

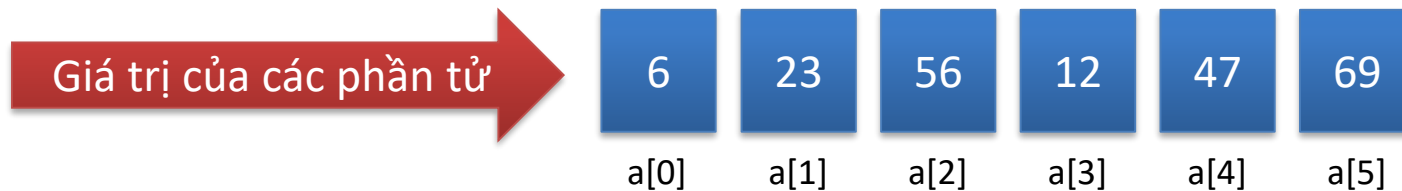


# Bài 05

## Mảng và Chỉ mục

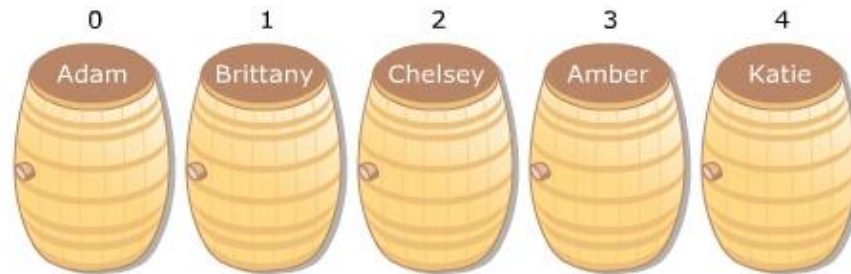
- Mảng một chiều
- Mảng hai chiều
- Lớp **Array**
- Chỉ mục

- Mảng là 1 kiểu dữ liệu tham chiếu
- Là tập các biến có cùng tên, cùng kiểu dữ liệu nhưng khác nhau về chỉ số
- Chỉ số (index) dùng để xác định vị trí phần tử trong mảng, chỉ số bắt đầu từ 0



- Trong đó a là tên mảng
- 0-5: là chỉ số

- **Mảng 1 chiều:** bao gồm các phần tử được lưu trữ trong 1 hàng trên bộ nhớ



- **Khai báo mảng**

`Kiểu_dữ_liệu[] tên_mảng;`

- **Khai báo và khởi tạo kích thước mảng**

```
Kiểu_dữ_liệu[] tên_mảng = new  
Kiểu_dữ_liệu[số_phần_tử];
```

## ■ Khai báo và khởi tạo số phần tử

Kiểu\_dữ\_liệu[] tên\_mảng=new Kiểu\_dữ\_liệu[]{gt1,gt2,...};

hoặc

Kiểu\_dữ\_liệu[] tên\_mảng={gt1,gt2,...};

### Ví dụ

```
//khai báo mảng
int[] a;
//khai báo và khởi tạo số phần tử
int[] b = new int[5];
//khai báo và khởi tạo giá trị
int[] c = new int[] { 4, 5, 6, 7, 8, 1 };
string[] names={"dung","thang","hai","thuy","long","son"};
```

- Khi các phần tử của mảng chưa được gán giá trị thì giá trị mặc định sẽ như sau:

Kiểu dữ liệu	Giá trị mặc định
int	0
float	0.0
double	0.0
char	'\0'
string	null

- **Truy cập tới 1 phần tử**

`Tên_mảng[chỉ_số];`

- **Lấy số phần tử mảng**

`Tên_mảng.Length`

- **Duyệt mảng**

`for(i=0;i<tên_mang.Length;i++)`

`{`

`//thao tác trên các phần tử của mảng`

`}`

## ▪ Dùng foreach duyệt mảng

```
foreach(kiểu_dữ_liệu tên_biến in tên_mảng)
{
    //thao tác trên tên_biến
}
```

### Ví dụ

```
int[] x = { 4, 5, 7, 9, 2, 12, 41 };
//duyet mảng bằng for
for (int i = 0; i < x.Length; i++)
{
    Console.WriteLine(x[i]);
}
//duyet mảng bằng foreach
foreach (int m in x)
{
    Console.WriteLine(m);
}
```



