

NGÔN NGỮ LẬP TRÌNH C# CƠ BẢN (tt)

21. Mảng

- Cú pháp khai báo mảng:

```
kiểu_dữ_liệu[] tên_mảng;
```

Trong đó:

- *kiểu_dữ_liệu*: kiểu của phần tử trong mảng
- *tên_mảng*: xác định tên mảng

Ví dụ:

```
double[] balance;
```

- Khởi tạo mảng:

```
kiểu_dữ_liệu[] tên_mảng = new kiểu_dữ_liệu[KÍCH_THƯỚC_MẢNG];
```

Ví dụ:

```
double[] balance = new double[10];
```

- Gán giá trị cho một mảng:

- Gán giá trị cho các phần tử mảng riêng biệt bởi sử dụng chỉ số mảng:

```
double[] balance = new double[10];  
balance[0] = 4500.0;
```

- Gán giá trị cho mảng khi khai báo mảng:

```
double[] balance = { 2340.0, 4523.69, 3421.0};
```

- Khai báo và khởi tạo một mảng:

```
int [] marks = new int[5] { 99, 98, 92, 97, 95};
```

Hoặc:

```
int [] marks = new int[] { 99, 98, 92, 97, 95};
```

- Truy cập các phần tử mảng:

Một phần tử được truy cập bởi chỉ mục mảng, phần tử đầu tiên của mảng có chỉ số là 0. Chỉ số của phần tử bên trong dấu ngoặc vuông ở sau tên mảng. Ví dụ:

```
double salary = balance[9];
```

Ví dụ:

```
using System;

namespace DenoLWin
{
    class Deno
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mang trong C#");
            Console.WriteLine("-----");
            int [] n = new int[10]; /* n la mot mang gom 10 so nguyen */
            int i,j;
            /* khoi tao cac phan tu cua mang n */
            for (i = 0; i < 10; i++)
            {
                n[i] = i + 100;
            }

            /* hien thi gia tri cac phan tu cua mang n */
```

```

    for (j = 0; j < 10; j++)
    {
        Console.WriteLine("Phan tu [{0}] = {1}", j, n[j]);
    }

    Console.ReadKey();
}
}
}

```

Sử dụng vòng lặp *foreach* trong C#

```

using System;

namespace DenoLWin
{
    class Deno
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Mang trong C#");
            Console.WriteLine("-----");
            int[] n = new int[10]; /* n la mot mang gom 10 so nguyen */

            /* khoi tao cac phan tu trong mang n */
            for (int i = 0; i < 10; i++)
            {
                n[i] = i + 100;
            }
        }
    }
}

```

```

    }

    /* hien thi cac gia tri cua phan tu trong mang n */
    foreach (int j in n)
    {
        int i = j - 100;

        Console.WriteLine("Phan tu [{0}] = {1}", i, j);

        i++;
    }

    Console.ReadKey();
}
}
}

```

22. Chuỗi

Trong C#, bạn có thể sử dụng chuỗi (string) như là mảng các ký tự.

Tuy nhiên, phổ biến hơn là để sử dụng từ khóa **string** (là một alias cho lớp **System.String**) để khai báo một biến chuỗi.

Tạo đối tượng String bằng cách sử dụng một trong các phương thức sau:

- Gán một hằng chuỗi cho một biến String
- Sử dụng một constructor của lớp String
- Sử dụng toán tử nối chuỗi (+)
- Gọi một phương thức mà trả về một chuỗi
- Gọi một phương thức định dạng để chuyển đổi một giá trị hoặc một đối tượng thành biểu diễn chuỗi của nó.

Ví dụ:

```
using System;
```

```
namespace DenoLWin
{
    class Deno
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Cac cach tao chuoi trong C#");
            Console.WriteLine("-----");

            //su dung phep gan hang chuoi va toan tu noi chuoi
            string fname, lname;
            fname = "Tran Thanh";
            lname = "Tuan";

            string fullname = fname + " " + lname;
            Console.WriteLine("Ho va ten: {0}", fullname);

            //su dung constructor cua lop string
            char[] letters = { 'H', 'e', 'l', 'l', 'o' };
            string greetings = new string(letters);
            Console.WriteLine("\nLoi chao bang tieng Anh: {0}", greetings);

            //tu cac phuong thuc ma tra ve mot chuoi
            string[] sarray = { "CKC", "xin", "chao", "cac", "ban" };
            string message = String.Join(" ", sarray);
```

```

        Console.WriteLine("\nThong diep: {0}", message);

        //dinh dang phuong thuc de chuyen doi mot gia tri
        DateTime waiting = new DateTime(2019, 2, 19, 12, 45, 55);
        string chat = String.Format("Thong diep duoc gui luc {0:t} ngay {0:D}",
waiting);

        Console.WriteLine("\nThong diep: {0}", chat);

        Console.ReadKey();
    }
}
}

