



# Mảng một chiều

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

- Chúng ta có thể khai báo mảng một chiều trong C# với cú pháp theo sau:

```
<kiểu dữ liệu>[] <tên mảng>;
```

Ví dụ ta có khai báo như sau:

```
int [] a;
```

```
float [] b;
```

.....

Cặp dấu ngoặc vuông ([]) báo cho trình biên dịch biết rằng chúng ta đang khai báo một mảng. Kiểu dữ liệu là kiểu của các thành phần chứa bên trong mảng. Trong ví dụ bên trên **a** được khai báo là mảng số nguyên, **b** là mảng các số thực.

Chúng ta tạo thể hiện của mảng bằng cách sử dụng từ khóa **new** như sau:

```
a = new int[6];
```

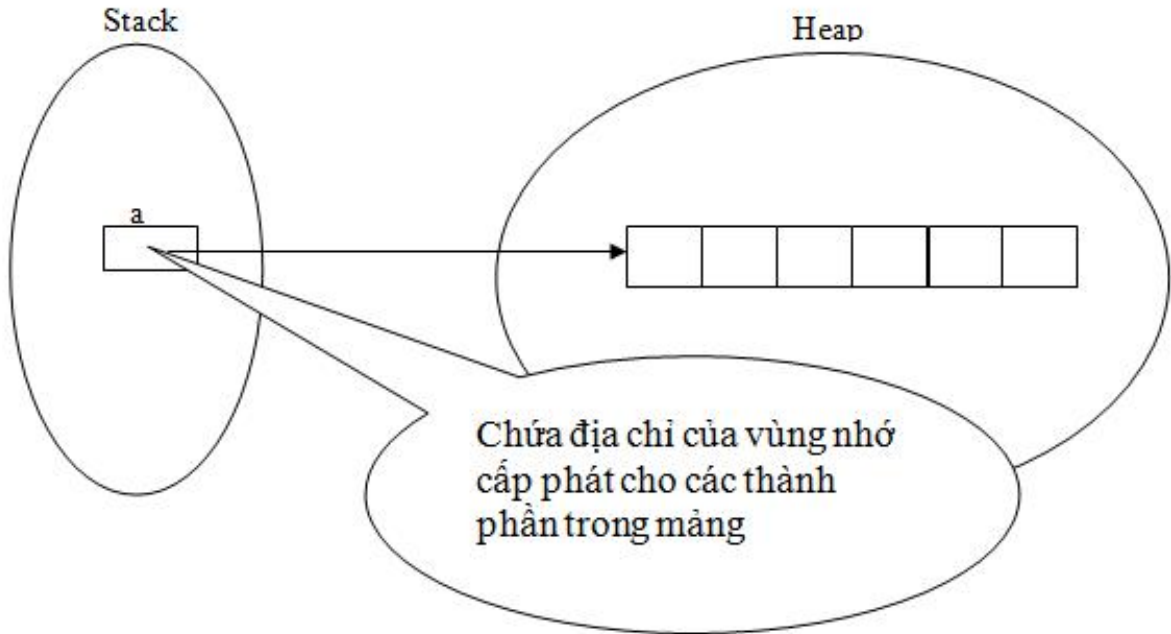
```
b=new double[20];
```

Khai báo này máy sẽ thiết lập bên trong bộ nhớ cho mảng a một dãy sáu vùng nhớ liên tiếp, mỗi vùng nhớ chứa được một số nguyên kiểu int tức là mỗi vùng nhớ gồm 4 byte liên tiếp, vậy tổng máy cấp phát cho mảng a là 24 byte liên tiếp . Mảng b máy sẽ cấp phát một dãy 20 vùng nhớ liên tiếp, mỗi vùng nhớ chứa được một số thực kiểu double tức là mỗi vùng nhớ gồm 8 byte liên tiếp, vậy tổng máy cấp phát cho mảng b là 120 byte liên tiếp .