- Là mảng mà các phần tử được lưu trữ và xác định dựa vào hai chỉ số (dòng, cột) giống như sheet trong Microsoft Excel
- Trong C# mảng 2 chiều chia làm 2 loại
  - Mảng hình chữ nhật (Rectangle Array): số phần tử trên các hàng là bằng nhau
  - Mảng zic zắc(Jagged array): số phần tử trên hàng là khác nhau

## ■ Khai báo

– Kiểu\_dữ\_liệu[,] tên\_mảng = new  
kiểu\_dữ\_liệu[dòng,cột];

### Ví dụ

```
//ví dụ mảng rectangular
int[,] numbers ={
    {3,4,1,7},
    {6,7,8,6},
    {6,2,9,7}
};

//duyet mảng để hiển thị
for (int i = 0; i < numbers.GetLength(0); i++)
{
    for (int j = 0; j < numbers.GetLength(1); j++)
    {
        Console.Write(numbers[i, j] + "\t");
    }
    Console.WriteLine();
}
```

## ■ Khai báo

Kiểu\_dữ\_liệu[][] tên\_mảng=new kiểu\_dữ\_liệu[dòng][];

### Ví dụ

```
//ví dụ về mảng jagged
string[][] companies = new string[3][];

companies[0] = new string[] { "Intel", "AMD", "Atom" };
companies[1] = new string[] { "IBM", "Microsoft", "Sun", "Facebook" };
companies[2] = new string[] { "HP", "Canon", "Lexmark", "Epson", "Samsung" };

//duyet mảng
for (int i = 0; i < companies.GetLength(0); i++)
{
    Console.WriteLine("Nhóm công ty {0}", i + 1);
    for (int j = 0; j < companies[i].GetLength(0); j++)
    {
        Console.Write(companies[i][j] + "\t");
    }
    Console.WriteLine();
}
```

- Lớp **Array** được xây cung cấp sẵn trong namespace System
  - Là lớp cơ sở cho tất cả các mảng trong C#.
  - Lớp Array cung cấp các phương thức khác nhau để thực hiện các công việc như:
    - Tạo
    - Tìm kiếm
    - Copy
    - Sắp xếp mảng

- Sắp xếp mảng

`Array.Sort(tên_mảng);`

- Đảo ngược mảng

`Array.Reverse(tên_mảng);`

- Tìm kiếm

`Array.IndexOf(tên_mảng, giá_trị_tìm);`//tìm phần tử ở đầu

`Array.LastIndexOf(tên_mảng, giá_trị_tìm);`//tìm phần tử ở cuối

- Thay đổi kích thước mảng

`Array.Resize<kiểu_dl>(ref tên_mảng,số_pt_mới);`

## Ví dụ

```
//một số thao tác với lớp Array
int[] arr = new int[] { 4, 5, 6, 7, 8, 1 };
//sắp xếp mảng
Array.Sort(arr);
//đảo ngược mảng
Array.Reverse(arr);
//Tìm kiếm
int value = 5;
int pos = Array.IndexOf(arr, value);
if (pos < 0)
    Console.WriteLine("Không tìm thấy");
else
    Console.WriteLine("Đã tìm thấy " + value + " tại vị trí " + pos);
//thay đổi kích thước
Array.Resize<int>(ref arr, 7); // tăng thêm một phần tử trong mảng
arr[6] = 12;
```

