**Використання оператора foreach з масивами**

Оператор **foreach** забезпечує простий і зрозумілий спосіб ітерації за елементами масиву або будь-якої перелічуваної колекції.

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

int[] numbers = { 4, 5, 6, 1, 2, 3, -2, -1, 0 };

foreach (int i in numbers)

{

System.Console.Write("{0} ", i);

}

// Output: 4 5 6 1 2 3 -2 -1 0

Для багатовимірних масивів елементи опрацьовуються так, що спочатку збільшуються індекси крайнього правого виміру, а потім - наступного лівого і так далі вліво:

int[,] numbers2D = new int[3, 2] { { 9, 99 }, { 3, 33 }, { 5, 55 } };

// Or use the short form:

// int[,] numbers2D = { { 9, 99 }, { 3, 33 }, { 5, 55 } };

foreach (int i in numbers2D)

{

System.Console.Write("{0} ", i);

}

// Output: 9 99 3 33 5 55

Проте для управління порядком оброблюваних елементів у багатовимірних масивах можна використовувати вкладений цикл for.

**Передача масивів як аргументів**

Масиви можна передавати як аргументи в параметри методу. Оскільки масиви являють собою посилальні типи, метод може змінювати значення елементів.

**Передача одновимірних масивів як аргументів**

Ініціалізований одновимірний масив можна передати в метод. Наприклад, наступний оператор передає масив у метод друку:

int[] theArray = { 1, 3, 5, 7, 9 };

PrintArray(theArray);

У наступному прикладі коду показана роздільна реалізація методу друку:

void PrintArray(int[] arr)

{

// Method code.

}

Новий масив можна ініціалізувати та передати за один крок, як показано в наступному прикладі:

PrintArray(new int[] { 1, 3, 5, 7, 9 });

У наступному прикладі масив рядків ініціалізовано і передано як аргумент у метод ***DisplayArray*** для рядків. Цей метод відображає елементи масиву. Потім метод **ChangeArray** розміщує елементи масиву в зворотному порядку, а метод ***ChangeArrayElements*** змінює перші три елементи масиву. Після повернення кожного методу метод ***DisplayArray*** показує, що передача масиву за значенням не перешкоджає зміні елементів масиву:

using System;

class ArrayExample

{

static void DisplayArray(string[] arr) => Console.WriteLine(string.Join(" ", arr));

// Change the array by reversing its elements.

static void ChangeArray(string[] arr) => Array.Reverse(arr);

static void ChangeArrayElements(string[] arr)

{

// Change the value of the first three array elements.

arr[0] = "Mon";

arr[1] = "Wed";

arr[2] = "Fri";

}

static void Main()

{

// Declare and initialize an array.

string[] weekDays = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

// Display the array elements.

DisplayArray(weekDays);

Console.WriteLine();

// Reverse the array.

ChangeArray(weekDays);

// Display the array again to verify that it stays reversed.

Console.WriteLine("Array weekDays after the call to ChangeArray:");

DisplayArray(weekDays);

Console.WriteLine();

// Assign new values to individual array elements.

ChangeArrayElements(weekDays);

// Display the array again to verify that it has changed.

Console.WriteLine("Array weekDays after the call to ChangeArrayElements:");

DisplayArray(weekDays);

}

}

// The example displays the following output:

// Sun Mon Tue Wed Thu Fri Sat

//

// Array weekDays after the call to ChangeArray:

// Sat Fri Thu Wed Tue Mon Sun

//

// Array weekDays after the call to ChangeArrayElements:

// Mon Wed Fri Wed Tue Mon Sun

**Передача багатовимірних масивів як аргументів**

Ініціалізований багатовимірний масив можна передати в метод так само, як і одновимірний масив:

int[,] theArray = { { 1, 2 }, { 2, 3 }, { 3, 4 } };

Print2DArray(theArray);

У наступному коді показано розділюване оголошення методу друку, який приймає як аргумент двовимірний масив:

void Print2DArray(int[,] arr)

{

// Method code.

}

Новий масив можна ініціалізувати і передати за один крок, як показано в наступному прикладі:

Print2DArray(new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } });

У наступному прикладі ініціалізується двовимірний масив цілих чисел, який передається в метод ***Print2DArray***. Цей метод відображає елементи масиву.

class ArrayClass2D

{

static void Print2DArray(int[,] arr)

{

// Display the array elements.

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

{

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

{

System.Console.WriteLine("Element({0},{1})={2}", i, j, arr[i, j]);

}

}

}

static void Main()

{

// Pass the array as an argument.

Print2DArray(new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } });

// Keep the console window open in debug mode.

System.Console.WriteLine("Press any key to exit.");

System.Console.ReadKey();

}

}

/\* Output:

Element(0,0)=1

Element(0,1)=2

Element(1,0)=3

Element(1,1)=4

Element(2,0)=5

Element(2,1)=6

Element(3,0)=7

Element(3,1)=8

\*/