Stanford
nâng tầm tri thức

www.stanford.com.vn

“HỌC ĐỂ LÀM VIỆC”

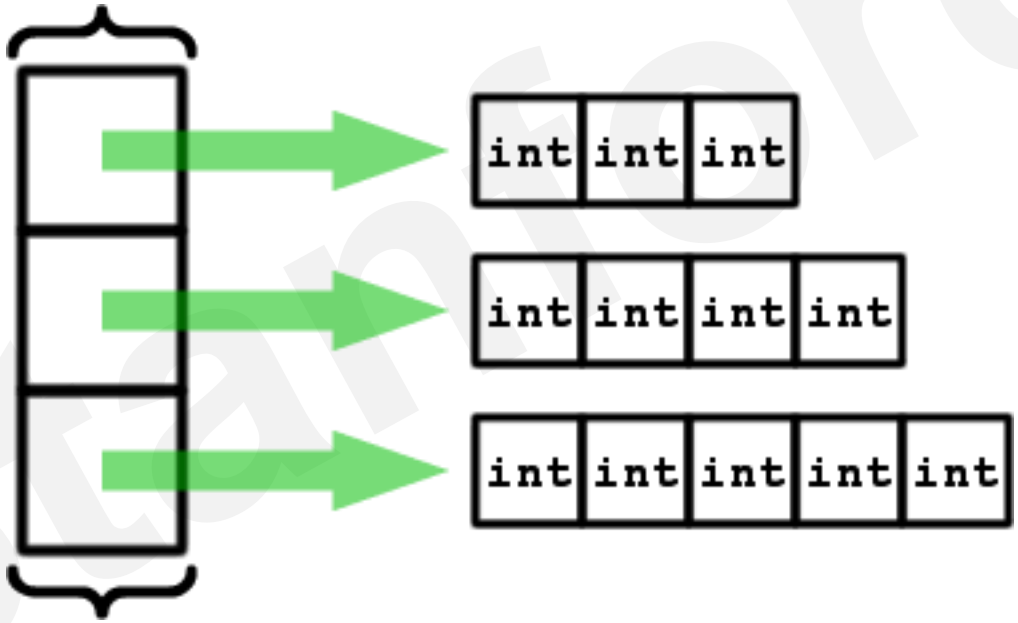
C#.NET for Base

Stanford – Dạy kinh nghiệm lập trình

Contents

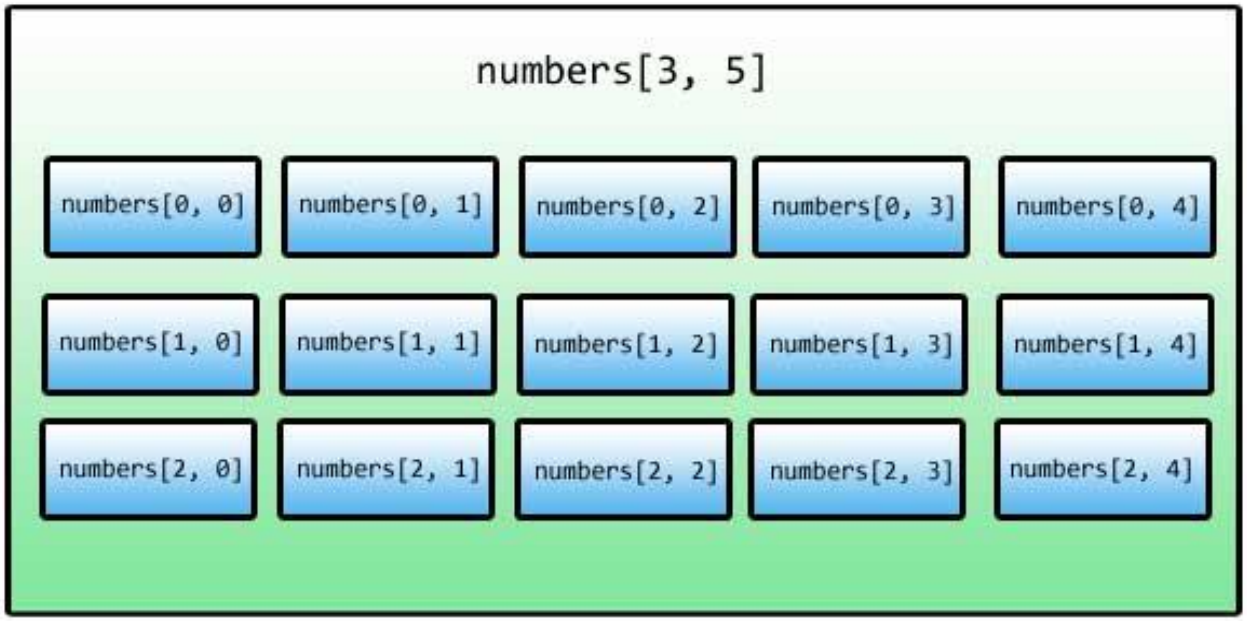
Array, List and Collection

- **Array, List and Collection**
  - **Mảng (array)**: là tập hợp các phần tử có cùng kiểu dữ liệu