```

Các thuộc tính của lớp String trong C#:

STT	Thuộc tính
1	Length Lấy số ký tự của đối tượng String hiện tại

Một số phương thức của lớp String trong C#:

STT	Phương thức
1	public static int Compare(string strA, string strB) So sánh hai đối tượng String cụ thể và trả về một số nguyên chỉ vị trí có liên quan của chúng trong thứ tự sắp xếp
2	public static int Compare(string strA, string strB, bool ignoreCase)

	So sánh hai đối tượng String cụ thể và trả về một số nguyên chỉ vị trí có liên quan của chúng trong thứ tự sắp xếp. Tuy nhiên, nó bỏ qua sự phân biệt hoa thường nếu tham số ignoreCase là true
3	public static String Concat(String str0, String str1) Nối chuỗi hai đối tượng String
4	public static String Concat(String str0, String str1, String str2) Nối chuỗi ba đối tượng String
5	public static String Concat(String str0, String str1, String str2, String str3) Nối chuỗi bốn đối tượng String
6	public boolean Contains(String value) Trả về một giá trị cho biết có hay không đối tượng value đã cho xuất hiện bên trong chuỗi hiện tại
7	public static String Copy(String str) Tạo một đối tượng str mới với cùng giá trị như chuỗi hiện tại
8	public void CopyTo(int sourceIndex, char[] destination, int destinationIndex, int count) Sao chép một số ký tự cụ thể từ một vị trí đã cho của đối tượng String tới một vị trí đã xác định trong một mảng các ký tự Unicode
9	public boolean EndsWith(String value) Xác định có hay không phần kết thúc của chuỗi hiện tại là đối tượng value
10	public boolean Equals(String value)

	Xác định có hay không đối tượng String hiện tại và đối tượng value đã cho là có cùng giá trị
11	public static bool Equals(string a, string b) Xác định có hay không hai đối tượng String đã cho có cùng giá trị
12	public static string Format(string format, Object arg0) Thay thế một hoặc nhiều mục định dạng trong một chuỗi đã cho với biểu diễn chuỗi của một đối tượng cụ thể
13	public int IndexOf(char value) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện đầu tiên của ký tự Unicode đã cho trong chuỗi hiện tại
14	public int IndexOf(string value) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện đầu tiên của chuỗi đã cho trong Instance (sự thể hiện) này
15	public int IndexOf(char value, int startIndex) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện đầu tiên của ký tự Unicode đã cho trong chuỗi này, bắt đầu tìm kiếm tại vị trí của ký tự đã cho
16	public int IndexOf(string value, int startIndex) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện đầu tiên của chuỗi đã cho trong Instance (sự thể hiện) này, bắt đầu tìm kiếm tại vị trí của ký tự đã cho
17	public int IndexOfAny(char[] anyOf) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện đầu tiên của Instance (sự thể hiện) này của bất kỳ ký tự nào trong một mảng ký tự Unicode đã xác định

18	public int IndexOfAny(char[] anyOf, int startIndex) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện đầu tiên của Instance (sự thể hiện) này của bất kỳ ký tự nào trong một mảng ký tự Unicode đã xác định, bắt đầu tìm kiếm tại vị trí của ký tự đã cho
19	public string Insert(int startIndex, string value) Trả về một chuỗi mới trong đó một chuỗi đã cho được chèn tại một vị trí có chỉ mục đã xác định trong đối tượng String hiện tại
20	public static bool IsNullOrEmpty(string value) Chỉ rằng có hay không chuỗi đã cho là null hoặc là một chuỗi Empty
21	public static string Join(string separator, params string[] value) Nối chuỗi tất cả phần tử của một mảng chuỗi, bởi sử dụng Separator (bộ tách) đã cho giữa mỗi phần tử
22	public static string Join(string separator, string[] value, int startIndex, int count) Nối chuỗi các phần tử đã xác định của một mảng chuỗi, bởi sử dụng Separator (bộ tách) đã cho giữa mỗi phần tử
23	public int LastIndexOf(char value) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện cuối cùng của ký tự Unicode đã cho bên trong đối tượng String hiện tại
24	public int LastIndexOf(string value) Trả về chỉ mục (dựa trên cơ sở 0) cho sự xuất hiện cuối cùng của một chuỗi đã cho bên trong đối tượng String hiện tại
25	public string Remove(int startIndex) Gỡ bỏ tất cả ký tự trong Instance hiện tại, bắt đầu tại vị trí đã xác định và tiếp tục tới vị trí cuối cùng, và trả về chuỗi đó

