



FPT POLYTECHNIC



Bài 3: COLLECTION TRONG VB.NET

www.poly.edu.vn

hoclaptrinhweb.com

- Nh c l i khái ni m l p trình h ng th t c (Procedural Programming)
- Tìm hi u các khái ni m và ý ngh a L p trình h ng i t ng (OOP)
- L p trình h ng i t ng trong VB.Net

Mục tiêu bài học hôm nay

■ Tìm hiểu về Collection trong VB.Net:

- Khái niệm Collection
- Phân loại các Collection
- Lập Collection trong VB.Net
- Lập CollectionBase trong VB.Net

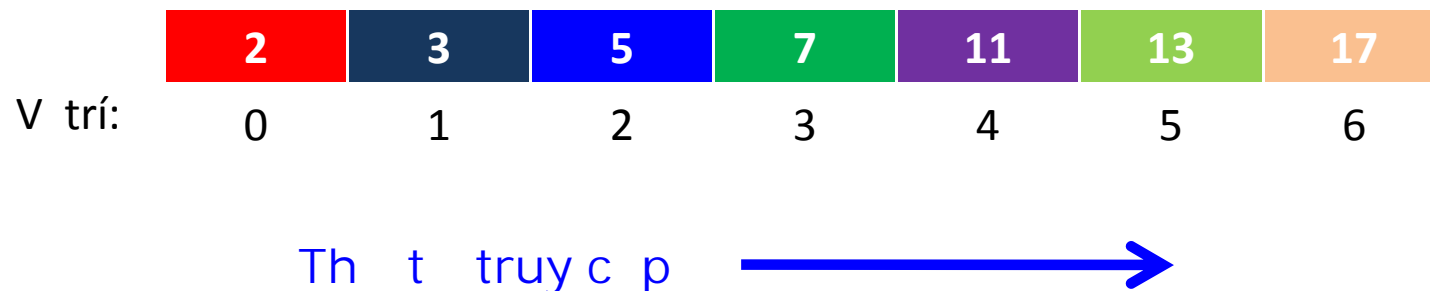
- **Collection** (tập hợp): là một kiếu dữ liệu cho phép lưu trữ tập hợp các đối tượng dữ liệu, theo một cách thức tổ chức dữ liệu nhất định (ví dụ tổ chức theo dạng tệp, cây, danh sách,...)
- Có thể phân loại thành:
 - Tập hợp tuyến tính
 - Tập hợp phi tuyến tính

Tập hợp tuyển tính

- Các phần tử sắp xếp theo vị trí
- Thuật truy cập:
 - Truy cập trực tiếp
 - Truy cập tuần tự

Tập hợp truy cập trực tiếp

- Ví dụ: Một tập hợp các phần tử có cùng kiểu dữ liệu và các phần tử truy cập trực tiếp thông qua các chỉ số là các số nguyên

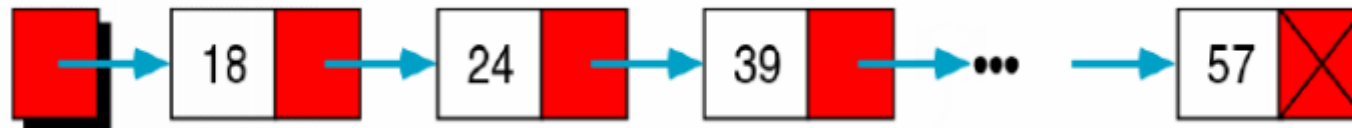


Tập hợp truy cập tuần

- Tập hợp truy cập tuần cho phép duy trì dữ liệu tập hợp một cách hiệu quả.
- Ví dụ :
 - Ngăn xếp (Stack)
 - Hàng đợi (Queue)
 - Danh sách liên kết (Linked list)

Tập hợp truy cập tuần tự

- Ví dụ: danh sách liên kết là một tập các phần tử có cùng kiểu dữ liệu và vị trí truy cập các phần tử có tính hành tuần tự theo danh sách nêu cụ thể danh sách



