

Міністерство освіти і науки України
Національний університет «Чернігівський колегіум» імені Т.Г. Шевченка
Навчально-науковий інститут професійної освіти та технологій
Кафедра технологічної освіти та інформатики

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 6
з дисципліни "Програмування"

Тема: **Масиви та рядки**

Виконав: студент 24 групи

Доля А.М.

Перевірив: викладач

Горчинський С.В.

Чернігів 2024

Відповіді на контрольні питання

1. Що таке масив.

Масив представляє собою сукупність змінних одного типу з загальним ім'ям для доступу до них. У С# масиви можуть бути як одновимірними, так і багатовимірними, хоча найчастіше використовуються одновимірні масиви. Масиви використовуються для різних цілей, оскільки вони надають зручні засоби для об'єднання пов'язаних змінних. Наприклад, у масиві можна зберігати максимальні денні температури, зареєстровані протягом місяця, перелік біржових курсів або назви книг з програмування з домашньої бібліотеки. Основна перевага масиву полягає в організації даних таким чином, щоб ними було легше маніпулювати.

Крім того, масиви дозволяють організувати дані таким чином, щоб легко відсортувати їх. Масивами в С# можна користуватися практично так само, як і в інших мовах програмування. Тим не менш у них є одна особливість: вони реалізовані у вигляді об'єктів.

2. Як проходить процес ініціалізації масиву.

```
// Обчислити середнє арифметичне ряду значень.
```

```
using System;
```

```
class Avarage
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        int[] nums = new int[10];
```

```
        int avg = 0;
```

```
        nums[0] = 99;
```

```
        nums[1] = 10;
```

```
        nums[2] = 100;
```

```
        nums[3] = 18;
```

```
        nums[4] = 78;
```

```
        nums[5] = 23;
```

```

        nums[6] = 63;

        nums[7] = 9;

        nums[8] = 87;

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

            avg = avg + nums[i];

        avg = avg / 10;

        Console.WriteLine("Середнє: " + avg);

    }
}

```

У вищезазначеній програмі початкові значення були задані для елементів масиву `nums` вручну в десяти окремих операторах присвоєння. Звичайно, така ініціалізація масиву абсолютно правильна, але те саме можна зробити набагато простіше. Адже масиви можуть ініціалізуватися при їх створенні. Нижче наведена загальна форма ініціалізації одновимірного масиву:

```
тип[] ім'я_масиву = {val1, val2, val3, ..., valN};
```

де `val1-valN` означають початкові значення, які присвоюються по черзі, зліва направо і за порядком індексування. Для зберігання ініціалізаторів масиву в C# автоматично розподіляється достатня кількість пам'яті. А необхідність користуватися оператором `new` явно відпадає само собою. Як приклад нижче наведено поліпшену версію програми, що обчислює середнє арифметичне:

```

// Обчислити середнє арифметичне ряду значень.

using System;

class Avarage
{
    static void Main()
    {
        int[] nums = { 99, 10, 100, 18, 78, 23, 63, 9, 87, 49 };

        int avg = 0;

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

```

```
        avg = avg + nums[i];

        avg = avg / 10;

        Console.WriteLine("Середнє: " + avg);

    }

}
```

3. Як дотримуватися меж масиву.

Межі масиву в C# строго дотримуються. Якщо межі масиву не досягаються або перевищуються, то виникає помилка при виконанні. Щоб переконатися в цьому, спробуйте виконати наведену нижче програму, в якій навмисно перевищуються межі масиву.

```
// Показати перевищення меж масиву.

using System;

class ArrayErr
{
    static void Main()
    {
        int[] nums = { 99, 10, 100, 18, 78, 23, 63, 9, 87, 49 };

        int[] sample = new int[10];

        int i;

        // Відтворити перевищення меж масиву.

        for (i = 0; i < 100; i = i + 1)

            sample[i] = i;

    }

}
```

Як тільки значення змінної *i* досягає 10, виникне виняткова ситуація типу `IndexOutOfRangeException`, пов'язана з виходом за межі індексування масиву, і програма передчасно завершиться.

4. Як працює оператор `foreach`.

Оператор `foreach` використовується для циклічного звернення до елементів колекції, що представляє собою групу об'єктів. У C# визначено кілька видів колекцій, кожна з яких є масивом. Нижче наведена загальна форма оператора циклу `foreach`.