Ở đây chúng ta cần phân biệt giữa hai thành phần bản thân mảng và các thành phần trong mảng. Như chúng ta đã biết trong C# mảng là một kiểu dữ liệu tham chiếu nên bản thân mảng trong ví dụ trên là **a** và các thành phần trong mảng là sáu số nguyên, thế thì bản thân mảng **a** được cấp phát trong stack còn các thành phần của mảng được cấp

## Mảng một chiều

phát trong heap, bản thân mảng **a** chỉ chứa địa chỉ của vùng nhớ cấp cho các thành phần của mảng trong heap mà thôi. Ta có thể minh họa như sau:



Ta có thể vừa khai báo mảng và vừa cấp phát bộ nhớ cho các phần tử của mảng như sau:

```
<kiểu dữ liệu>[] <tên mảng>=new <kiểu dữ liệu>[kích thước];
```

Ví dụ:

```
int []a=new int[40];
```

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

.....

Ta có thể khai báo mảng và đồng thời khởi tạo cho các phần tử của mảng như sau:

Chúng ta có thể khởi tạo nội dung của một mảng ngay lúc tạo thể hiện của mảng bằng cách đặt những giá trị bên trong dấu ngoặc (**{}**). C# cung cấp hai cú pháp để khởi tạo các thành phần của mảng, một cú pháp dài và một cú pháp ngắn:

Mảng một chiều

```
int[] myIntArray1 = new int[5] { 2, 4, 6, 8, 10};
```

```
int[] myIntArray2 = { 2, 4, 6, 8, 10};
```

Không có sự khác biệt giữa hai cú pháp trên, và hầu hết các chương trình đều sử dụng cú pháp ngắn hơn do sự tự nhiên và lười đánh nhiều lệnh của người lập trình.

Khi khai báo mảng nếu ta không khởi đầu giá trị cho chúng thì các phần tử của mảng tự động được khởi tạo một giá trị ngầm định theo bảng thống kê sau:

Kiểu dữ liệu	Giá trị mặc định
int, long, byte,...	0
bool	false
char	'\0' (null)
enum	0
reference	null

Ví dụ: Khi ta khai báo mảng

```
int []a=new int[5];
```

Ta sẽ thu được mảng a gồm 5 phần tử và mỗi phần tử của mảng được khởi đầu giá trị là 0

Mảng trong ngôn ngữ C# có một vài sự khác biệt so với mảng trong ngôn ngữ C++ và một số ngôn ngữ khác, bởi vì chúng là những đối tượng. Điều này sẽ cung cấp cho mảng sử dụng các phương thức và những thuộc tính.

Ngôn ngữ C# cung cấp cú pháp chuẩn cho việc khai báo những đối tượng Array. Tuy nhiên, cái thật sự được tạo ra là đối tượng của kiểu System.Array. Mảng trong ngôn ngữ C# kết hợp cú pháp khai báo mảng theo kiểu ngôn ngữ C và kết hợp với định nghĩa lớp do đó thể hiện của mảng có thể truy cập những phương thức và thuộc tính của System.Array.

Một số các thuộc tính và phương thức của lớp System.Array

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

## Mảng một chiều

Truy nhập vào các phần tử của mảng: Để truy nhập vào các phần tử của mảng chúng ta truy nhập thông qua tên mảng và chỉ số tương ứng. Đặc biệt trong C# phần tử đầu tiên của mảng có chỉ số là 0. Phần tử cuối cùng của mảng có chỉ số là Tên\_Mảng.Length-1. Nhớ rằng chỉ số của mảng phải ở trong giới hạn cho phép. Ta có thể truy nhập vào các phần tử của mảng thông qua công thức tổng quát sau: Tên\_Mảng[chỉ số]

Ví dụ:

```
int []=new int[5];
```

```
a[0] // Truy nhập vào phần tử đầu tiên của mảng
```

```
a[i] // Truy nhập vào phần tử thứ i của mảng  $i \in [0, \text{Tên\_Mảng.Length}-1]$ 
```

Một số ví dụ áp dụng

**Ví dụ 1:** Nhập vào một dãy số nguyên sau đó thực hiện các yêu cầu sau:

- Tính tổng các phần tử của mảng chia hết cho 3
- Tìm giá trị lớn nhất trong số các phần tử của mảng
- Nhập vào từ bàn phím một số nguyên x và cho biết số nguyên này xuất hiện trong mảng bao nhiêu lần
- Rút gọn mảng(nghĩa là mỗi phần tử chỉ đượ xuất hiện một lần)

```
-----  
using System;  
  
class VD  
{  
  
    static int[] a;  
  
    static int n;  
  
    static void Nhap()  
{  
  
        int i;
```

