsoleColor.Cyan;

Console.Write(list[i] + " ");

Console.ResetColor();

}

Console.ReadKey();

}

}

***Результат роботи програми:***



**Завдання 5**

Використовуючи завдання власного варіанта до лабораторної роботи 9 з електронних методичних вказівок з основ програмування та алгоритмічних мов створити проект з реалізації цих завдань засобами мови C#.

*Лабораторна робота 9:* *Створити квадратну матрицю розміром 9х9. В індивідуальних завданнях вказано, який тип даних використати та яку обробку матриці потрібно виконати. Якщо по умові завдання матрицю слід заповнити випадковими числами, рекомендується вибирати ці числа з діапазону 0 – 99. Якщо за умовою завдання в матрицю слід записати ЛПЧ – лінійну послідовність чисел, мається на увазі послідовність: 1,2,3...*

|  |  |  |
| --- | --- | --- |
| **№ варіанту** | **Зміст завдання** | **Ілюстрація** |
| 9 | Заповнити матрицю ЛПЧ, від лівого нижнього кута по діагоналі: вліво – вгору. |  |

***Текст програми:***

using System;

class task\_05

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

int n = 9;

int[,] matrix = new int[n, n];

int number = 1;

// Заповнення матриці лінійною послідовністю чисел (ЛПЧ) від лівого нижнього кута по діагоналі вліво-вгору

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

{

int row = n - 1;

int col = i;

while (row >= 0 && col >= 0)

{

matrix[row, col] = number++;

row--;

col--;

}

}

// Заповнення залишкових діагоналей

for (int j = 1; j < n; j++)

{

int row = n - 1 - j;

int col = n - 1;

while (row >= 0 && col >= 0)

{

matrix[row, col] = number++;

row--;

col--;

}

}

// Виведення матриці

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

{

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

{

Console.Write(matrix[i, j].ToString("D2") + " ");

}

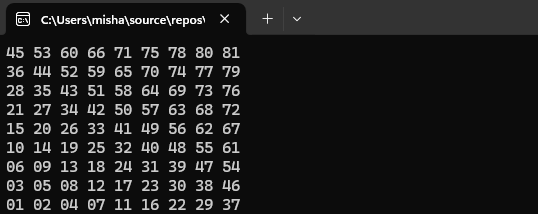
Console.WriteLine();

}

Console.ReadKey();

}

}

***Результат роботи програми:***

***Опис програми:***

Ця програма на C# генерує та заповнює квадратну матрицю (двовимірний масив) розміром 9x9 числами за певним порядком, заповнюючи діагоналі знизу вгору. Числа записуються по діагоналях, спершу з правого краю, потім продовжуючи з правого верхнього кута, доки вся матриця не буде заповнена. Після заповнення програма виводить цю матрицю на екран, форматуючи кожен елемент у двозначному вигляді.

***Ієрархія та структура класів:***

У цій програмі використовується лише один клас task\_05 і один метод Main, що містить всю логіку програми.

**Клас task\_05**

* **Призначення**: Основний клас, який відповідає за генерування та заповнення квадратної матриці, а також виведення результату на екран.
* **Методи**:
  + Main(): Головний метод, що містить такі кроки:
    1. Оголошення змінних для розміру матриці та початкового числа.
    2. Логіка заповнення двох груп діагоналей матриці числами.
    3. Виведення матриці у форматованому вигляді на консоль.

**Завдання 6\***

Виконати минуле завдання, зробивши матрицю динамічною, задавши її розміри з консолі, використовуючи для цього прямокутний ([ , ]) двовимірний масив.