- **Array, List and Collection**

Ví dụ: Mảng 2 chiều gồm 3 hàng và 5 cột



- **Array, List and Collection**
  - Mảng 1 chiều
  - Mảng nhiều chiều
  - Mảng Jagged
  - Đối tượng Array
  - Đối tượng ArrayList
  - List

- ## Array, List and Collection
  - ### Method

Method	Description
Clear	Thiết lập các thành phần của mảng về 0 hay null
Copy	Sao chép một vùng của mảng vào mảng khác
CreateInstance	Tạo một thể hiện mới cho mảng
IndexOf	Trả về chỉ mục đầu tiên chứa giá trị trong mảng một chiều.
LastIndexOf	Trả về chỉ mục cuối cùng chứa giá trị trong mảng một chiều
Reverse	Đảo thứ tự các thành phần trong mảng một chiều

- **Array, List and Collection**
  - **Method**

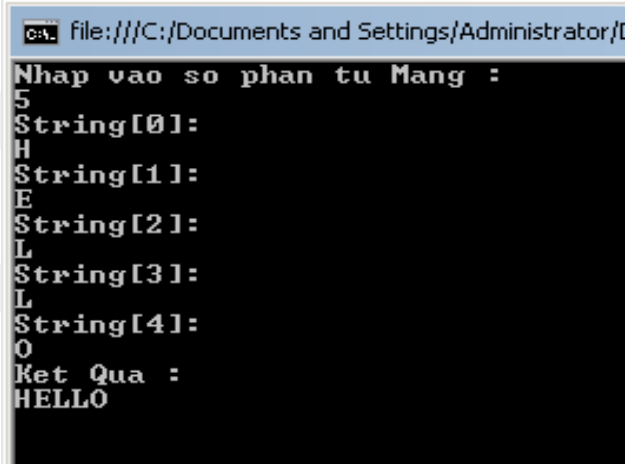
Method	Description
GetLength	Phương thức trả về kích thước của một chiều cố định trong mảng.
SetValue	Thiết lập giá trị cho một thành phần xác định trong mảng

- **Array, List and Collection**
  - **Properties**

Properties	Description
IsFixedSize	Thuộc tính trả về giá trị bool thể hiện mảng có kích thước cố định hay không
IsReadOnly	Thuộc tính trả về giá trị bool thể hiện mảng chỉ đọc hay không
Length	Thuộc tính trả về chiều dài của mảng

- **Array, List and Collection**
  - **Mảng 1 chiều:**
    - Mảng chứa nhiều phần tử có cùng kiểu dữ liệu.
    - Sử dụng không gian tên: System.Array
    - Cú pháp khai báo mảng:
    - **<Kiểu dữ liệu>[] <Tên mảng> = new [KieuDuLieu][SoPhanTu];**
    - SetValue(value,index) : Gán giá trị cho từng phần tử tương ứng
    - GetValue(index): Nhận về giá trị của từng phần tử

- ## Array, List and Collection
  - **Mảng 1 chiều:**
    - **Ví dụ 1**: Khai báo Mảng gồm n phần tử có kiểu String do người sử dụng nhập vào. Sau đó yêu cầu người sử dụng nhập giá trị cho n phần tử đó → in chúng ra ngoài màn hình



- ## Array, List and Collection
  - ### Mảng 1 chiều:
    - **Ví dụ 2**: Khai báo Mảng là kiểu số nguyên

```
//Khai báo mảng 1 chiều
int[] songuyen = new int[3] { 1, 3, 5 };

//Có thể gán theo cách trên khi khởi tạo một mảng
//Hoặc gán giá trị như sau
songuyen[0] = 5;
songuyen[1] = 6;
songuyen[2] = 8;

//Lấy thông tin từ mảng
int i = 0;
for (i = 0; i < songuyen.Length; i++)
{
    lbKetQua.Items.Add(songuyen[i]);
}
```