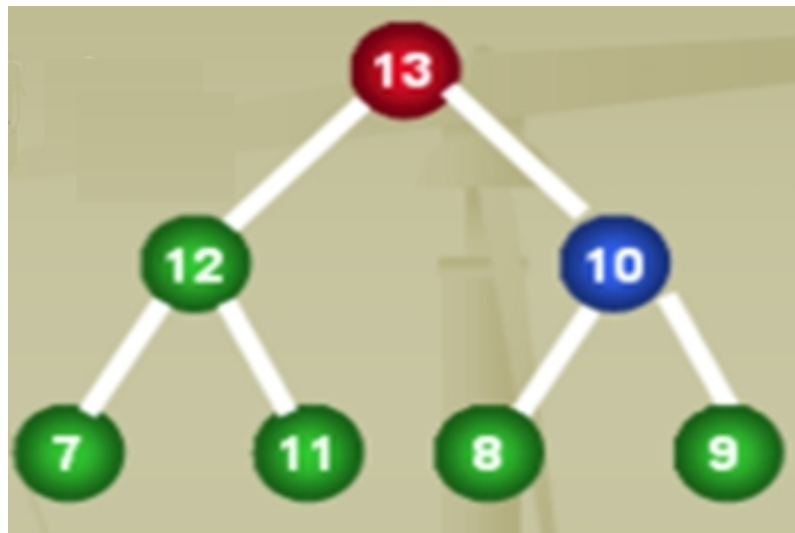
Hình minh họa một danh sách liên kết. Vị trí duy nhất danh sách này có tính hành tuần tự phần tử 18, tiếp theo là 24, 39, ...

Tập hợp phi tuyến tính

- Các phần tử không có sắp xếp theo vị trí
- Thuật truy cập:
 - Truy cập theo phân cấp. Ví dụ : Cây (Tree)
 - Truy cập theo phân nhóm. Ví dụ : ...

Truy cập theo phân cấp

- Ví dụ : Cây (Tree) là một cấu trúc phi tuyến tính
- Truy nhập các phần tử theo cấp từ nút gốc đến nút lá -> cây biểu diễn mạng {13, 12, 10, 7, 11, 8, 9}



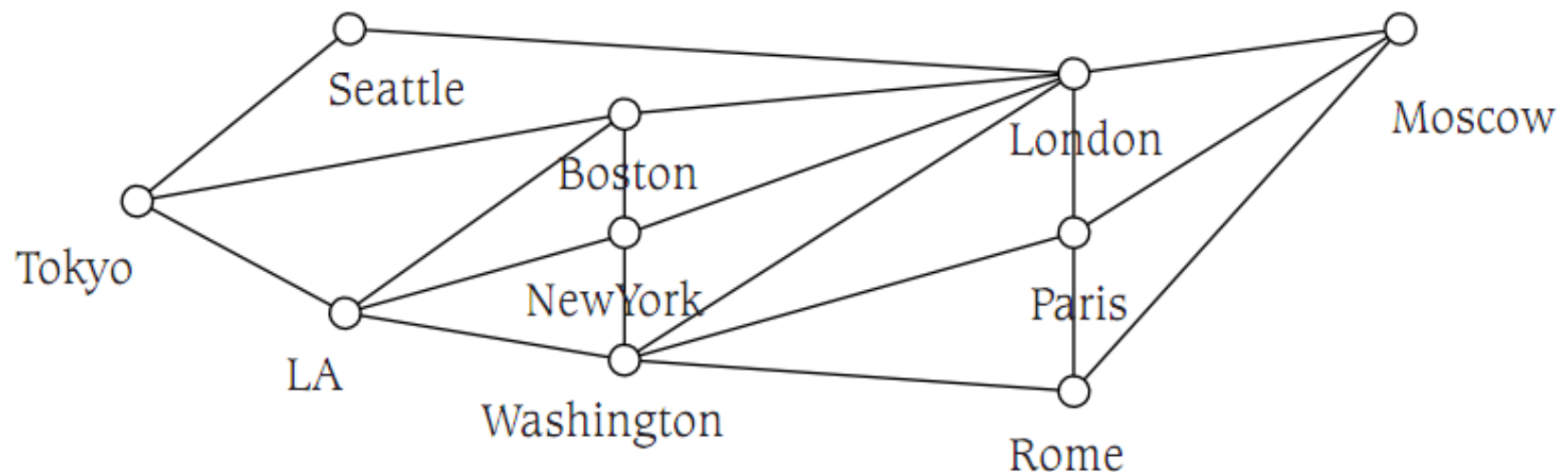
Hình minh họa một cấu trúc dữ liệu dạng cây

