



البرمجة الإجرائية Procedural Programming

IPG202

الفصل الثاني أنواع البيانات المركبة : السلاسل المحرفية Compound Data Types: Strings

الكلمات المفتاحية

بنية مركبة، مصفوفة، مصفوفة محارف، سلسلة محرفية، string.

ملخص الفصل

يركز هذا الفصل على استخدام السلاسل المحرفية، حيث يستفيد من المفاهيم المعروضة في فصل المصفوفات لعرض التمثيل الداخلي للسلسلة المحرفية، ومن ثم يقدم استعراضاً للصنف string مسبق التعريف في لغة C# ويسرد أهم الطرائق الجاهزة للتعامل مع القيم النصية.

أهداف الفصل

بنهاية هذا الفصل سيكون الطالب قادراً على:

- معرفة طريقة تحقيق السلاسل النصية بلغة C#.
- التصريح عن سلسلة نصية.
- استخدام التوابع مسبقة التعريف الخاصة بالصنف string.

محتويات الفصل

1. مقدمة
2. تحقيق السلاسل باستخدام المصفوفات في C#.
3. الصنف string وطرائقه.
4. أمثلة حول استخدام طرائق الصنف string.

1- مقدمة.

السلسلة المحرفية هي عبارة عن تتال من المحارف يستخدم للتعبير عن قيمة نصية، حيث يمكن أن تكون مؤلفة من محرف واحد أو عدة محارف أو عدة كلمات.

بالنظر لخصوصية هذا النمط من البيانات فقد تضمنت لغة C# تعريفاً مسبقاً لنمط بيانات خاص بالتعامل معها يدعى باسم string (وقد سبق لنا استخدامه مراراً)، كما تضمنت مجموعة غنية من الطرائق مسبقاً التعريف الخاصة بإجراء أنماط معالجات مختلفة على هذه القيم.

2- تحقيق السلاسل باستخدام المصفوفات في C#.

يمكن النظر إلى السلسلة المحرفية على أنها مصفوفة أحادية البعد كل عنصر من عناصرها حول عبارة عن قيمة من النوع المحرفي char، وبالتالي ينطبق عليها كل ماتعلمناه في الفصل السابق حول التعامل مع المصفوفات.

التصريح عن مصفوفة محارف

يمكن باعتماد الأسلوب السابق نفسه، إنشاء مصفوفة محارف كما يلي:

```
char [ ] Course;  
char [ ] Course = new char [20];  
char [ ] Course = {'P','r','o','g','r','a','m','m','i','n','g'};
```

الوصول إلى عناصر مصفوفة المحارف والعمليات عليها

يمكن تمثيل السلسلة السابقة Course كما يلي:

Index	0	1	2	3	4	5	6	7	8	9	10
Course value	P	r	o	g	r	a	m	m	i	n	g

وبالتالي للوصول إلى أي عنصر، لابد من معرفة رقم فهرسه أو دليله.

مثال 1- التجول على مصفوفة المحارف Course وطباعتها على الشاشة

```
char[] Course = { 'P', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g' };  
for (int i = 0; i < Course.Length; i++)  
    Console.Write(Course[i]);  
Console.WriteLine();
```

مثال 2- استبدال كل محرف 'm' بالمحرف 'M' في مصفوفة المحارف Course

```
char[] Course = { 'P', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g' };  
for (int i = 0; i < Course.Length; i++)  
    if (Course[i] == 'm')  
        Course[i] = 'M';
```

3- الصنف String وطرائقه

بالنظر إلى مصفوفة المحارف، قد يبدو أمراً غير مرّن القيام بعمليات المعالجة للقيم النصية (وخاصة القيم النصية الطويلة) باستخدام عمليات مصفوفات مباشرة على هذه القيم.

تضمنت لغة C# الصنف مسبق التعريف string الذي قام بتغليف هذه البنية وتعريف كم هائل من الطرائق الشائعة للتعامل مع القيم من هذا النوع.

يمكن التعامل برمجياً مع القيم من النوع string كما يتم التعامل مع أي متحول من صنف بسيط.

مثال 1- عمليات التصريح والإدخال والإخراج:

```
string Course = "Programming";  
Console.WriteLine(Course);  
Course = Console.ReadLine();
```

مثال 2- عمليات الدمج والمقارنة:

```
string Course1 = "Programming";  
string Course2 = "Introduction To";  
string Course3 = Course1 + Course2;
```

```

if (Course1 == Course2)
    Console.WriteLine(Course1+" is identical to "+Course2);
else
    Console.WriteLine(Course1+" is not Identical to "+Course1);

```

ملاحظة: فقط الجمع + (من بين العمليات الحسابية)، والمساواة == (من بين عمليات المقارنة) يمكن استخدامها مع القيم من النمط string.