- **Array, List and Collection**

✓ **Mảng nhiều chiều:**

➢ **Mảng 2 chiều:**

- **Cú pháp:**
  - <Kiểu dữ liệu>[,] <Tên Mảng> = new <Kiểu dữ liệu>[m,n];
  - m: Số dòng được tính theo hàng ngang của Mảng.
  - n: Số cột được tính theo hàng dọc của Mảng.
  - Ví dụ: Khai báo mảng 2 chiều:
  - int[,] iArray = new int[2,3];

- **Array, List and Collection**

✓ **Mảng nhiều chiều:**

➢ **Mảng 2 chiều:**

- **Ví dụ 1:** Khai báo Mảng 2 chiều, đưa giá trị vào mảng rồi in chúng ra ngoài màn hình.

```
//Khai báo mảng 2 chiều
string[,] arr = new string[2, 3];

//Gán giá trị
arr[0, 0] = "Stanford";
arr[0, 1] = "Dạy kinh nghiệm lập trình";
arr[0, 2] = "Xây dựng và phát triển phần mềm";

arr[1, 0] = "Website: www.stanford.com.vn";
arr[1, 1] = "C#, ASP.NET, Windows Mobile";
arr[1, 2] = "eTraining, eTesting, HocLapTrinh App";
```

- **Array, List and Collection**

✓ **Mảng nhiều chiều:**

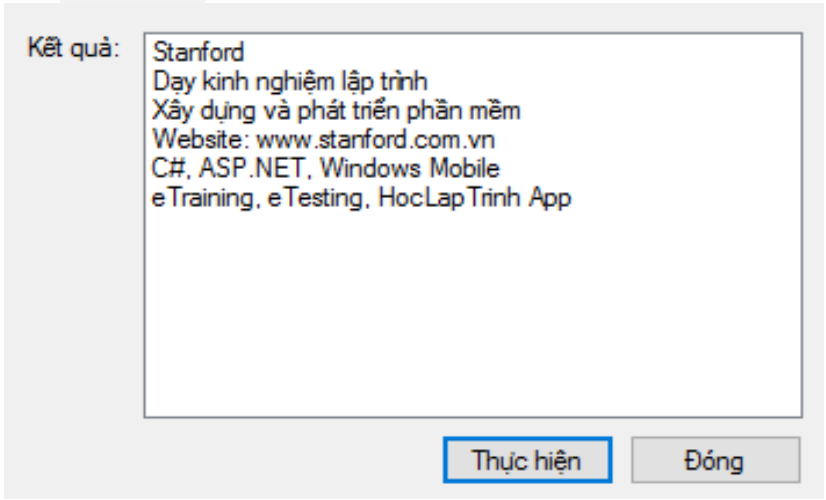
➢ **Mảng 2 chiều:**

- **Ví dụ 1:** Hiển thị các phần tử của mảng lên giao diện

```
//Hiển thị thông tin mảng lên giao diện
int j = 0;

lbKetQua.Items.Clear();

for (i = 0; i < 2; i++)
{
    for (j = 0; j < 3; j++)
    {
        lbKetQua.Items.Add(arr[i, j]);
    }
}
```



- **Array, List and Collection**

✓ **Mảng nhiều chiều:**

➢ **Mảng n chiều:**

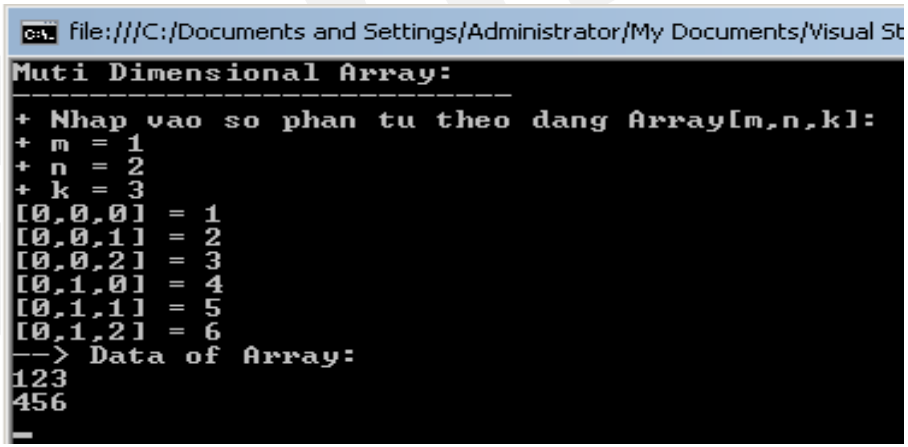
- **Tương tự như mảng 2 chiều. Cú pháp:**
- <Kiểu dữ liệu>[,,] <tên mảng> = new <kiểu dữ liệu>[,,];
- Ví dụ khai báo mảng nhiều chiều :
- String[, ,] strArray = new string[2,3,4];

- **Array, List and Collection**

✓ **Mảng nhiều chiều:**

➢ **Mảng n chiều:**

- **Ví dụ 1:** Khai báo Mảng 3 chiều, duyệt qua từng phần tử và yêu cầu người sử dụng nhập dữ liệu vào rồi in chúng ra ngoài màn hình.