Truy cập theo phân nhóm

- Tập phi tuyến tính vì các phần tử không có sự phụ thuộc lẫn nhau là **nhóm**
- Tập hợp nhóm có phân chia thành ba nhóm chính là:
Set, **th (graph)** và **mạng l (network)**

Truy cập theo phân nhóm

- Ví dụ : mô hình mạng liên lạc giữa các thành phố



Hình minh họa mô hình mạng liên lạc giữa các thành phố

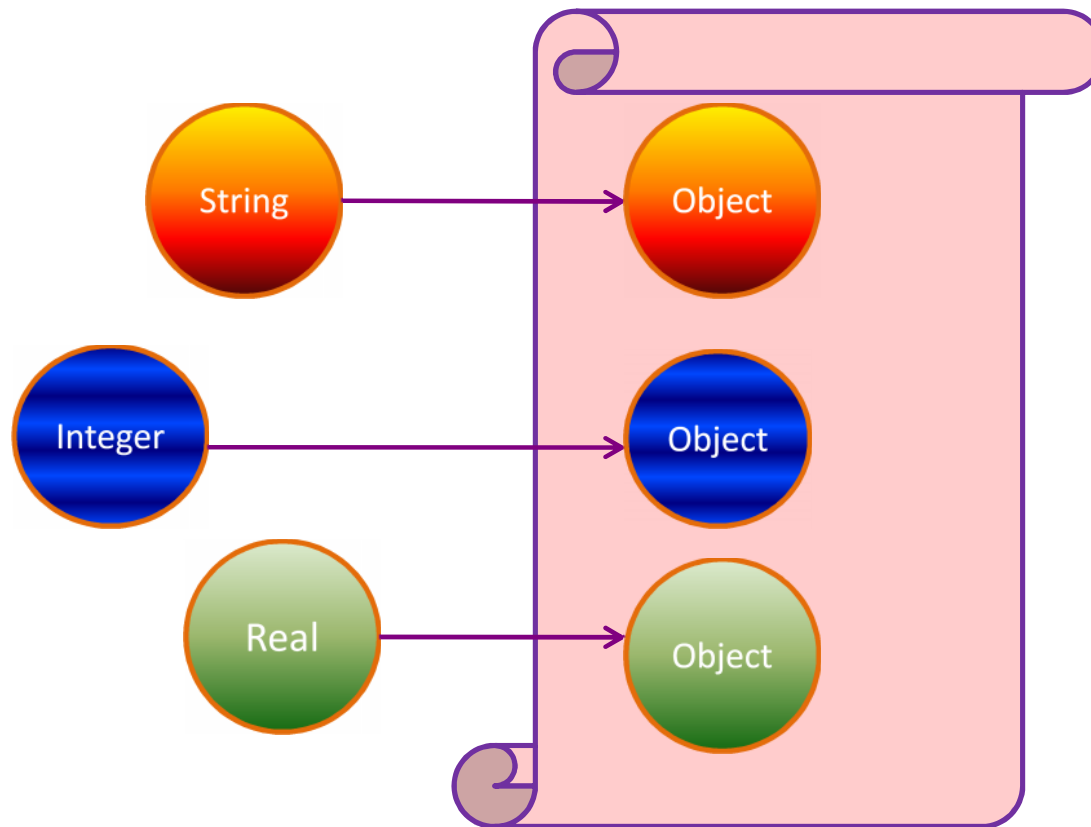
Cài đặt tập hợp trong VB.Net

- Có hai lớp cài đặt cấu trúc dữ liệu collection trong VB.Net
 - Sử dụng lớp **Collection**
 - Sử dụng lớp **CollectionBase**

■ Tính chất:

- Các phần tử không ghi index vào tập hợp
- Các phần tử có kiểu bất kỳ có thể thêm vào collection nhưng lưu trữ kiểu Object.

