

Лабораторна робота №4

Робота з масивами. Створення ASP.NET Web Forms-застосування

Мета роботи: розробка додатків на мові C# у інтегрованій середі розробки Microsoft Visual Studio.Робота з масивами.

Постановка завдання: створити в Microsoft Visual Studio Web-додаток, що містить форму з заданою кількістю полів для введення значень елементів числового або символьного масиву, та скласти і відлагодити програму (мовою C#) для обробки елементів масиву, згідно з варіантом завдання.

Теоретичні відомості

Масив, це послідовність однотипних елементів, упакована під одним ім'ям. Ці дані легко відсортувати, перебрати і обробляти. Масив в C# є об'єктом посилального типу. Створенням масиву є двоступінчатий процес: спочатку оголошується посилальна змінна на масив, а потім для нього виділяється пам'ять і змінною привласнюється посилання на цю пам'ять.

Одновимірні масиви

Так для одновимірного масиву синтаксис створення буде таким:

```
тип[ ] ім'я_масиву = new тип[розмір];
```

Наприклад, зарезервувати пам'ять під 10 елементів цілого типу можна так:

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

Адресація елементів усередині масиву виконується за допомогою цілочисельного індексу, нумерація якого починається з нуля. Вихід індексу за межі меж компілятора не контролюється і виявляється середовищем CLR

Якщо масив оголошується як поле класу, то всі елементи за умовчанням набувають найближчого за типом значення (для числового - нуль, для строкового - порожньо, для булевого - false, для посилального - null). Якщо масив оголошується усередині методу як локальна змінна, то його перед використанням потрібно явно ініціалізувати.

Масив в C# – це тип, похідний від класу `System.Array`, тому всі масиви успадковують від цього класу фіксований набір об'єктних властивостей і методів. Крім того, сам клас `Array` містить ряд статичних методів (рівня класу), що дозволяють виконувати обробку масивів як об'єктів.

Одне і те ж посилання на масив можна використовувати багато разів при створенні декількох сумісних з її типом масивів. У такому разі адресація колишнього масиву буде загублена. Наприклад:

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

```
.....
```

```
array = new int[20];
```

Можна виконувати відразу і оголошення і ініціалізацію масиву, тоді створення масиву і підрахунок його розмірності виконає компілятор за списком ініціалізації. В цьому випадку ключове слово `new` не потрібне. Наприклад:

```
int[ ] array = { 1, 2, 3, 4, 5 };
```

Можна створити масив фіксованої розмірності і відразу його ініціалізувати. Тоді розмір списку ініціалізації повинен строго відповідати замовленій розмірності масиву. Наприклад:

```
Int[ ] array = new int[5]{ 1, 2, 3, 4, 5 };
```

Допустимий і такий синтаксис - без вказівки розмірності, але з ініціалізацією:

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

Можна спочатку створити масив, а потім поелементно привласнити йому значення. І теж потрібно стежити, щоб не вийти за межі встановленої розмірності масиву.

```
int[ ] array = new int[5];
```

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

```
    array[i]= i;
```

Елементами масиву можуть бути значення довільного типу, зокрема масиви, класи, структури і інтерфейси. Масиви можуть бути як одновимірними, так і багатовимірними.

Приклад створення масиву, що складається з дванадцяти елементів типу `string`:

```
String[ ] st = new string[12];
```

При створенні масивів необхідно враховувати, що для створення масиву фіксованої довжини, як в попередніх прикладах, необхідно завжди

використовувати ключове слово `new`. Таким чином, наступне визначення масиву неприпустимо і приведе до виникнення помилки ще на етапі компіляції: `int[3] a={3,5,6};`

Багатовимірні масиви

Окрім простих одновимірних масивів `C#` підтримує також багатовимірні масиви, які у свою чергу діляться на дві категорії. До першої категорії відносяться масиви, кількість елементів кожного рядка яких складається з однакової кількості елементів. Таким чином, масив можна розглядати як прямокутник, а самі такі масиви називають "прямокутними". До другої категорії відносяться масиви, кількість елементів в рядках у яких не однаково. Такі масиви утворюють прямокутник, у якого одна сторона представляє ламану лінію, тому такі масиви називають "ламаними".