***Текст програми:***

using System;

class task\_06

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

// Запит розміру матриці у користувача

Console.Write("Введіть розмір матриці: ");

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

int[,] matrix = new int[n, n];

int number = 1;

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

{

int row = n - 1;

int col = i;

while (row >= 0 && col >= 0)

{

matrix[row, col] = number++;

row--;

col--;

}

}

for (int j = 1; j < n; j++)

{

int row = n - 1 - j;

int col = n - 1;

while (row >= 0 && col >= 0)

{

matrix[row, col] = number++;

row--;

col--;

}

}

// Виведення матриці

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

{

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

{

Console.Write(matrix[i, j].ToString("D2") + " ");

}

Console.WriteLine();

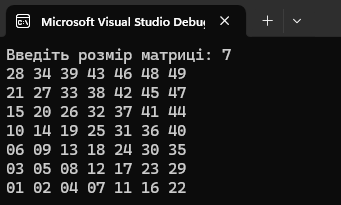
}

Console.ReadKey();

}

}

***Результат роботи програми:***



**Завдання 7\***

Виконати минуле завдання, зробивши матрицю динамічною, задавши її розміри з консолі, використовуючи для цього зубчастий ([][]) двовимірний масив.

***Текст програми:***

using System;

class task\_07

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.Write("Введіть розмір матриці: ");

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

//використання зубчастого двовимірного масиву

int[][] matrix = new int[n][];

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

{

matrix[i] = new int[n];

}

int number = 1;

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

{

int row = n - 1;

int col = i;

while (row >= 0 && col >= 0)

{

matrix[row][col] = number++;

row--;

col--;

}

}

for (int j = 1; j < n; j++)

{

int row = n - 1 - j;

int col = n - 1;

while (row >= 0 && col >= 0)

{

matrix[row][col] = number++;

row--;

col--;

}

}

// Виведення матриці

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

{

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

{

Console.Write(matrix[i][j].ToString("D2") + " ");

}

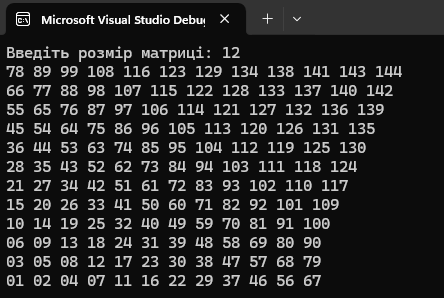
Console.WriteLine();

}

Console.ReadKey();

}

}

***Результат роботи програми:***

**Завдання 8\***

Виконати минуле завдання, використовуючи кольорові можливості консолі для покращення інтерфейсу роботи програми. Наприклад, виділяти різними кольорами знайдені в матриці за завданням елементи чи їхні послідовності, розраховані параметри, тощо.

***Текст програми:***

using System;

class task\_08

{

static void Main()

{

Console.InputEncoding = System.Text.Encoding.UTF8;

Console.OutputEncoding = System.Text.Encoding.UTF8;

Console.Write("Введіть розмір матриці: ");

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

int[][] matrix = new int[n][];

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

{

matrix[i] = new int[n];

}

int number = 1;

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

{

int row = n - 1;

int col = i;

while (row >= 0 && col >= 0)

{

matrix[row][col] = number++;

row--;

col--;

}

}

for (int j = 1; j < n; j++)

{

int row = n - 1 - j;

int col = n - 1;

while (row >= 0 && col >= 0)

{

matrix[row][col] = number++;

row--;

col--;

}

}

// Виведення матриці з кольоровими елементами

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

{

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

{

if (matrix[i][j] % 2 == 0)

{

Console.ForegroundColor = ConsoleColor.Green; // Парні числа зеленим кольором

}

else

{

Console.ForegroundColor = ConsoleColor.Red; // Непарні числа червоним кольором

}

Console.Write(matrix[i][j].ToString("D2") + " ");

Console.ResetColor();

}

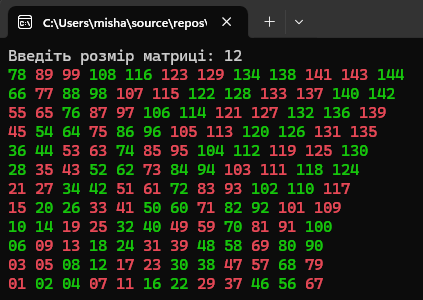
Console.WriteLine();

}

Console.ReadKey();

}

}

***Результат роботи програми:***

**Завдання 9**

Виконати підсумкове завдання згідно із номером свого варіанту

9. Знайти кількість рядків матриці, середнє арифметичне елементів яких менше за задану величину.

