



FPT POLYTECHNIC



Bài 4: ARRAY VÀ ARRAYLIST

www.poly.edu.vn

hoclaptrinhweb.com

■ Tìm hi u v Collection trong VB.Net:

- Khái ni m Collection
- Phân lo i các Collection
- L p Collection trong VB.Net
- L p CollectionBase trong VB.Net

Mục tiêu bài học hôm nay

- Tìm hiểu về cách sử dụng mảng thông thường trong VB.Net
- Tìm hiểu về lớp ArrayList và cách sử dụng trong VB.Net
- So sánh mảng thông thường và ArrayList
- Áp dụng ví dụ có thời gian thực hiện

■ Cấu trúc dữ liệu mảng:

- Là cấu trúc lưu trữ tập hợp các phần tử cùng kiểu dữ liệu để truy cập theo chỉ số (index)
- Chỉ số của mảng bắt đầu từ 0.
- Kích thước của mảng được xác định khi khai báo mảng
- Trong VB.Net, mảng là một từ viết tắt của **Array**

Khai báo và kh i t _ o m _ n g

■ Có hai cách:

- Cách 1: Khai báo và kh i t _ o tách riêng

//khai báo bi n

Dim Tênm ng([s ph n t]) As Ki u

//t o m ng và tr bi n tham chi u n m ng

Tênm ng = New Ki u() {giá tr ph n t 1, giá tr ph n t 2,...}

- Cách 2: Khai báo và kh i t _ o ng th i

Dim Tênm ng([s ph n t]) As Ki u = {giá tr ph n t 1, giá tr
ph n t 2,...}

Khai báo và khởi tạo mảng

■ Ví dụ : khai báo mảng nguyên

● Cách 1:

```
Dim numbers() As Integer  
numbers = New Integer() {10, 20, 30, 40, 50}
```

● Cách 2:

```
Dim numbers() As Integer = {10, 20, 30, 40,  
50}
```

Thi t l p giá tr và truy c p các ph n t c a m ng

■ Thi t l p b ng ph ng th c SetValue

- Ví d : `numbers.SetValue(1, "222")`

-> `numbers = {10, 222, 30, 40, 50}`

■ Truy c p ph n t c a m ng:

- Thông qua ch s `index`

- Ví d : `num = numbers(1)` -> giá tr c a num là 222

- Thông qua ph ng th c `GetValue(index)`

- Ví d : `num2 = numbers.GetValue(2)` -> giá tr c a num2 là 30

- **Length()**: trả về số phần tử trong tất cả các chiều của Array
- **GetLength()**: trả về số phần tử của một chiều cụ thể của mảng
- **Rank()**: trả về số chiều kích thước của Array
- **GetType()**: trả về kiểu phần tử hiển thị trong Array
- **IsArray(a)**: trả về **True** nếu **a** là mảng tĩnh, ngược lại trả về **False**

- **Dùng l i nh ReDim:** cho phép thay i l i kích th c c a Array và giá tr c a các ph n t u c thi t l p l i là giá tr m c nh.

- Ví d :

```
Dim grades() As Integer = {87, 76, 99, 65, 89}
```

```
ReDim grades(9)
```

-> M ng grades lúc này có kích th c là 10 ph n t , và giá tr các ph n t là 0.

- Dùng I nh Preserve: N u mu n gi nguyên giá tr các ph n t trong m ng, s d ng I nh Preserve i kèm I nh ReDim.

- Ví d :

ReDim Preserve grades(9)

- > M ng grades lúc này có kích th c là 10 ph n t , và giá tr 5 ph n t ban u v n gi nguyên.