Оголошення і заповнення масиву проводиться стандартним способом: для цього організовується два цикли:

```
Random rnd=new Random();
int[,] Matrix;
Matrix = new int[5, 4];
for (int i = 0; i < 5; i++)
    for (int j=0;j<4;j++)
        Matrix[i,j]= rnd.Next(10,99);
```

При необхідності виведення значень елементів масиву на екран також організовується цикл, що дозволяє послідовно перебирати всі елементи масиву:

```
for (int i = 0; i < 5; i++)
{
    for (int j = 0; j < 4; j++)
    {
        Response.Write(Matrix[i, j].ToString());
        Response.Write(" ");
    }
    Response.Write("<br/>");
}
```

"Ламаний" масив може розглядатися як масив, кожен осередок якого є масивом. Як приклад створимо масив із змінною кількістю елементів в рядках:

```
Random rnd = new Random();
```

```
int[][] JMatrix = new int[5][];
for (int i = 0; i < JMatrix.Length; i++)
{
    JMatrix[i] = new int[rnd.Next(1,7)];
}
```

Кількість елементів в рядку визначається випадковим чином у момент формування масиву. Кількість рядків задана жорстко і рівна п'яти. Масив не заповнюється ніякими числами, тому на екран виводитимуться значення, якими заповнюється масив за умовчанням (в даному випадку це нуль). Для виведення інформації на екран необхідно використовувати два цикли, як це показано нижче.

```
for (int i = 0; i < JMatrix.Length; i++)
{
    Response.Write("Кількість елементів в рядку " +
        i.ToString() + "=" + JMatrix[i].Length.ToString()+" ");
    for (int j = 0; j < JMatrix[i].Length; j++)
    {
        Response.Write(JMatrix[i][j].ToString() + " ");
    }
    Response.Write("<br/>");
}
```

У циклі по j відбувається визначення кількості елементів в рядку за допомогою властивості Length.

Клас Array

Для детальнішого розуміння особливостей використання масивів в C# необхідно розглянути спеціалізований клас, що реалізовує функції масиву. Всі типи даних в C# є класами, для яких як базовий виступає клас Object.

Клас Array - не виключення. Він реалізує всі базові властивості класів і є предком для всіх типів масивів, до яких ми звикли в мові C++ і синтаксис опису яких був приведений вище. Те, що клас Array є нащадком класу Object, дає можливість в класі Array визначати безліч різноманітних операцій, таких як копіювання, пошук, звернення, сортування і т. д. Нижче приведені найцікавіші методи класу Array:

BinarySearch()	Пошук елементів в одновимірному відсортованому масиві
----------------	---

Sort()	Сортування елементів одновимірного масиву
Clear()	Очищення елементів масиву в заданому діапазоні індексів
CopyTo()	Копіювання елементів початкового масиву в масив призначення
GetLength(), Length	Визначення кількості елементів у вказаному вимірюванні масиву
GetLowerBound()	Визначення нижньої межі масиву
GetUpperBound()	Визначення верхньої межі масиву
GetValue()	Повертає значення вказаного індексу для масиву
SetValue()	Встановлює значення вказаного індексу для масиву
Reverse()	Розставляє елементи одновимірного масиву в зворотному порядку
Rank	Визначення кількості вимірювань вказаного масиву

Розглянемо приклад використання класу `Array`. Для цього створимо масив і забезпечимо можливість пошуку в ньому елементів.

Оголосимо масив `myArray` як статичний член класу `Page`, що складається з шести елементів типу `int`:

```
static Array myArray = Array.CreateInstance(typeof(Int32), 6);
```

Розмістимо на формі елементи `TextBox` і `Button`, яким привласнимо імена `tb_value` і `btn_find` відповідно.

У обробник процедури натиснення на кнопку введемо наступний код, що заповнює масив випадковими числами, сортує його і здійснюючий пошук введеного в текстове поле елементу. Код процедури натиснення на кнопку приведений нижче.