Array, List and Collection

✓ Mảng nhiều chiều:

➢ Mảng Jagged:

- C# cung cấp cú pháp cho phép bạn khai báo mảng nhiều chiều, mà từng phần tử có thể có kiểu dữ liệu là một mảng gọi là mảng Jagged.
- Mỗi phần tử trong mảng Jagged có thể là mảng có kích thước khác nhau, chính vì vậy mảng Jagged còn được gọi là mảng của mảng (Array of Array).
- Cú pháp : **<Kiểu dữ liệu>[] [] obj = new <kiểu dữ liệu>[row] [];**
- Mảng Jagged cho phép chúng ta sử dụng bộ nhớ một cách tiết kiệm hơn.
- Cú pháp: **Int[] [] jaggedArray = new int[3][];**

- **Array, List and Collection**

✓ **Mảng nhiều chiều:**

➢ **Mảng Jagged:**

- **Ví dụ 1:** Khai báo Mảng Jagged gồm 3 phần tử và gán giá trị cho từng phần tử rồi in chúng ra màn hình.

```
//Khai báo mảng Jagged
int[][] arrJagged = new int[3][];

//Gán giá trị
arrJagged[0] = new int[] { 3, 4 };
arrJagged[1] = new int[4] { 4, 5, 6, 8 };
arrJagged[2] = new int[5] { 8, 9, 12, 14, 15 };
```

- **Array, List and Collection**

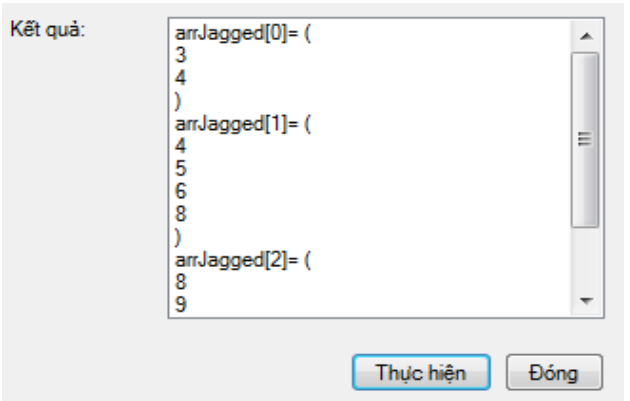
✓ **Mảng nhiều chiều:**

➢ **Mảng Jagged:**

- **Ví dụ 1:** Hiển thị các phần tử của mảng Jagged ra giao diện

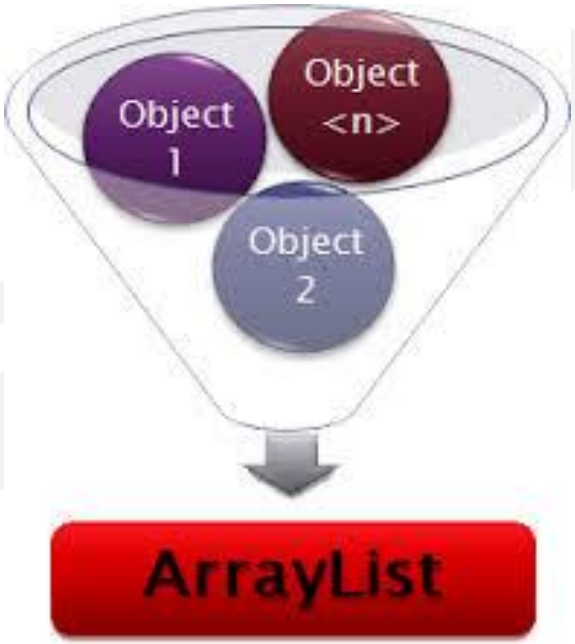
```
//Hiển thị thông tin mảng
for (i = 0; i < 3; i++)
{
    lbKetQua.Items.Add(string.Format("arrJagged[{0}]= (", i));
    for (j = 0; j < arrJagged[i].Length; j++)
    {
        lbKetQua.Items.Add(arrJagged[i][j]);
    }

    lbKetQua.Items.Add(")");
}
```



- **Array, List and Collection**

✓ **List:** là một mảng động bao gồm một tập hợp các phần tử có cùng kiểu dữ liệu.