Mảng một chiều

```
Console.WriteLine("Nhap so phan tu cua mang n=");

n = int.Parse(Console.ReadLine());

a = new int[n];

Console.WriteLine("Hay nhap cac phan tu cho mang");

for (i = 0; i < n; ++i)

{

Console.WriteLine("a[{0}]=", i);

a[i] = int.Parse(Console.ReadLine());

}

}

static void Hien(int []x)

{

int i;

for (i = 0; i < x.Length ; ++i)

Console.WriteLine("{0}\t", x[i]);

Console.WriteLine();

}

static int Tong3()

{

int i,t=0;

for(i=0;i<n;++i)

if(a[i]%3==0) t=t+a[i];
```

Mảng một chiều

```
return t;
```

```
}
```

```
static int Max()
```

```
{
```

```
int i,mx;
```

```
mx = a[0];
```

```
for (i = 1; i < n; ++i)
```

```
if (mx < a[i]) mx = a[i];
```

```
return mx;
```

```
}
```

```
static int Dem(int x)
```

```
{
```

```
int i,d=0;
```

```
for (i = 0; i < n; ++i)
```

```
if (a[i] == x) d++;
```

```
return d;
```

```
}
```

```
static void RutGon(out int[] kq)
```

```
{
```

```
int i,d=0,j;
```

```
int[] tmp = new int[n];
```

```
bool ok;
```

Mảng một chiều

```
for (i = 0; i < n; ++i)
{
    ok = true;
    for(j=0;j<d;++j)
    if (tmp[j] == a[i])
    { ok = false; break; }
    if (ok) tmp[d++] = a[i];
}

kq = new int[d];
Array.Copy(tmp, kq, d);
}

static void Main()
{
    ConsoleKeyInfo kt;

    int x;

    int []b=null;

    do{

        Console.Clear();

        Console.WriteLine("\t\t\tMain Menu");

        Console.WriteLine("\t1. Nhap mang");

        Console.WriteLine("\t2. Hien Mang");

        Console.WriteLine("\t3. Cac phan tu cua mang chi het cho 3");
```



Mảng một chiều

```
Console.WriteLine("\t4. Gia tri lon nhat cua mang");  
Console.WriteLine("\t5. So lan xuat hien cua phan tu x trong mang");  
Console.WriteLine("\t6. Rut gon mang");  
Console.WriteLine("\t7. Thoat khoi chuong trinh");  
Console.Write(" Ban hay chon mot cong viec tu 1->7:");  
kt=Console.ReadKey();  
Console.WriteLine();  
switch(kt.KeyChar)  
{  
case '1':  
Nhap();  
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");  
Console.ReadKey();  
break;  
case '2':  
Console.WriteLine("Cac phan tu cua mang la:");  
Hien(a);  
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");  
Console.ReadKey();  
break;  
case '3':  
Console.WriteLine(" Ket qua la {0:8}",Tong3());
```

Mảng một chiều

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '4':
```

```
Console.WriteLine(" Ket qua la {0,8}",Max());
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '5':
```

```
Console.Write("Nhap x=");x=int.Parse(Console.ReadLine());
```

```
Console.WriteLine(" Ket qua la {0:8}",Dem(x));
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '6':
```

```
Console.WriteLine("Cac phan tu cua mang sau khi rut gon:");
```

```
RutGon(out b);
```

```
Hien(b);
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '7':
```

Mảng một chiều

```
Environment.Exit(0); break;  
  
}  
  
} while (true);  
  
}  
  
}
```

---

**Ví dụ 2:** Cho hai dãy số  $a_1, a_2, \dots, a_n$  và  $b_1, b_2, \dots, b_m$  có các phần tử là các số nguyên được nhập vào từ bàn phím. Sau thực hiện các yêu cầu sau:

1. Xây dựng mảng  $c$  bằng cách ghép hai mảng  $a$  và  $b$  với nhau
2. Kiểm tra mảng  $c$  có lập thành một cấp số cộng không
3. Đưa ra các phần tử của mảng  $c$  xuất hiện đúng một lần
4. Tách mảng  $c$  thành hai mảng: một mảng chứa toán số chẵn, một mảng chứa toàn số lẻ
5. Sắp xếp mảng  $a, b$  theo thứ tự tăng dần. Sau đó xây dựng mảng  $d$  bằng cách chen các phần tử của mảng  $a, b$  sao cho ta thu được mảng  $d$  cũng có thứ tự tăng dần

---