`foreach (тип ім'я_змінної_циклу in колекція) оператор;`

Тут тип `ім'я_змінної_циклу` означає тип і ім'я змінної управління циклом, яка отримує значення наступного елемента колекції на кожному кроці виконання циклу `foreach`. Колекція означає циклічно опитувану колекцію, яка тут і далі представляє собою масив. Отже, тип змінної циклу повинен відповідати типу елемента масиву. Крім того, тип може позначатися ключовим словом `var`. У цьому випадку компілятор визначає тип змінної циклу, виходячи з типу елемента масиву. Це може бути корисним для роботи з певного роду запитам, як буде показано далі. Але, як правило, тип вказується явно.

```
// Використання оператора циклу foreach.
```

```
using System;
```

```
class ForeachDemo
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        int[] nums = { 99, 10, 100, 18, 78, 23, 63, 9, 87, 49 };
```

```
        int sum = 0;
```

```
        int[] nums = new int[10];
```

```
        // Задати початкові значення елементів масиву nums.
```

```
        for (int i = 0; i < 10; i++)
```

```
            nums[i] = i;
```

```
        // Використовувати цикл foreach для виведення значень
```

```
        // елементів масиву та підрахунку їх суми.
```

```
        foreach (int x in nums)
```

```
        {
```

```
            Console.WriteLine("Значення елемента дорівнює: " + x);
```

```
        sum += x;

    }

    Console.WriteLine("Сума дорівнює: " + sum);

}

}
```

Виконання наведеного вище коду дає наступний результат.

Значення елемента дорівнює: 0

Значення елемента дорівнює: 1

Значення елемента дорівнює: 2

Значення елемента дорівнює: 3

Значення елемента дорівнює: 4

Значення елемента дорівнює: 5

Значення елемента дорівнює: 6

Значення елемента дорівнює: 7

Значення елемента дорівнює: 8

Значення елемента дорівнює: 9

Сума дорівнює: 45

Висновки: виконуючу дану лабораторну роботу ми почали вивчати масиви та рядки на C#.


Ознайомившись з методичним посібником, а саме з темою №6, в якому ми вивчили та повторили що таке: масиви; одновимірні масиви; ініціалізація масиву; дотримання меж масиву; оператор циклу foreach.

Також ми виконали чотири практичних завдання пов'язаних із: створення масиву чисел та виведенням їх в консоль; додали можливість задати фіксовану кількість повторів; додали можливість обчислення суми усіх чисел масиву; розробили консольну програму, яка сортує введений масив за зростанням. В ході роботи ми використали масиви, з якими перед цим ознайомились в методичному посібнику. На початку роботи ми виконали безкоштовне практичне завдання від Microsoft.

Отже, ознайомившись з методичним посібником та виконавши всі завдання до лабораторної роботи, ми отримали потрібні нам знання про масиви та рядки. Тому ми можемо продовжувати вивчати таку мову програмування як C#.

Скріншоти:

1) Початок:



900 XP

Зберігайте та переглядайте послідовності даних за допомогою масивів і оператора foreach у C#

51 хв • Модуль • 7 одиниць

[Зворотній зв'язок](#)

Початківець

Розробник

Вихователь вища

К-12 Вихователь

студент

.NET

Код Visual Studio

Робота з послідовностями пов'язаних даних у структурах даних, відомих як масиви. Потім навчіться повторювати кожен елемент у послідовності.

Мета навчання

У цьому модулі ви:

- Створіть та ініціалізуйте новий масив.
- Встановлення та отримання значень у масивах.
- Ітерація по кожному елементу масиву за допомогою оператора foreach.

старт >

+ додати

2) Блок 1:

Вступ

✓ 100 XP

3 хвилини

Масиви C# дозволяють зберігати послідовності значень в одній структурі даних. Іншими словами, уявіть одну змінну, яка може містити багато значень. Якщо у вас є єдина змінна, яка зберігає всі значення, ви можете сортувати значення, змінювати порядок значень на протилежний, переглядати кожне значення та перевіряти його окремо тощо.

Припустимо, що ви працюєте у відділі безпеки компанії, яка підбирає онлайн-продавців із комісійними рекламодавцями. Вас попросили написати код C#, який перебиратиме ідентифікатори вхідних замовлень. Вам потрібно перевірити кожен ідентифікатор замовлення, щоб визначити замовлення, які можуть бути шахрайськими. Щоб виконати це програмне завдання, вам потрібно реалізувати масиви.