فيما يلي قائمة بأهم طرائق الصنف string المسبقة التعريف:

الشرح	الطريقة
تُستخدم للمقارنة بين السلاسل. تكون القيمة المعادة: <ul style="list-style-type: none"> • 0 إذا تساوت السلسلتين. • 1 إذا كانت السلسلة الأولى أكبر من الثانية. • -1 إذا كانت السلسلة الثانية أصغر من الثانية. تأخذ افتراضياً هذه الطريقة حالة الأحرف عند المقارنة. إذا أردنا تجاهل حالة الأحرف نضع القيمة true في المعامل الثاني.	Compare()
جمع سلسلتين نصيتين.	Concat()
إنشاء نسخة من سلسلة.	Copy()
تُستخدم للمقارنة بين السلاسل: تُعيد true إذا تساوت السلسلتين.	Equals()
عدد محارف السلسلة النصية	Length
اختبار انتهاء السلسلة بسلسلة أخرى	EndsWith()
إدراج سلسلة ضمن السلسلة اعتباراً من فهرس معين	Insert()
فهرس أول ظهور لسلسلة ضمن سلسلة أخرى	IndexOf
فهرس آخر ظهور لسلسلة ضمن سلسلة أخرى	LastIndexOf()
اختبار ابتداء السلسلة بسلسلة أخرى	StartsWith()
الحصول على سلسلة جزئية من السلسلة	Substring()
التحويل لأحرف صغيرة	ToLower()
التحويل لأحرف كبيرة	ToUpper()
حذف الفراغات الزائدة من بداية ونهاية السلسلة	Trim()
حذف الفراغات الزائدة من نهاية السلسلة	TrimEnd()
حذف الفراغات الزائدة من بداية السلسلة	TrimStart()

4- أمثلة حول استخدام طرائق الصنف String

المثال الأول : يبين المثال التالي استخدام بعض طرق السلاسل النصية:

```
// create some strings to work with
string s1 = "abcd";
string s2 = "ABCD";
string s3 = @"Liberty Associates, Inc.
           provides custom .NET development,
           on-site Training and Consulting";

int result; // hold the results of comparisons
// compare two strings, case sensitive
result = string.Compare(s1, s2);
Console.WriteLine("compare s1: {0}, s2: {1}, result: {2}\n", s1, s2, result);

result = string.Compare("Said", "Ahmad Said");
Console.WriteLine("compare Said, Ahmad Said, result: {2}\n", s1, s2, result);

// overloaded compare, takes boolean "ignore case"
//(true = ignore case)
result = string.Compare(s1,s2, true);
Console.WriteLine("compare insensitive\n");
Console.WriteLine("s4: {0}, s2: {1}, result: {2}\n", s1, s2, result);
// concatenation method
string s6 = string.Concat(s1,s2);
Console.WriteLine("s6 concatenated from s1 and s2: {0}", s6);

// use the overloaded operator
string s7 = s1 + s2;
Console.WriteLine("s7 concatenated from s1 + s2: {0}", s7);

// the string copy method
string s8 = string.Copy(s7);
Console.WriteLine("s8 copied from s7: {0}", s8);

// use the overloaded operator
string s9 = s8;
Console.WriteLine("s9 = s8: {0}", s9);

// three ways to compare.
Console.WriteLine("\nDoes s9.Equals(s8)?: {0}", s9.Equals(s8));

Console.WriteLine("Does Equals(s9,s8)?: {0}", string.Equals(s9,s8));

Console.WriteLine("Does s9==s8?: {0}", s9 == s8);
```

```
// Tow useful properties: the index and the length
Console.WriteLine( "\nString s9 is {0} characters long. ", s9.Length);

Console.WriteLine("The 5th character is {0}\n", s9[4]);

// test whether a string ends with a set of characters
Console.WriteLine("s3:{0}\nEnds with Training?: {1}\n",s3, s3.EndsWith("Training") );
Console.WriteLine("Ends with Consulting?: {0}",s3.EndsWith("Consulting"));
// return the index of the substring
Console.WriteLine("\nThe first occurrence of Training ");
Console.WriteLine( "in s3 is {0}\n", s3.IndexOf("Training"));
// insert the word excellent before "training"
string s10 = s3.Insert(101,"excellent ");
Console.WriteLine("s10: {0}\n",s10);
// you can combine the two as follows:
string s11 = s3.Insert(s3.IndexOf("Training"), "excellent ");
Console.WriteLine("s11: {0}\n",s11);
```