***Текст програми:***

using System;

class task\_09

{

static void Main()

{

Console.Write("Введіть розмір матриці: ");

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

// Ініціалізація зубчастого масиву

int[][] matrix = new int[n][];

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

{

matrix[i] = new int[n];

}

int number = 1;

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

{

int row = n - 1;

int col = i;

while (row >= 0 && col >= 0)

{

matrix[row][col] = number++;

row--;

col--;

}

}

for (int j = 1; j < n; j++)

{

int row = n - 1 - j;

int col = n - 1;

while (row >= 0 && col >= 0)

{

matrix[row][col] = number++;

row--;

col--;

}

}

// Виведення матриці

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

{

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

{

Console.Write(matrix[i][j].ToString("D2") + " ");

}

Console.WriteLine();

}

Console.Write("Введіть задану величину: ");

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

// Підрахунок кількості рядків, середнє арифметичне яких менше за задану величину

int count = 0;

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

{

double sum = 0;

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

{

sum += matrix[i][j];

}

double average = sum / n;

if (average < threshold)

{

count++;

}

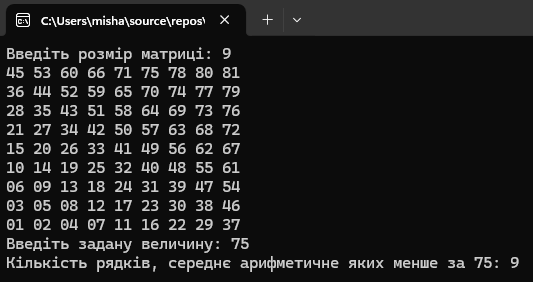
}

Console.WriteLine($"Кількість рядків, середнє арифметичне яких менше за {threshold}: {count}");

}

}

***Результат роботи програми:***



***Опис програми:***

Ця програма на C# створює квадратну матрицю заданого користувачем розміру, заповнює її числами по діагоналях, а потім обчислює кількість рядків, середнє арифметичне яких менше за задану величину (threshold). Після виконання цих дій програма виводить результат на екран.

***Ієрархія та структура класів:***

У програмі використовується лише один клас task\_09 і один метод Main, який містить всю логіку програми.

**Клас task\_09**

* **Призначення**: Основний клас, що відповідає за:
  + Введення розміру матриці та порогової величини.
  + Генерацію та заповнення двовимірної матриці.
  + Обчислення середнього арифметичного рядків і підрахунок рядків, що задовольняють умову.
  + Виведення результатів.
* **Методи**:
  + Main(): Головний метод, що виконує наступні функції:
    1. Введення розміру матриці.
    2. Створення матриці як масиву масивів (int[][]).
    3. Заповнення матриці числами по діагоналях.
    4. Виведення матриці.
    5. Введення порогової величини (threshold).
    6. Обчислення середнього арифметичного для кожного рядка і підрахунок рядків, які відповідають умові (середнє менше за threshold).
    7. Виведення результату на екран.

**Висновок**

У процесі виконання лабораторної роботи №2 було досягнуто основної мети – набуто практичних навичок у написанні програм з використанням масивів і колекцій на мові програмування C#. Було розроблено програми, які ефективно оперують двовимірними масивами, виконують їх заповнення за алгоритмом та проводять аналіз елементів, що відповідає завданням роботи. Також вивчено методи введення і виведення даних через консоль, реалізацію операцій із середнім арифметичним і використання циклів для маніпуляції з елементами масиву. Отримані знання сприятимуть подальшому вдосконаленню в програмуванні, зокрема в роботі з структурами даних у C#.