

## Chương 4

# Array

## Nội dung

1. Mảng
2. Câu lệnh lặp foreach
3. Mảng đa chiều
4. Câu hỏi

## 1. Mảng

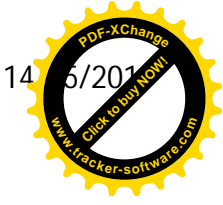
- Khai báo mảng
- Giá trị mặc định
- Truy cập các thành phần trong mảng
- Khởi tạo thành phần trong mảng
- Sử dụng từ khóa params
- Phương thức và thuộc tính cơ bản

## Mảng là gì?

- Mảng là một tập hợp có thứ tự của những **đối tượng**
- Tất cả các **đối tượng** này cùng một kiểu
- Ngôn ngữ C# cung cấp cú pháp chuẩn cho việc khai báo những **đối tượng** mảng



Tuy nhiên, cái thật sự được tạo ra là **đối tượng** của kiểu `System.Array`



### Mô tả

Name of array (c)	c[ 0 ]	-45
	c[ 1 ]	6
	c[ 2 ]	0
	c[ 3 ]	72
	c[ 4 ]	1543
	c[ 5 ]	-89
	c[ 6 ]	0
	c[ 7 ]	62
	c[ 8 ]	-3
	c[ 9 ]	1
Index (or subscript) of the element in array c	c[ 10 ]	6453
	c[ 11 ]	78

### Sử dụng một mảng

- Khai báo
- Tạo thể hiện (xác định số phần tử)
- Gán giá trị (giá trị mặc định; Giá trị cụ thể)

1. Khai báo  
 2. Tạo thể hiện + Gán giá trị mặc định  
 3. Gán giá trị cụ thể

Khai báo + Gán giá trị cụ thể

Khai báo + Tạo thể hiện + Gán giá trị cụ thể

### Sử dụng một mảng

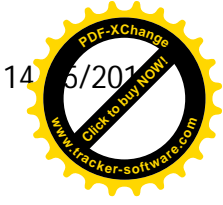
```
int[] myIntArray;
myIntArray = new int[6];
// Chương trình
```

1. Khai báo  
 2. Tạo thể hiện + Gán giá trị mặc định  
 3. Gán giá trị cụ thể

### Sử dụng một mảng

```
int[] myIntArray = { 2, 4, 6, 8, 10};
```

Khai báo + Gán giá trị cụ thể

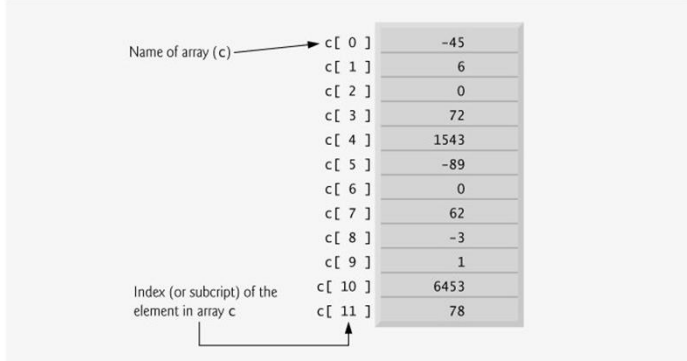


### Sử dụng một mảng

```
int[] myIntArray = new int[5] { 2, 4, 6, 8, 10};
```

Khai báo + Tạo  
thể hiện + Gán giá  
trị cụ thể

### Truy cập các phần tử



```
namespace mang_1
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] mangso1;
            mangso1 = new int[5] { 2, 5, 7, 9, 1 };
            for (int i = 1; i < 6; i++)
            {
                System.Console.WriteLine(mangso1[i]);
            }
            System.Console.ReadLine();
        }
    }
}
```

### Tìm kiếm phần tử

```
namespace mang_timpt
{
    class Program
    {
        static void Main(string[] args)
        {
            int gt, vitri, sotim;
            sotim = 5;
            vitri = 0;
            int[] mang = new int[5] { 3, 5, 6, 5, 8 };
            bool kq = false;
            for (int i = 0; i < 5; i++)
            {
                gt = mang[i];
                if (gt==sotim)
                {
                    kq = true;
                    vitri = i; }
            }
            if (kq==true)
            { System.Console.WriteLine("Co phan tu " + sotim + " trong mang");
              System.Console.WriteLine("Vi tri " + sotim+ " trong mang la "+vitri); }
            else
            { System.Console.WriteLine("Khong co phan tu " + sotim + " trong mang"); }
            System.Console.ReadLine();
        }
    }
}
```

## Ví dụ

Tạo mảng chứa điểm của sinh viên với các yêu cầu:

- Số phần tử do người nhập quyết định
- Điểm ứng với mỗi phần tử do người nhập
- Sắp xếp mảng theo điểm từ nhỏ đến lớn
- Xuất ra kết quả