У цьому модулі ви будете створювати та ініціалізувати масиви. Ви встановлюєте та отримуєте значення з елементів масиву, звертаючись до кожного елемента за допомогою його індексу. Ви створите циклічну логіку, яка дозволить вам працювати з кожним елементом у масиві.

До кінця цього модуля ви попрацюєте зі своєю першою структурою для зберігання кількох значень даних. Пізніше, в інших модулях, ви дізнаєтеся, як сортувати, фільтрувати, запитувати, агрегувати та виконувати інші операції з вашими даними.

3) Блок 2:

```
using System.Text;

namespace App09
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.OutputEncoding = Encoding.UTF8;

            string[] fraudulentOrderIDs = { "A123", "B456", "C789" };

            Console.WriteLine($"Перший: {fraudulentOrderIDs[0]}");
            Console.WriteLine($"Другий: {fraudulentOrderIDs[1]}");
            Console.WriteLine($"Третій: {fraudulentOrderIDs[2]}");

            fraudulentOrderIDs[0] = "F000";

            Console.WriteLine($"Перепризначити спочатку: {fraudulentOrderIDs[0]}");

            Console.WriteLine($"Існує {fraudulentOrderIDs.Length} шахрайських замовлень для обробки.");
        }
    }
}
```

Microsoft Visual Studio Debug Console

```
Перший: A123
Другий: B456
Третій: C789
Перепризначити спочатку: F000
Існує 3 шахрайських замовлень для обробки.
```


Перевір свої знання

1. Що таке масив? *

☐ Рядкова змінна

☒ Послідовність окремих елементів даних, доступна через одне ім'я змінної.

✓ Correct.

☐ Бібліотека класів .NET.

2. Що з цього є правильним прикладом створення масиву та його ініціалізації? *

☐ `string[] myarray = new string[3]; myarray = "test1"; myarray = "test2"; myarray = "test3";`.

☐ `string[] myarray = string[3]; myarray[0] = test1; myarray[1] = test2; myarray[2] = test3;`.

☒ `int[] myarray = new int[3]; myarray[0] = 1; myarray[1] = 2; myarray[2] = 3;`.

✓ Correct!

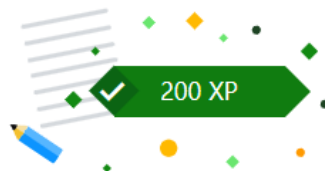
element

Наступний

Продовжити >

Щиро вітаю!

Ви заробили бали досвіду за проходження перевірки знань.



Продовжити

Увійдіть, щоб зберегти прогрес

Перегляньте відповіді

1. Що таке масив? *

☐ Рядкова змінна.

☒ Послідовність окремих елементів даних, доступна через одне ім'я змінної.

✓ Correct. An array data structure with a single name and multiple data locations.

☐ Бібліотека класів .NET.

2. Що з цього є правильним прикладом створення масиву та його ініціалізації? *

☐ `string[] myarray = new string[3]; myarray = "test1"; myarray = "test2"; myarray = "test3";`.

☐ `string[] myarray = string[3]; myarray[0] = test1; myarray[1] = test2; myarray[2] = test3;`.

☒ `int[] myarray = new int[3]; myarray[0] = 1; myarray[1] = 2; myarray[2] = 3;`.

✓ Correct! The array `myarray` is declared correctly and the code statements used to assign values to the array elements specify the array index locations.

4) Блок 3:

```
using System.Text;

namespace App10
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.OutputEncoding = Encoding.UTF8;

            int[] inventory = { 200, 450, 700, 175, 250 };
            int sum = 0;
            int bin = 0;
            foreach (int items in inventory)
            {
                sum += items;
                bin++;
                Console.WriteLine($"Кошик {bin} = {items} елементів (Загальна сума: {sum})");
            }
            Console.WriteLine($"Ми маємо {sum} предметів в інвентарі.");
        }
    }
}
```

Microsoft Visual Studio Debug Console

```
Кошик 1 = 200 елементів (Загальна сума: 200)
Кошик 2 = 450 елементів (Загальна сума: 650)
Кошик 3 = 700 елементів (Загальна сума: 1350)
Кошик 4 = 175 елементів (Загальна сума: 1525)
Кошик 5 = 250 елементів (Загальна сума: 1775)
Ми маємо 1775 предметів в інвентарі.
```

5) Блок 4:

Вправа. Виконайте завдання для вкладених операторів ітерації та вибору

✓ 100 XP

9 хвилин

Випробування коду зміцнюють те, що ви дізналися, і допомагають вам набути певної впевненості, перш ніж продовжувати.