- Trong VB.Net, mảng có kích thước tối đa là 32 chi u.
- Thường sử dụng mảng kích thước 2 chi u.
- Khai báo mảng 2 chi u:
 - Ví dụ :
 - Khai báo mảng số nguyên hai chi u

`Dim IntArray(,) As Integer`

■ Ví d :

- Khai báo và kh i t o m ng kích th c 2x3

Dim intArray(2,3) As Integer

intArray(2,3) = New Integer() { {5, 15, 25}, {10, 20, 30} }

	<u>Cột 0</u>	<u>Cột 1</u>	<u>Cột 2</u>
<u>Hàng 0</u>	5	15	25
<u>Hàng 1</u>	10	20	30

■ Hai cách truy cập các phần tử của mảng:

- Thông qua các chỉ số của phần tử

- Ví dụ : `intArray(0, 1)` → 15

- Thông qua phương thức `GetValue()`

- Ví dụ : `IntArray.GetValue(0, 1)` → 15

- Là một cấu trúc dữ liệu kỹ thuật cài đặt trong VB.Net với tên ArrayList
- Tính chất:
 - ArrayList cung cấp khả năng tăng kích thước động
 - Các phần tử lưu trong ArrayList có kiểu là Object.

■ Thu c tính Capacity c a l p ArrayList:

- Dùng l u kích th c c a m ng.
- Giá tr kh i t o c a Capacity = 16.
- Khi s l ng ph n t c a m ng v t quá 16, Capacity t ng thêm 16 ph n t vào trong không gian l u tr c a m ng.

Các phương thức và thuộc tính của ArrayList

- Add()
- AddRange()
- Capacity