1 4 5 7 8 9 8  
1 4 5 7 8 9 8

```
for (int i = 0; i <= sopt - 2; i++)
{
    for (int j = i+1; j <= sopt-1; j++)
    {
        if (mangsv[i]>mangsv[j])
        {
            int tg = mangsv[j];
            mangsv[j] = mangsv[i];
            mangsv[i] = tg;
        }
    }
}
```

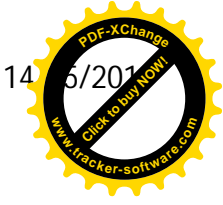
## Sử dụng từ khóa params

1. Tạo một phương thức :
  - Để lấy một số các biến của tham số nguyên
  - Sử dụng từ khóa **params** để truyền một số biến của tham số mà không cần thiết phải tạo một mảng
  - Sử dụng vòng lặp **foreach** truy xuất các phần tử
2. Tại Main() :
  - Định nghĩa một đối tượng
  - Truyền giá trị “của mảng” cho phương thức

## Ví dụ

```
namespace mang_params
{
    class Program
    {
        public void DisplayVals(params int[] intVals)
        {
            foreach (int i in intVals)
            {
                Console.WriteLine("Hiển thị {0}", i);
            }
            Console.ReadLine();
        }

        static void Main(string[] args)
        {
            Program m = new Program();
            m.DisplayVals(6, 7, 8, 9);
            int[] mang2=new int[5]{3, 4, 6, 2, 8};
            m.DisplayVals(mang2);
        }
    }
}
```



## Phương thức và thuộc tính

Thành viên	Mô tả
BinarySearch()	Phương thức tính public tìm kiếm một mảng một chiều đã sắp thứ tự.
Clear()	Phương thức tính public thiết lập các thành phần của mảng về 0 hay null.
Copy()	Phương thức tính public đã nạp chồng thực hiện sao chép một vùng của mảng vào mảng khác.
CreateInstance()	Phương thức tính public đã nạp chồng tạo một thể hiện mới cho mảng
IndexOf()	Phương thức tính public trả về chỉ mục của thể hiện đầu tiên chứa giá trị trong mảng một chiều
LastIndexOf()	Phương thức tính public trả về chỉ mục của thể hiện cuối cùng của giá trị trong mảng một chiều
Reverse()	Phương thức tính public đảo thứ tự của các thành phần trong mảng một chiều
Sort()	Phương thức tính public sắp xếp giá trị trong mảng một chiều.
IsFixedSize	Thuộc tính public giá trị bool thể hiện mảng có kích thước cố định hay không.
IsReadOnly	Thuộc tính public giá trị bool thể hiện mảng chỉ đọc hay không

Array.Clear(mang, n, m);  
 Array.Copy(mang, mang1, n);  
 Array.IndexOf(mang, "12");  
 Array.LastIndexOf(mang, "12");  
 Array.Sort(mang);  
 Array.Reverse(mang);

## Phương thức và thuộc tính

IsSynchronized	Thuộc tính public giá trị bool thể hiện mảng có hỗ trợ thread-safe
Length	Thuộc tính public chiều dài của mảng
Rank	Thuộc tính public chứa số chiều của mảng
SyncRoot	Thuộc tính public chứa đối tượng dùng để đồng bộ truy cập trong mảng
GetEnumerator()	Phương thức public trả về IEnumerator
GetLength()	Phương thức public trả về kích thước của một chiều cố định trong mảng
GetLowerBound()	Phương thức public trả về cận dưới của chiều xác định trong mảng
GetUpperBound()	Phương thức public trả về cận trên của chiều xác định trong mảng
Initialize()	Khởi tạo tất cả giá trị trong mảng kiểu giá trị bằng cách gọi bộ khởi dựng mặc định của từng giá trị.
SetValue()	Phương thức public thiết lập giá trị cho một thành phần xác định trong mảng.

mang.Length;  
 mang.Rank;

## Một số phương thức

- <tên mảng>.<Sum()/Max()/Min()/Average(>>;
- <tên mảng>.Clear();
- <tên mảng>.Count();
- <tên mảng>.First();
- <tên mảng>.Last();

```
namespace mangnc
{
    public class Tester
    {
        public static void PrintArray(object[] mArray)
        {
            foreach( object obj in mArray)
            {
                Console.WriteLine("Hien thi "+ obj);
            }
            Console.WriteLine("\n");
        }
        static void Main()
        {
            string[] m1Array = {"Ban", "la", "Nam", "Nu"};
            PrintArray( m1Array );
            Array.Reverse( m1Array );
            PrintArray( m1Array );
            string[] m2Array = {"Chung", "toi", "la", "nhung", "nguoi", "lap", "trinh", "may", "tin"};
            PrintArray( m2Array );
            Array.Sort( m2Array );
            PrintArray( m2Array );
            Console.ReadLine();
        }
    }
}
```