Виклик шахрайського замовлення

Раніше в цьому модулі ви збиралися написати код, який зберігатиме ідентифікатори замовлень, що належать до потенційно шахрайських замовлень. Ваша мета — якомога раніше знаходити шахрайські замовлення та позначати їх для глибшого аналізу.

6) Блок 5:

```
using System.Text;

namespace App11
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.OutputEncoding = Encoding.UTF8;

            string[] orderIDs = { "B123", "C234", "A345", "C15", "B177", "G3003", "C235", "B179" };

            foreach (string orderID in orderIDs)
            {
                if (orderID.StartsWith("B"))
                {
                    Console.WriteLine(orderID);
                }
            }
        }
    }
}
```

Microsoft Visual Studio Debug Console

```
B123
B177
B179
```

7) Блок 6:

Перевірка знань

3 хвилини

Перевір свої знання

1. Що з наведеного є масивом?

☐ myArray[3]

☒ myArray[2]

✓ Correct!

☐ myArray{3}

2. Який із наведених типів є масивом? *

☒ int[] myArray

✓ Correct!

☐ int[] myArray

☐ int myArray

3. Яке з наведених кодіків є правильним?

☒ foreach (int value in values).

✓ Correct! This foreach statement is defined using valid syntax.

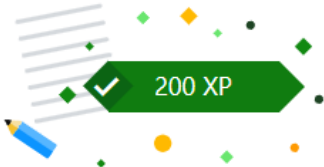
☐ foreach (int value of values).

☐ foreach (int value with values).

✓ 200 XP

Щиро вітаю!

Ви заробили бали досвіду за проходження перевірки знань.



Продовжити

Увійдіть, щоб зберегти прогрес

Перегляньте відповіді

1. Що з наведеного нижче можна використати для доступу до третього елемента в масиві? *

☐ `myArray[3].`

☒ `myArray[2].`

✓ Correct! Indexes are zero-based, so an index of two returns the third element in an array.

☐ `myArray{3}.`

2. Який із наведених варіантів описує правильний спосіб створення нового цілочисельного масиву з трьох елементів? *

☒ `int[] myArray = new int[3];.`

✓ Correct! This is the correct way to create an array with three elements.

☐ `int[] myArray = new int[2];.`

☐ `int myArray = new int[3];.`

3. Яке з наведених `foreach` тверджень є синтаксично правильним? *

☒ `foreach (int value in values).`

✓ Correct! This `foreach` statement is defined using valid syntax.

☐ `foreach (int value of values).`

☐ `foreach (int value with values).`

8) Блок 7, кінець:

Резюме

✓ 100 XP

2 хвилини

Ваша мета в цьому модулі полягала в програмній роботі з послідовністю ідентифікаторів замовлень, щоб ідентифікувати потенційно шахрайські замовлення. Ви створили масив ідентифікаторів замовлень, а потім повторили кожен елемент послідовності, шукаючи загальну ознаку шахрайства.

Масиви дозволяють зберігати кожен ідентифікатор замовлення як елемент однієї змінної. Ви звернулися до певного елемента масиву за допомогою індексу — числового значення від нуля. Ви змогли отримати та встановити значення кожного елемента. Ви перебирали елементи масиву, щоб перевірити або вивести значення кожного елемента.

Уявіть, як важко було б обробляти пов'язані дані у вашому коді, якби у вас *не* було такої структури, як масив? Вам потрібно заздалегідь знати, скільки елементів даних ви плануєте працювати, і визначити окрему змінну для кожного значення. Це створило б крихке рішення.

Під час створення масивів можна змінювати розмір відповідно до обсягу даних, з якими вам потрібно працювати. Коли ви ще більше ознайомитеся з масивами (та подібними структурами даних), ви часто будете використовувати їх у своїх програмах для обробки та керування даними.