- Clear()
- Contains()
- CopyTo()
- Count
- GetEnumerator()
- GetRange()
- IndexOf()

Các phương thức và thuộc tính của ArrayList

- Insert()
- InsertRange()
- Item()
- Remove()
- RemoveAt()
- Reverse()
- Sort()
- ToArray()
- TrimToSize()

■ Ví dụ về sử dụng List và ArrayList

```
' Khai báo 1 ArrayList
Dim grades As New ArrayList()
Dim i As Integer
' thêm các phần tử vào ArrayList
grades.Add(10)      grades.Add(20)      grades.Add(30)

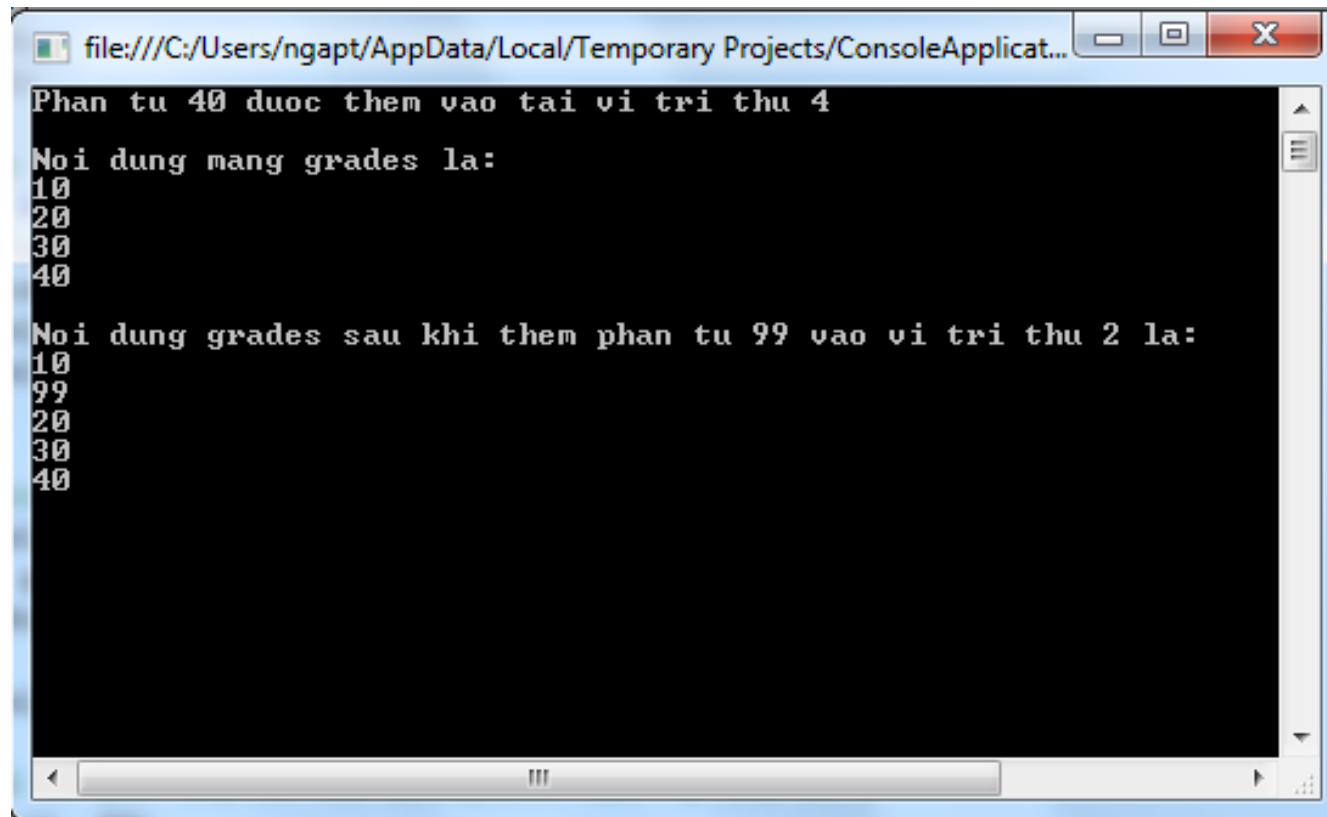
'position lưu trữ vị trí của phần tử thêm vào bằng cách Add
Dim position As Integer
position = grades.Add(40)
position = position + 1
Console.WriteLine("Phần tử 40 được thêm vào tại vị trí thứ " & position)

' Hiển thị nội dung ArrayList
Console.WriteLine("Nội dung mảng grades là: ")
For i = 0 To grades.Count - 1
    Console.WriteLine(grades.Item(i))
Next

'Thêm phần tử có giá trị 99 vào ArrayList tại vị trí số 2
grades.Insert(1, 99)
' Hiển thị nội dung ArrayList
Console.WriteLine("Nội dung grades sau khi thêm phần tử 99 vào vị trí thứ 2 là: ")
For i = 0 To grades.Count - 1
    Console.WriteLine(grades.Item(i))
Next
```