```
using System;  
  
class VD  
  
{  
  
static void Nhap(char ten, out int []x)  
  
{  
  
int i,n;  
  
Console.WriteLine("Nhap thông tin cho các phần tử của mảng {0}",ten);  
  
Console.Write("Nhap số phần tử của mảng:");  
  
n = int.Parse(Console.ReadLine());
```

Mảng một chiều

```
x = new int[n];
```

```
Console.WriteLine("Hay nhap cac phan tu cho mang");
```

```
for (i = 0; i < n; ++i)
```

```
{
```

```
Console.Write("{0}[" + i + "]=" + ten, i);
```

```
x[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
}
```

```
static void Hien(int[] x)
```

```
{
```

```
int i;
```

```
for (i = 0; i < x.Length; ++i)
```

```
Console.Write("{0}\t", x[i]);
```

```
Console.WriteLine();
```

```
}
```

```
static void GhepMang(int[] x, int[] y, out int[] kq)
```

```
{
```

```
kq = new int[x.Length + y.Length];
```

```
Array.Copy(x, kq, x.Length);
```

```
Array.Copy(y, 0, kq, x.Length, y.Length);
```

```
}
```

```
static bool CapSoCong(int[] x)
```

Mảng một chiều

```
{  
bool ok = true;  
int i;  
for (i = 1; i < x.Length - 1; ++i)  
if (x[i] != (x[i - 1] + x[i + 1]) / 2)  
{ ok = false; break; }  
return ok;  
}  
static void MotLan(int[] x, out int[] kq)  
{  
int i,j,d=0;  
int []tmp=new int[x.Length];  
bool ok;  
for (i = 0; i < x.Length; ++i)  
{  
ok = true;  
for(j=0;j<x.Length;++j)  
if (x[i] == x[j] && i != j)  
{ ok = false; break; };  
if(ok) tmp[d++]=x[i];  
}  
kq = new int[d];
```

Mảng một chiều

```
Array.Copy(tmp, kq, d);  
}  
  
static void Tach(int[] x, out int[] chan, out int[] le)  
{  
    int i,d1=0,d2=0;  
  
    int[] tmp1 = new int[x.Length];  
    int[] tmp2 = new int[x.Length];  
  
    for (i = 0; i < x.Length; ++i)  
        if (x[i] % 2 == 0)  
            tmp1[d1++] = x[i];  
        else  
            tmp2[d2++] = x[i];  
  
    chan = new int[d1]; Array.Copy(tmp1, chan, d1);  
    le = new int[d2]; Array.Copy(tmp2, le, d2);  
}  
  
static void Chen(int[] x, int[] y, out int[] kq)  
{  
    int i,n,k,j;  
  
    Array.Sort(x);  
    Array.Sort(y);  
  
    kq = new int[x.Length + y.Length];  
  