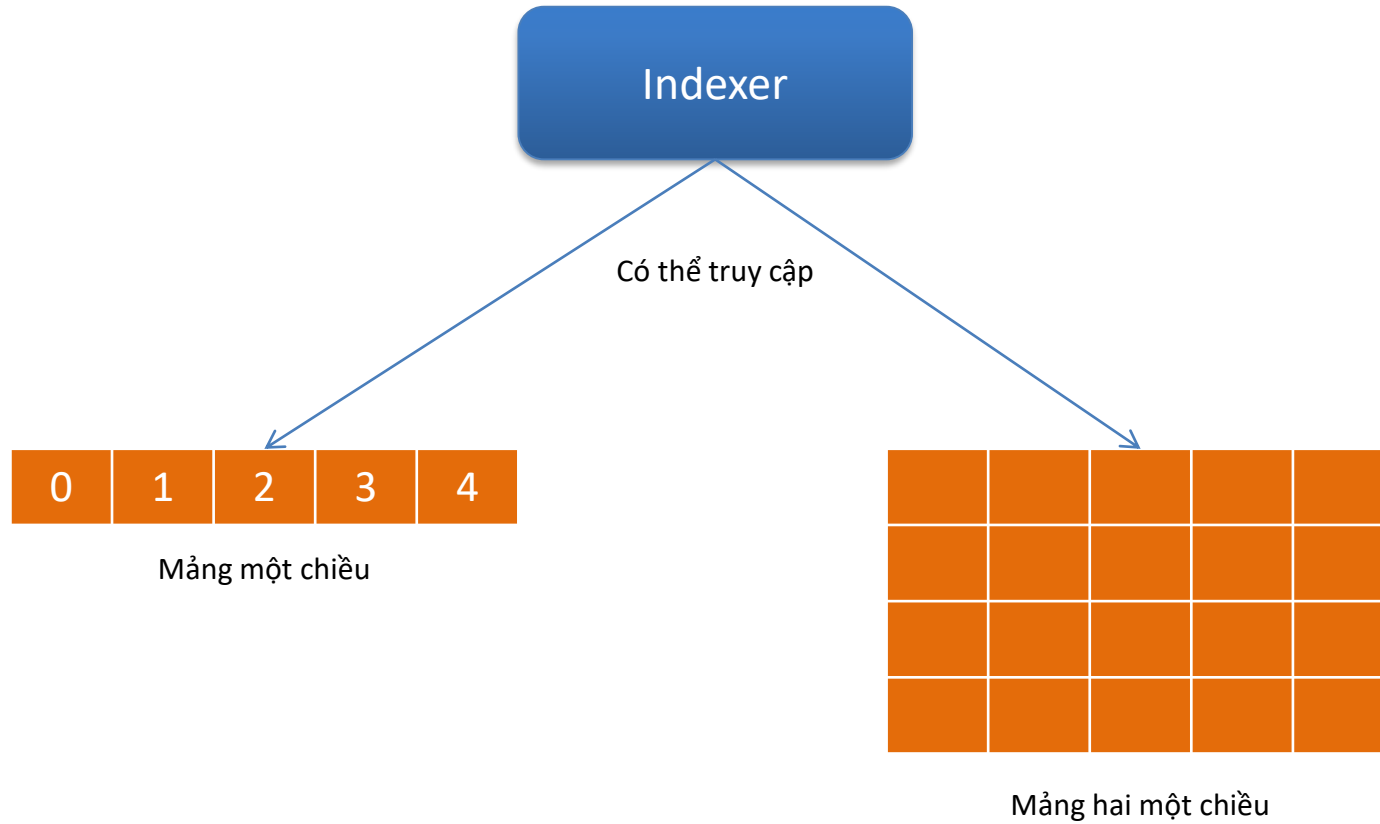
- Chỉ mục là các thành viên dữ liệu cho phép truy cập dữ liệu trong các đối tượng theo một cách tương tự như truy cập mảng.
- Chỉ mục cung cấp truy cập nhanh hơn vào các dữ liệu trong một đối tượng.
- Việc thực hiện các chỉ mục cũng tương tự như các thuộc tính, ngoại trừ việc khai báo một chỉ mục chứa các tham số.
- Trong C #, chỉ mục cũng được gọi là mảng thông minh.

# Chỉ mục (indexer)

## Ví dụ

```
//ví dụ về chỉ mục
class Student
{
    string[] names;
    public Student(int n)
    {
        names = new string[n];
    }
    public string this[int i]
    {
        get { return names[i]; }
        set { names[i] = value; }
    }
    static void Main(string[] args)
    {
        Student st = new Student(3);
        st[0] = "Nam";
        st[1] = "Long";
        st[2] = "Thủy";
    }
}
```





# Đa chỉ mục(More indexer) 1-2

## Ví dụ

```
//ví dụ về chỉ mục
class Student
{
    string[] names;
    double[,] marks;
    public Student(int n,int m)
    {
        names = new string[n];
        marks = new double[n, m];
    }
    public string this[int i]
    {
        get { return names[i]; }
        set { names[i] = value; }
    }
    public double this[int i, int j]
    {
        get { return marks[i, j]; }
        set { marks[i, j] = value; }
    }
    static void Main(string[] args)
    {
        Student st = new Student(3,2);
        st[0] = "Nam";
        st[0, 0] = 9;
        st[0, 1] = 6.5;
        st[1] = "Long";
        st[1, 0] = 5;
        st[1, 1] = 7.5;
        st[2] = "Thủy";
        st[2, 0] = 10;
        st[2, 1] = 9.5;
    }
}
```

# Hỏi Đáp