Sử dụng lớp ArrayList

- Chương trình cho kết quả :



```
file:///C:/Users/ngapt/AppData/Local/Temporary Projects/ConsoleApplicat...
Phan tu 40 duoc them vao tai vi tri thu 4
Noi dung mang grades la:
10
20
30
40
Noi dung grades sau khi them phan tu 99 vao vi tri thu 2 la:
10
99
20
30
40
```

Sử dụng List ArrayList

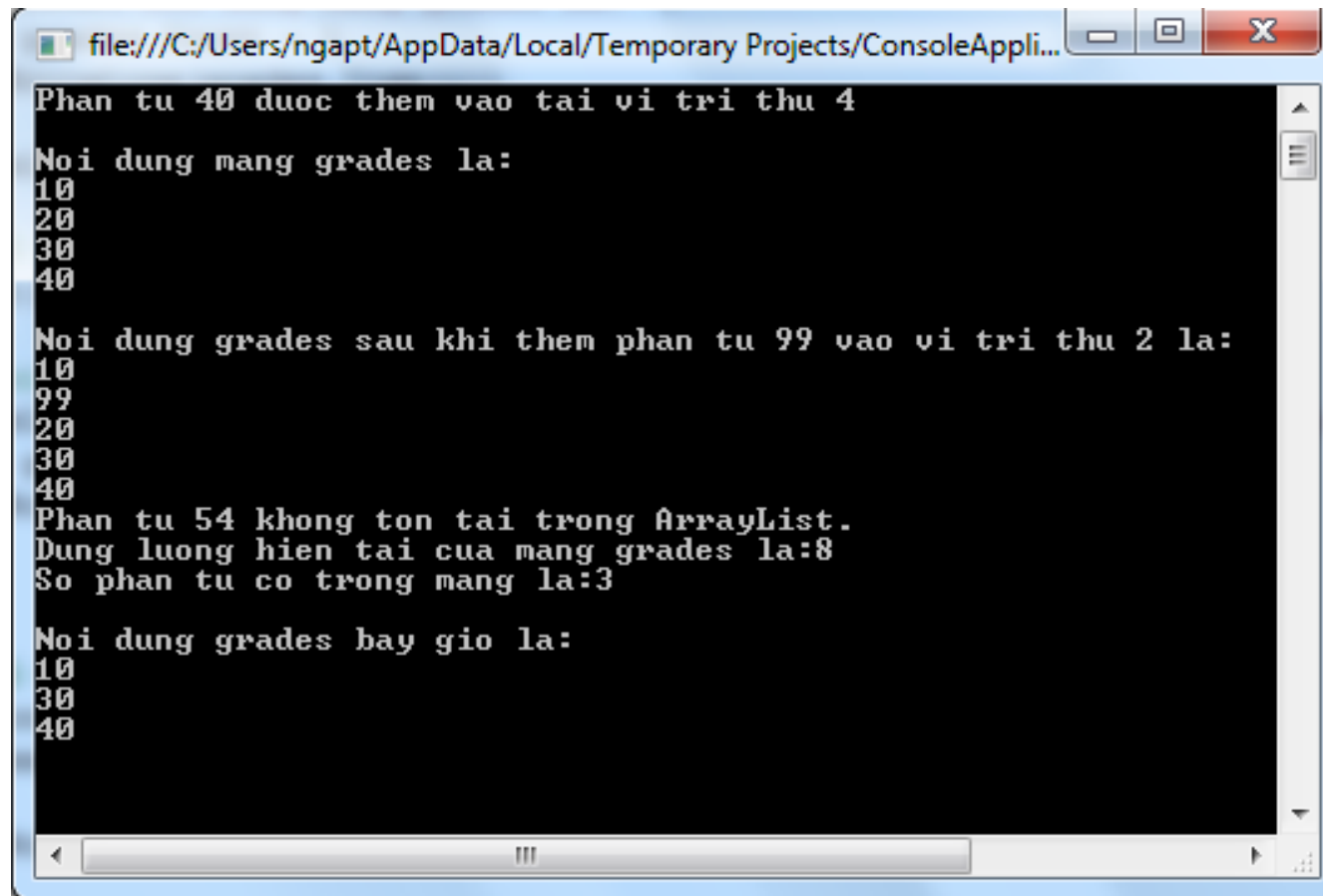
```

' Xóa phần tử có giá trị 54 khỏi ArrayList dùng Remove
If (grades.Contains(54)) Then
    grades.Remove(54)
Else
    Console.WriteLine("Phần tử này không tồn tại trong ArrayList.")
End If
'Xóa phần tử thứ 3 khỏi ArrayList dùng RemoveAt
grades.RemoveAt(2)
'Xóa phần tử khỏi danh sách dùng IndexOf kết hợp với RemoveAt
Dim pos As Integer
pos = grades.IndexOf(99)
grades.RemoveAt(pos)
' Kiểm tra dung lượng hiện tại của ArrayList
Console.WriteLine("Dung lượng hiện tại của mảng grades là:" &
grades.Capacity)
Console.WriteLine("Số phần tử có trong mảng là:" & grades.Count)
Console.WriteLine("Nội dung grades bây giờ là: ")
For i = 0 To grades.Count - 1
    Console.WriteLine(grades.Item(i))
Next
Console.ReadLine()