26	public string Remove(int startIndex, int count) Gỡ bỏ số ký tự đã cho trong chuỗi hiện tại bắt đầu tại một vị trí đã xác định và trả về chuỗi đó
27	public string Replace(char oldChar, char newChar) Thay thế tất cả ký tự Unicode đã cho xuất hiện trong đối tượng String hiện tại với ký tự Unicode đã xác định và trả về chuỗi mới
28	public string Replace(string oldValue, string newValue) Thay thế tất cả chuỗi đã cho xuất hiện trong đối tượng String hiện tại với đối tượng string đã xác định và trả về chuỗi mới
29	public string[] Split(params char[] separator) Trả về một mảng chuỗi mà chứa các chuỗi phụ trong đối tượng String hiện tại, được giới hạn bởi các phần tử của một mảng ký tự Unicode đã cho
30	public string[] Split(char[] separator, int count) Trả về một mảng chuỗi mà chứa các chuỗi phụ trong đối tượng String hiện tại, được giới hạn bởi các phần tử của một mảng ký tự Unicode đã cho. Tham số int xác định số chuỗi phụ tối đa để trả về
31	public bool StartsWith(string value) Xác định có hay không phần bắt đầu của instance của chuỗi này so khớp với chuỗi đã cho
32	public char[] ToCharArray() Trả về một mảng ký tự Unicode với tất cả ký tự trong đối tượng String hiện tại
33	public char[] ToCharArray(int startIndex, int length)

	Trả về một mảng ký tự Unicode với tất cả ký tự trong đối tượng String hiện tại, bắt đầu từ chỉ mục đã xác định và tới độ dài đã cho
34	public string ToLower() Trả về một bản sao của chuỗi này đã được biến đổi thành chữ thường
35	public string ToUpper() Trả về một bản sao của chuỗi này đã được biến đổi thành chữ hoa
36	public string Trim() Gỡ bỏ tất cả ký tự whitespace từ đối tượng String hiện tại

Ví dụ:

So sánh chuỗi:

```
using System;

namespace DenoLWin
{
    class Deno
    {
        static void Main(string[] args)
        {
            Console.WriteLine("So sanh chuoi trong C#");
            Console.WriteLine("-----");

            string str1 = "So sanh chuoi trong C#";
            string str2 = "So sanh chuoi trong Csharp";
```

```

        if (String.Compare(str1, str2) == 0)
        {
            Console.WriteLine(str1 + " va " + str2 + " la giong nhau.");
        }
        else
        {
            Console.WriteLine(str1 + " va " + str2 + " la khong giong nhau.");
        }

        Console.ReadKey();
    }
}

```

Kiểm tra chuỗi con:

```

using System;

namespace DenoLTWin
{
    class Deno
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Kiem tra chuoai con trong C#");
            Console.WriteLine("-----");
        }
    }
}

```

```

string str = "Chuoi con trong C#";
if (str.Contains("trong"))
{
    Console.WriteLine("Tim thay chuoi con 'trong'.");
}

Console.ReadKey();
}
}
}

```

Lấy chuỗi con:

```

using System;

namespace DenoLTWin
{
    class Deno
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Lay chuoi con trong C#");
            Console.WriteLine("-----");

            string str = "Lay chuoi con trong C#";
            Console.WriteLine("Chuoi ban dau: " +str);
        }
    }
}

```

```

        string substr = str.Substring(10);

        Console.WriteLine("Chuoi con: " +substr);


        Console.ReadKey();
    }
}
}

```

Nối chuỗi:

```

using System;

namespace DenoLWin
{
    class Deno
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Noi chuoi trong C#");
            Console.WriteLine("-----");

            string[] starray = new string[]{"Hoc C# co ban.",
            "Chuong nay trinh bay ve chuoi trong C#.",
            "Chung ta dang tim hieu ve noi chuoi trong C#.",
            "Xin chuc cac ban hoc tot.",
            "Xin tran trong cam on!"};

```

```
string str = String.Join("\n", starray);

Console.WriteLine(str);

Console.ReadKey();

}

}

}
```