## 2. Câu lệnh lặp foreach

- Câu lệnh lặp foreach cho phép chúng ta lặp qua tất cả các mục trong một mảng hay trong một tập hợp

```
foreach (<kiểu dữ liệu thành phần> <tên truy cập> in <mảng/tập hợp>)
{
    // thực hiện thông qua <tên truy cập> tương ứng với
    // từng mục trong mảng hay tập hợp
}
```

```
int[] mang = new int[3] {3, 5, 7};
foreach (int i in mang)
{
    System.Console.WriteLine("Xuat phan tu mang {0}", i.ToString(), '\t');
}
```

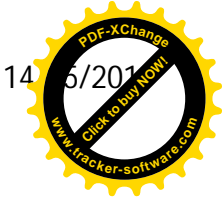
i.ToString() : Chuyển số qua chuỗi  
'\t' : tab ngang

## 3. Mảng đa chiều

- **Mảng một chiều** trong đó các thành phần của nó chỉ đơn giản là các đối tượng kiểu giá trị hay đối tượng tham chiếu
- **Mảng hai chiều** được tổ chức thành các **dòng** và **cột** :
  - Các dòng được tính theo hàng ngang của mảng
  - Các cột được tính theo hàng dọc của mảng
- **Mảng đa chiều**:
  - Là mảng có tổ chức rất phức tạp
  - Mỗi thành phần là một mảng khác

## C# hỗ trợ hai kiểu

- **Mảng đa chiều cùng kích thước**:
  - Trong mảng này mỗi dòng trong mảng có cùng kích thước với nhau
  - Mảng này có thể là hai hay nhiều hơn hai chiều
- **Mảng đa chiều không cùng kích thước**:
  - Trong mảng này các dòng không cùng kích thước với nhau



## Mảng đa chiều cùng kích thước

- Còn gọi là mảng hình chữ nhật:
  - Chiều đầu tiên được tính bằng số **đòng** của mảng
  - Chiều thứ hai được tính bằng số **cột** của mảng
- Khai báo :  
`<kiểu dữ liệu>[,] <tên mảng>;`

## Tạo mảng

```
const int rows = 4;  
const int columns = 3;  
int [,] mangso1 = new int[rows, columns];
```

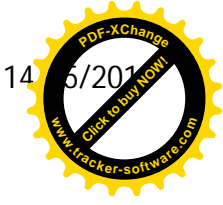
```
int[] mangso2 = new int[rows];  
int[] mangso3 = new int[columns];
```

## khởi tạo các thành phần trong mảng

```
for(int i = 0; i < rows; i++)  
{  
    for(int j = 0; j < columns; j++)  
    {  
        mangso[i,j] = i+j;  
    }  
}
```

## xuất nội dung ra màn hình

```
for(int i = 0; i < rows; i++)  
{  
    for(int j = 0; j < columns; j++)  
    {  
        Console.WriteLine("mangso[{0},{1}] = {2}",i, j, mangso[i, j]);  
    }  
}
```



## Mảng đa chiều có kích khác nhau

- Hình dạng không phải hình chữ nhật vì các chiều của chúng không đều nhau
- Khai báo :
  - Khai báo số dòng trong mảng trước
  - Mỗi dòng sẽ giữ một mảng, có kích thước bất kỳ
  - Những mảng này được khai báo riêng
  - Sau đó chúng ta khởi tạo giá trị các thành phần trong những mảng bên trong

## Khai báo

```
const int rows = 4; // khai báo mảng tối đa bốn dòng
int[][] mArray = new int[rows][];
mArray[0] = new int[5]; // dòng đầu tiên có 5 phần tử
mArray[1] = new int[2]; // dòng thứ hai có 2 phần tử
mArray[2] = new int[3]; // dòng thứ ba có 3 phần tử
mArray[3] = new int[5]; // dòng cuối cùng có 5 phần tử
```

0					
1					
2					
3					

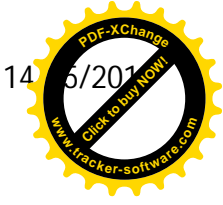
## Khởi tạo một vài giá trị

```
mArray[0][3] = 15;
mArray[1][1] = 12;
mArray[2][1] = 9;
mArray[2][2] = 99;
mArray[3][0] = 10;
mArray[3][1] = 11;
mArray[3][2] = 12;
mArray[3][3] = 13;
mArray[3][4] = 14;
```

0				15	
1		12			
2		9	99		
3	10	11	12	13	14

## Xuất kết quả

```
for(int i = 0; i < 5; i++)
{
    Console.WriteLine("mArray[0][{0}] = {1}", i, m[0][i]);
}
for(int i = 0; i < 2; i++)
{
    Console.WriteLine("mArray[1][{0}] = {1}", i, mArray[1][i]);
}
```



#### 4. Câu hỏi

- Điều phân biệt giữa mảng và các thành phần bên trong một mảng?
- Từ khoá params được sử dụng làm gì?
- Ý nghĩa của lệnh lặp foreach? Lệnh này được sử dụng với kiểu dữ liệu nào?
- Hãy cho biết sự khác nhau giữa hai cách gọi Arr[i][j] và Arr[i, j]?
- Làm thế nào để biết kích thước của một mảng?