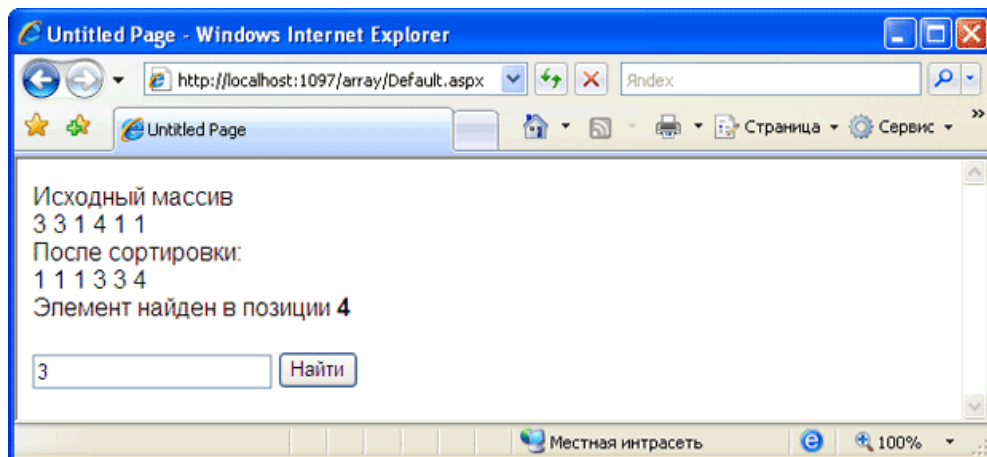
```
protected void btn_find_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    Response.Write("Початковий масив <br/>");
    for (int i = myArray.GetLowerBound(0); i <=
        myArray.GetUpperBound(0); i++)
    {
        myArray.SetValue(rnd.Next(1, 10), i);
        Response.Write(myArray.GetValue(i) + "\t");
    }
}
```

```

Response.Write("<br/>");
Array.Sort(myArray);
Response.Write("Після сортування:<br/>");
for (int i = myArray.GetLowerBound(0); i <=
myArray.GetUpperBound(0); i++)
{
    Response.Write(myArray.GetValue(i) + "\t");
}
object npo = Convert.ToInt32(tb_value.Text);
int findIndex = Array.BinarySearch(myArray, npo);
if (findIndex<0)
{
    Response.Write("<br/>Елемент не знайдений");
}
else
{
    Response.Write("<br/>Елемент знайдений у позиції <b>" + findIndex.
ToString() + "</b>");
}
}
}

```

Приклад роботи програми в результаті виконання приведеної вище коди представлений на мал. 7.



Результат роботи програми заповнення, сортування і пошуку елементів масиву

Як видно, працювати з масивами в С# досить просто, особливо враховуючи досить великі можливості класу Array.

Варіанти завдань

Створити в Microsoft Visual Studio форму, що містить задану кількість полів для введення значень елементів числового або символьного масиву, і кнопку, при натисненні на яку виконується обробка масиву згідно варіанту завдання.

Завдання 1

1. Задати масив, що складається з 10 чисел. Знайти найменший з позитивних елементів масиву.
2. Задати масив з 10 елементів, що містить назви країн. Підрахувати кількість елементів масиву, що починаються на літеру «А».
3. Задати масив, що складається з 10 цілих чисел. Підрахувати кількість невід'ємних парних елементів масиву.
4. Задати масив, що складається з 10 цілих чисел. Обчислити добуток всіх непарних від'ємних елементів масиву.
5. Задати масив з 10 елементів, що містить чоловічі і жіночі імена. Підрахувати кількість букв в щонайдовшому імені.
6. Задати масив, що складається з 10 чисел. Обчислити суму елементів масиву, які більші заданого числа.
7. Задати масив, що складається з 10 цілих чисел. Обчислити добуток всіх парних елементів масиву, не менших заданого числа.
8. Задати масив з 10 елементів, що містить назви рослин. Підрахувати, скільки разів у всіх елементах масиву зустрічається літера «ь».
9. Задати масив, що складається з 10 чисел. Визначити номер останнього невід'ємного елемента.
10. Задати масив, що складається з 10 цілих чисел. Знайти найменший з парних елементів масиву.
11. Задати масив з 10 елементів, що містить назви міст. Підрахувати кількість елементів масиву, що починаються на приголосну літеру.
12. Задати масив, що складається з 10 чисел. Обчислити суму елементів масиву, що мають парні номери.
13. Задати масив з 10 елементів, що містить назви тварин. Підрахувати середню довжину елементів масиву.
14. Задати масив, що складається з 10 чисел. Підрахувати кількість елементів масиву, менших заданого числа.