BÀI TẬP

1. Viết chương trình nhập một mảng số nguyên có N phần tử, in các phần tử mảng này trên màn hình.
 2. Viết chương trình nhập một mảng số nguyên có N phần tử, in các phần tử mảng theo chiều đảo ngược.
 3. Viết chương trình nhập một mảng số nguyên có N phần tử, tính tổng các phần tử của mảng và hiển thị kết quả trên màn hình.
 4. Viết chương trình nhập một mảng số nguyên có N phần tử, sao chép các phần tử của mảng này sang mảng khác.
 5. Viết chương trình nhập một mảng số nguyên có N phần tử, tìm số phần tử giống nhau trong mảng đó và in kết quả trên màn hình.
 6. Viết chương trình nhập một mảng số nguyên có N phần tử, in các phần tử là duy nhất trong mảng trên màn hình hiển thị (các phần tử là duy nhất nếu không có phần tử khác giống nó).
 7. Viết chương trình nhập hai mảng số nguyên lần lượt có N, M phần tử, trộn (ghép) hai mảng trên và sau đó sắp xếp các phần tử của mảng trộn này theo thứ tự tăng dần.
 8. Viết chương trình nhập một mảng số nguyên có N phần tử, đếm số lần xuất hiện của từng phần tử trong mảng và in kết quả trên màn hình
- Ví dụ:

Phần tử 3 xuất hiện 1 lần

Phần tử 4 xuất hiện 2 lần

...

9. Viết chương trình nhập một mảng số nguyên có N phần tử, tìm phần tử lớn nhất và phần tử nhỏ nhất và in kết quả trên màn hình.
10. Viết chương trình nhập một mảng số nguyên có N phần tử, chia mảng này thành hai mảng: một mảng chứa số chẵn và một mảng chứa số lẻ rồi in kết quả trên màn hình.
11. Viết chương trình nhập một mảng số nguyên có N phần tử, sắp xếp mảng theo thứ tự tăng dần và in mảng đã sắp xếp trên màn hình.
12. Viết chương trình nhập một mảng số nguyên có N phần tử, sắp xếp mảng theo thứ tự giảm dần và in mảng đã sắp xếp trên màn hình.
13. Viết chương trình nhập một mảng số nguyên có N phần tử, chèn thêm phần tử X tại vị trí K vào mảng rồi in mảng trên màn hình. (X, K là 2 số nguyên được nhập từ bàn phím)
14. Viết chương trình nhập một mảng số nguyên có N phần tử, sau đó xóa phần tử tại vị trí K (nhập từ bàn phím) rồi in mảng kết quả trên màn hình.
15. Viết chương trình nhập một chuỗi từ bàn phím, rồi sau đó in chuỗi trên màn hình.
16. Viết chương trình tìm độ dài của một chuỗi mà không sử dụng hàm trong thư viện.
17. Viết chương trình đếm số từ có trong một chuỗi. Ví dụ:

Nhập chuỗi: CKC xin chao cac ban

Chuỗi vừa nhập có 5 từ

18. Viết chương trình so sánh hai chuỗi để xem hai chuỗi này có bằng nhau không. Ví dụ:

Chuỗi 1: Bai tap C# co giai

Chuỗi 2: Bai tap C# co giai

--> Hai chuỗi là bằng nhau

19. Viết chương trình đếm số chữ cái, số chữ số, số ký tự đặc biệt trong một chuỗi. Ví dụ:

Nhập chuỗi: Tuyen tap 150 bai tap C# co giai!

Chuỗi này có: 21 chữ cái, 3 chữ số, 9 ký tự đặc biệt

20. Viết chương trình đếm số phụ âm, số nguyên âm có trong một chuỗi bất kỳ. Ví dụ:

Nhập chuỗi: Thanh Tuan

Chuỗi trên có 3 nguyên âm, 6 phụ âm

21. Viết chương trình lấy chuỗi con, với vị trí bắt đầu K và độ dài L của chuỗi con từ chuỗi S được nhập từ bàn phím. (K, L là 2 số nguyên được nhập từ bàn phím)
22. Viết chương trình nhập một chuỗi, sau đó sắp xếp mảng các ký tự trong chuỗi theo thứ tự tăng dần rồi in kết quả trên màn hình
23. Viết chương trình nhập một chuỗi, sau đó chuyển chữ hoa thành chữ thường và chữ thường thành chữ hoa rồi in chuỗi kết quả trên màn hình.
24. Viết chương trình nhập hai chuỗi: một chuỗi ban đầu S1 và một chuỗi con S2 cần đếm, sau đó đếm số lần xuất hiện của chuỗi S2 trong chuỗi S1.