    Array.Copy(x, kq, x.Length);
```

Mảng một chiều

```
n=x.Length-1;
```

```
for (i = 0; i < y.Length; ++i)
```

```
{
```

```
if (kq[n] < y[i]) kq[++n] = y[i];
```

```
else
```

```
{
```

```
j = 0;
```

```
while (y[i] >= kq[j]) j++;
```

```
for (k = ++n; k > j; --k)
```

```
kq[k] = kq[k - 1];
```

```
kq[j] = y[i];
```

```
}
```

```
}
```

```
}
```

```
static void Main()
```

```
{
```

```
int []a=null;
```

```
int []b=null;
```

```
int []c=null;
```

```
int []d=null;
```

```
int[] kq = null;
```

```
int[] chan = null;
```

Mảng một chiều

```
int[] le = null;
```

```
ConsoleKeyInfo kt;
```

```
do
```

```
{
```

```
Console.Clear();
```

```
Console.WriteLine("\t\t\tMain Menu");
```

```
Console.WriteLine("\t1. Nhập thông tin cho hai mảng");
```

```
Console.WriteLine("\t2. Mảng ghép lại");
```

```
Console.WriteLine("\t3. Kiểm tra dãy có là cấp số công không?");
```

```
Console.WriteLine("\t4. Các phần tử của mảng xuất hiện đúng một lần");
```

```
Console.WriteLine("\t5. Tách mảng(chẵn, lẻ)");
```

```
Console.WriteLine("\t6. Chèn mảng");
```

```
Console.WriteLine("\t7. Thoát khỏi chương trình");
```

```
Console.Write(" Bạn hãy chọn một công việc từ 1->7:");
```

```
kt = Console.ReadKey();
```

```
Console.WriteLine();
```

```
switch (kt.KeyChar)
```

```
{
```

```
case '1':
```

```
Nhap('A', out a);
```

```
Nhap('B', out b);
```

```
Console.WriteLine("Bạn hãy nhấn phím bất kỳ để tiếp tục...");
```



Mảng một chiều

```
Console.ReadKey();
```

```
break;
```

```
case '2':
```

```
Console.WriteLine("Mang ghep la");
```

```
GhepMang(a, b, out c);
```

```
Hien(c);
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '3':
```

```
if(CapSoCong(c)==true )
```

```
Console.WriteLine(" Day da cho la cap so cong");
```

```
else
```

```
Console.WriteLine(" Day da cho kong phai la cap so cong");
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '4':
```

```
Console.WriteLine("Cac phan tu cua mang xaut hien dung mot lan");
```

```
MotLan(c,out kq);
```

```
Hien(kq);
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

Mảng một chiều

```
Console.ReadKey();
```

```
break;
```

```
case '5':
```

```
Tach(c, out chan, out le);
```

```
Console.WriteLine("Cac phan tu chan");
```

```
Hien(chan);
```

```
Console.WriteLine("Cac phan tu le");
```

```
Hien(le);
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '6':
```

```
Console.WriteLine("Mang chen la:");
```

```
Chen(a, b,out d);
```

```
Hien(d);
```

```
Console.WriteLine("Ban hay nhan phim bat ky de tiep tục...");
```

```
Console.ReadKey();
```

```
break;
```

```
case '7':
```

```
Environment.Exit(0); break;
```

```
}
```

```
} while (true);
```

Mảng một chiều

}

}

---

## Câu lệnh lặp foreach

Câu lệnh lặp foreach khá mới với những người đã học ngôn ngữ C, từ khóa này được sử dụng trong ngôn ngữ Visual Basic. Câu lệnh foreach cho phép chúng ta lặp qua tất cả các mục trong một mảng hay trong một tập hợp.

Cú pháp sử dụng lệnh lặp foreach như sau:

```
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
```

```
}
```

Ví dụ: Nhập vào một dãy số nguyên sau đó đưa ra màn hình các số chẵn trên một dòng, các số lẻ trên một dòng

---

```
using System;
```

```
class ViDu
```

```
{
```

```
static int[] a;
```

```
static void Nhap()
```

```
{
```

```
int i=0,x;
```

```
ConsoleKeyInfo kt;
```

Mảng một chiều

```
Console.WriteLine("Ban hay nhap vao mot day so nguyen");

do

{

Console.Write("So nguyen thu {0}:", i);

x = int.Parse(Console.ReadLine());

Array.Resize(ref a, ++i);

a[i - 1] = x;

Console.Write("Ban co nhap tiep C/K"); kt = Console.ReadKey();

Console.WriteLine();

} while (kt.KeyChar == 'c' || kt.KeyChar == 'C');

}

static void Hien()

{

Console.WriteLine("Cac phan tu chan la:");

foreach (int i in a)

if(i%2==0)

Console.Write("{0}\t", i);

Console.WriteLine("\nCac phan tu le la:");

foreach (int i in a)

if(i%2!=0)

Console.Write("{0}\t", i);

}
```

Mảng một chiều

```
static void Main()
```

```
{
```

```
Nhap();
```

```
Hien();
```

```
Console.ReadKey();
```

```
}
```

```
}
```

---