يكون ناتج التنفيذ:

```
C:\WINDOWS\system32\cmd.exe
s7 concatenated from s1 + s2: abcdABCD
s8 copied from s7: abcdABCD
s9 = s8: abcdABCD

Does s9.Equals(s8)? : True
Does Equals(s9,s8)? : True
Does s9==s8? : True

String s9 is 8 characters long.
The 5th character is A

s3:Liberty Associates, Inc.
    provides custom .NET development,
    on-site Training and Consulting
Ends with Training?: False

Ends with Consulting?: True

The first occurrence of Training
in s3 is 99

s10: Liberty Associates, Inc.
    provides custom .NET development,
    on-site Trexcellent aining and Consulting

s11: Liberty Associates, Inc.
    provides custom .NET development,
    on-site excellent Training and Consulting

Press any key to continue . . .
```

المثال الثاني : يبين المثال التالي طرق البحث ضمن السلاسل النصية:

```
string s1 = "One Two Three Four";
int ix;
// get the index of the last space
ix = s1.LastIndexOf(" ");

// get the last word.
string s2 = s1.Substring(ix + 1);

// set s1 to the substring starting at 0
// and ending at ix (the start of the last word)
// thus s1 has one two three
s1 = s1.Substring(0, ix);

// find the last space in s1 (after two)
ix = s1.LastIndexOf(" ");
// set s3 to the substring starting at
// ix, the space after "two" plus one more
// thus s3 = "three"
string s3 = s1.Substring(ix + 1);

Console.WriteLine("s2: {0}\ns3: {1}", s2, s3);
// reset s1 to the substring starting at 0
// and ending at ix, thus the string "one two"
s1 = s1.Substring(0, ix);

// reset ix to the space between "one" and "two"
ix = s1.LastIndexOf(" ");

// set s4 to the substring starting one space after ix, thus the substring "two"
string s4 = s1.Substring(ix + 1);

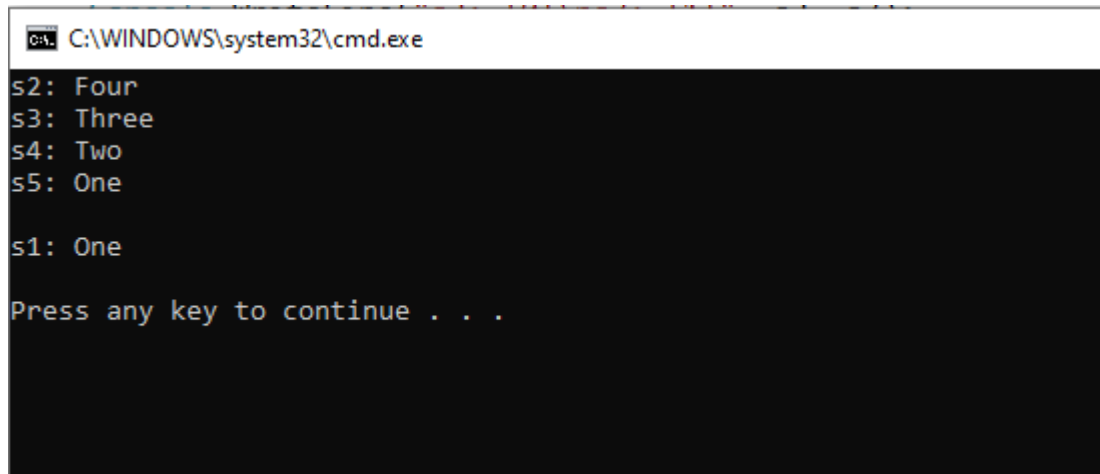
// reset s1 to the substring starting at 0 and ending at ix, thus "one"
s1 = s1.Substring(0, ix);

// set ix to the last space, but there is none so ix now = -1
ix = s1.LastIndexOf(" ");

// set s5 to the substring at one past the last space.
// there was no last space so this sets s5 to the substring starting at zero
string s5 = s1.Substring(ix + 1);

Console.WriteLine("s4: {0}\ns5: {1}\n", s4, s5);
Console.WriteLine("s1: {0}\n", s1);
```


يكون ناتج التنفيذ:



```
C:\WINDOWS\system32\cmd.exe
s2: Four
s3: Three
s4: Two
s5: One

s1: One

Press any key to continue . . .
```