15. Задати масив з 10 елементів, що містить назви фруктів. Підрахувати кількість елементів масиву, що не містять літери «ш».

Завдання 2

1. Дано одновимірний масив. Скласти програму для визначення квадратного кореня з кожного елемента масиву.

2. Дано одновимірний масив. Збільште кожен його елемент удвічі.

3. У заданому двовимірному масиві відсортуйте елементи кожного стовпця за зростанням.

4. У заданому двовимірному масиві відсортуйте елементи кожного рядка за зростанням.

5. Дано двовимірний масив. Вивести на консоль значення правого нижнього елемента.

6. Дано квадратний двовимірний масив. Вивести на консоль сему елементів головної діагоналі масиву.

7. Дано квадратний двовимірний масив. Вивести на консоль сему елементів побічної діагоналі масиву.

8. Заповнити масив розмірністю [6,5] добутком номера стовпця на подвоєний квадратний корінь номера рядка.

9. З двовимірного масиву надрукувати на екрані в стовпчик найбільший елемент кожного рядка.

10. Дано двовимірний масив. Вивести на екран суму всіх елементів третього рядка.

11. Дано двовимірний масив. Вивести на консоль значення лівого нижнього елемента.

12. Дано квадратний двовимірний масив. Вивести на консоль сему елементів масиву без головної діагоналі.

13. Дано квадратний двовимірний масив. Вивести на консоль сему елементів без побічної діагоналі масиву.

14. Заповнити масив розмірністю [6,5] коренем добутку номера стовпця на номер рядка.

15. Для двовимірного масиву надрукувати на екрані в стовпчик найбільший елемент кожного стовпця.

Контрольні за питання

1. Дайте визначення масиву.
2. Як можна виконувати відрізу і оголошення і ініціалізацію масиву у програмі?
3. Що таке розмірність масиву, які масиви бувають у C# ?
4. Як одержати доступ до певного елемента масиву?
5. Як у програмі організувати введення й вивід масиву?
6. Які основні методи виконує Клас Array?
7. Які операції дозволено над елементами масиву?

ЛІТЕРАТУРА

Основна

1. Эспозито Д. Microsoft ASP.NET 2.0. Базовый курс. Мастер-класс / Пер. с англ. – М.: Издательство «Русская редакция»; СПб.: Питер, 2007. – 688 с.
2. Браст Эндрю Дж., Форте Стивен. Разработка приложений на основе Microsoft SQL Server. Мастер-класс. / Пер. с англ. – М.: Издательство «Русская редакция», 2007. – 880 с.

Додаткова

1. Канноли Томас, Бегг Каролин, Страчан Анна, Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: Пер.с англ.: Уч. пос.- М.: Издательский дом «Вильямс», 2000г.
2. Троелсен Эндрю. Язык программирования C# и платформа .NET 2.0, 3-е издание.: Пер. с англ.-М.:ООО «И.Д.Вильямс»,2007. -1168с.
3. Мак-Дональд Мэтью,Шпунта Марио. Microsoft ASP.NET 2.0 с примерами на C# для профессионалов.:Пер. с англ.-М.:ООО «И.Д.Вильямс»,2006. -1408с.
4. Эспозито Д. Microsoft ASP.NET 2.0. Углубленное изучение. Мастер-класс / Пер. с англ. – М.: Издательство «Русская редакция»;СПб.:Питер, 2007.– 688 с.
5. Рейли Д. Создание приложений Microsoft ASP.NET / Пер. с англ. – М.: Издательство «Русская редакция», 2002. – 480 с
6. Харинатх С., Куинн С , SQL Server Analysis Services и MDX для профессионалов «Диалектика»,2008 г.-848с.