Array, List and Collection

List:

- Cú pháp : **List<type> name = new List<type>();**
- Sử dụng namespace: **System.Collections**
  - **Add()** : Thêm phần tử vào Lists
  - **Remove()** : Xóa phần tử từ Lists
  - **Insert()** : Chèn phần tử vào List, Có thể chỉ định ví trí thêm vào. Có 2 tham số (index,value).
  - **Sort()** : Sắp xếp giá trị trong list.
  - **Contains()** : Tìm giá trị trong list, return true or false.

- **Array, List and Collection**

✓ **List:**

- **Ví dụ:**

```
List<string> lstGiatri = new List<string>();
Console.WriteLine("So phan tu toi thieu:" + lstGiatri.Capacity);

//Gán giá trị
lstGiatri.Add("C# for Base");
lstGiatri.Add(".NET framework");
lstGiatri.Add("Stanford.com.vn - Day kinh nghiem lap trinh");
lstGiatri.Insert(3,"Tang 5, so 20 ngo 100 Nguyen Chi Thanh");

//Hiển thị giá trị
foreach (string a in lstGiatri)
{
    Console.WriteLine(a);
}
//Số phần tử tối thiểu
Console.WriteLine("So phan tu toi thieu:" + lstGiatri.Capacity);
//Số phần tử
Console.WriteLine(lstGiatri.Count);
```

- **Array, List and Collection**

✓ **ArrayList:**

- **Ví dụ:**

```
//Khai báo arraylist
ArrayList arrList = new ArrayList();

//Gán giá trị cho mảng arraylist
arrList.Add("Stanford");
arrList.Add(20);
arrList.Add(DateTime.Now);


//Gỡ bỏ phần tử có giá trị là 20
arrList.Remove(arrList[1]);

for (int i = 0; i < arrList.Count; i++)
{
    lbKetQua.Items.Add(arrList[i]);
}
```

String

String

- **System.String** là lớp được thiết kế để lưu trữ chuỗi.
- **Các phương thức:**

Phương thức	Mục đích
Compare	so sánh nội dung của 2 chuỗi
Format	định dạng một chuỗi chứa 1 giá trị khác và chỉ định cách mỗi giá trị nên được định dạng.
IndexOf	vị trí xuất hiện đầu tiên của 1 chuỗi con hoặc kí tự trong chuỗi
IndexOfAny	vị trí xuất hiện đầu tiên của bất kì 1 hoặc 1 tập kí tự trong chuỗi

- **String**
  - **Các phương thức:**

Phương thức	Mục đích
LastIndexOf	Lấy vị trí cuối của một kí tự trong một chuỗi
Substring(start, length)	Lấy một chuỗi con từ một chuỗi cha với start là vị trí bắt đầu, length là độ dài chuỗi cần lấy
Split	Sử dụng để chuyển từ một chuỗi sang một mảng bằng cách chia chuỗi ra theo một định dạng nào đó
Replace(str_old, str_new)	Thay thế một kí tự, chuỗi bằng một kí tự, chuỗi mới
Trim	Cắt khoảng trắng đầu và cuối dòng của một chuỗi
Length	Lấy độ lớn của một chuỗi
ToLower, ToUpper	Chuyển chuỗi về dạng chữ thường, chữ in hoa

- **String**
  - **Ví dụ:** Cho một chuỗi thông tin như sau, thực hiện lấy từ Stanford trong chuỗi đã cho.

```
string input = "Stanford - Dạy kinh nghiệm lập trình\r\n";
input += "Địa chỉ: Số 20 ngõ 678 Đường Láng, Đống Đa, Hà Nội";
input += "Hotline: 0936.172.315-0962.723.236    Website: www.stanford.com.vn";
string strStanford = input.Substring(0, 8); //Lấy từ Stanford
```

- **String**
  - **StringBuilder**

giúp thao tác nhanh các chuỗi với ít tổn hao bộ nhớ hơn so với lớp **String**.

- **Ví dụ:**

Cho một chuỗi "**Cong ty Stanford – Dao tao va phat trien cong nghe**" thay thế các ký tự trong chuoi này bằng ký tự đứng sau nó 2 vị trí.

=> Nếu dùng lệnh for và chuỗi để xử lý yêu cầu này sẽ tốn bộ nhớ vì mỗi lần thay đổi cần thêm một chuỗi mới trong khi đó StringBulder chỉ tạo ra một vùng để lưu trữ chuỗi ký tự này.

“HỌC ĐỂ LÀM VIỆC”

Thank You !