```

Sử dụng lớp ArrayList

- Lúc này chúng ta sẽ viết chương trình cho kết quả :



```
file:///C:/Users/ngapt/AppData/Local/Temporary Projects/ConsoleAppli...
Phan tu 40 duoc them vao tai vi tri thu 4
Noi dung mang grades la:
10
20
30
40

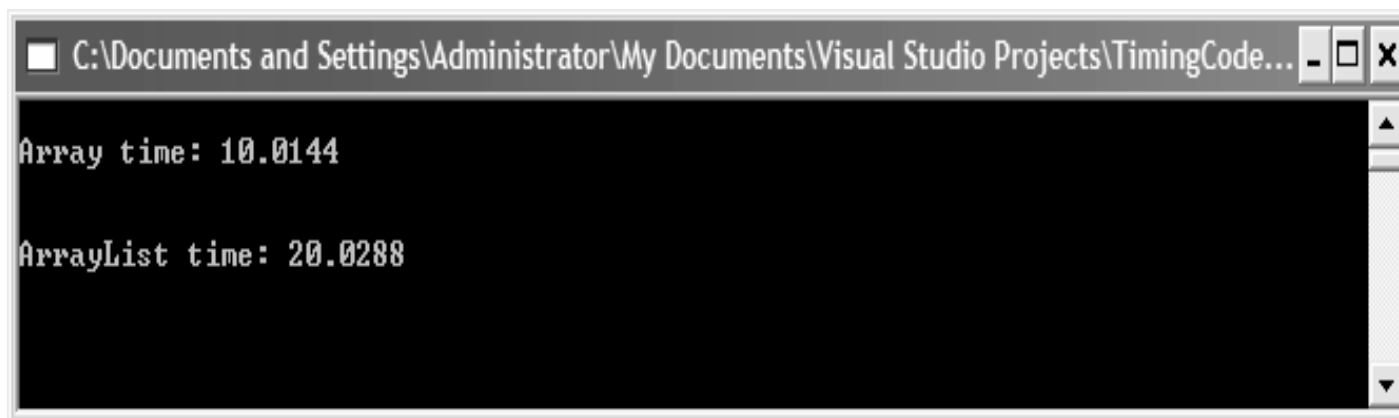
Noi dung grades sau khi them phan tu 99 vao vi tri thu 2 la:
10
99
20
30
40

Phan tu 54 khong ton tai trong ArrayList.
Dung luong hien tai cua mang grades la:8
So phan tu co trong mang la:3

Noi dung grades bay gio la:
10
30
40
```

So sánh Array và ArrayList

- Xem ví dụ SGK so sánh thời gian khi sử dụng mảng thông thường và ArrayList tính toán của 100.000 phần tử.

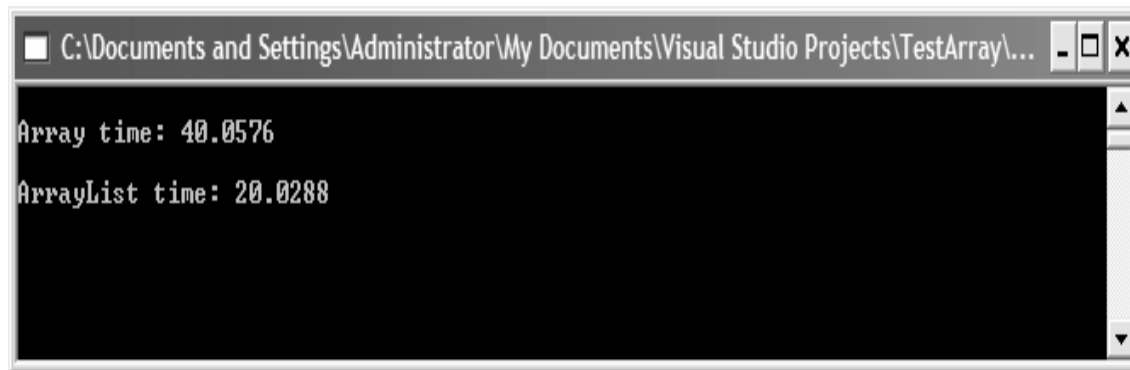


```
C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects\TimingCode...  
Array time: 10.0144  
ArrayList time: 20.0288
```

- Kết quả cho thấy nên sử dụng mảng thông thường khi tính toán với số lượng phần tử lớn.

So sánh Array và ArrayList

- So sánh thời gian khi sử dụng mảng thông thường và ArrayList thêm phần tử vào mảng đã có.



```
C:\Documents and Settings\Administrator\My Documents\Visual Studio Projects\TestArray\...  
Array time: 40.0576  
ArrayList time: 20.0288
```

- Kết quả cho thấy nên sử dụng ArrayList khi cần thực hiện các thao tác như thêm, xóa phần tử trong mảng.

- Trong lập trình, mảng là CTDL hay cấu trúc dữ liệu dùng để lưu trữ các dữ liệu cùng loại.
- Mảng trong VB.Net có nghĩa là một tập hợp các dữ liệu cùng loại.
- Cả Array và ArrayList đều cho phép thay đổi kích thước mảng.
- Dùng mảng thông thường cho bài toán tính toán trên dữ liệu đơn giản.
- Dùng cấu trúc [ArrayList](#) xây dựng mảng cho các bài toán thêm phần tử vào mảng và kiểm tra dữ liệu không phải là số.