Sử dụng List Collection



Hình minh họa các phần tử có kiểu bất kỳ được lưu trong Collection với kiểu là Object

- Cách khai báo: **Dim names As New Collection**

- Trong đó:

- “names” là tên của collection

Sử dụng List Collection

- List Collection cung cấp các tính chất:

Thuộc tính (Property)	Giải thích
Count	cho biết số phần tử có trong collection
Item	trả về một phần tử của collection theo chỉ số hoặc khóa

- List Collection cung cấp các phương thức chính:

Add	thêm phần tử vào collection
Remove	xóa một phần tử khỏi collection
Clear	xóa tất cả các phần tử của collection

Có thể xem chi tiết API của List trên MSDN

<http://msdn.microsoft.com/en-us/library/system.collections.aspx>

Ph ng th c Add

- Ví d : dùng ph ng th c **Add** thêm d li u vào Collection

```
Dim names As New Collection

' Thêm tên các ph n t  vào Collection
names.Add("David Durr")
names.Add("Raymond Williams")
names.Add("Bernica Tackett")
names.Add("Beata Lovelace")
```

Ph ng th c Add

- Ví d : Thêm tên các ph n t vào t p h p cùng v i khóa (key) i kèm. L u ý: khóa c a m i ph n t ph i là bi u th c String duy nh t

```
Dim names As New Collection

' Thêm tên các ph  n t  vào Collection
names.Add("David Durr", "300")
names.Add("Raymond Williams", "301")
names.Add("Bernica Tackett", "302")
names.Add("Beata Lovelace", "303")

'Hi  n th  s  ph  n t  có trong Collection
Console.WriteLine(names.Count)
```

- Ví d : thêm các ph n t vào t p h p cùng v i khóa và ch nh v trí thêm vào trong t p h p.

```
Dim names As New Collection
```

```
names.Add("David Durr", "300")
```

```
names.Add("Raymond Williams", "301")
```

```
'Ch  nh v  trí thêm vào tr  c ph  n t  u tiên
```

```
names.Add("Bernica Tackett", "302", 1)
```

```
'Ch  nh v  trí thêm vào tr  c ph  n t  th  3
```

```
names.Add("Beata Lovelace", "303", 3)
```

Thu_c tính Item

- Ví d : Hi n th ra màn hình thông tin m i ph n t trong collection thông qua thu_c tính Item truy n theo ch s :

```
' Khai báo bi n index
Dim index As Integer

'Hi n th ra màn hình
For index = 1 To names.Count
    Console.WriteLine(names.Item(index))
Next
```

Thu_c tính Item

- Ví d : Hi n th ra màn hình thông tin m i ph n t trong t p h p thông qua thu_c tính Item truy n theo khóa (v i tr ng h p các ph n t có ch s b t u t 300):

```
Dim index As Integer 'khai báo bi n index
Dim key As Integer = 300 'khai báo bi n key
Dim name As String 'khai báo bi n l p

'Hi n th ra màn hình tên m i ph n t trong
Collection
For Each name In names
    key = CStr(index)
    Console.WriteLine(names.Item(key))
    index += 1
Next
```

Chuyển kiểu tập hợp List Collection

- Khi List tập hợp trả về Collection, tập hợp sẽ có kiểu List. Chuyển kiểu tập hợp mình, cần dùng hàm chuyển CType

```
Option Strict On
Dim myString As String = "This is my String"
Dim aString As String
Dim myCollection As New Collection()
myCollection.Add(myString)
aString = CType(myCollection.Item(1), String)
```

Sử dụng Lớp Collection

■ Nhận xét:

- Các phần tử có kiểu bất kỳ có thể thêm vào collection nhưng lưu trữ với kiểu Object
- Do đó các phương thức dành riêng cho các lớp String, Integer, Decimal ... không có giá trị cụ thể, phải sử dụng hàm CType chuyển các Object về kiểu dữ liệu ban đầu (String, Integer, Decimal...)
- Có thể thêm nhiều đối tượng với kiểu khác nhau vào cùng một collection

➔ Sử dụng Lớp CollectionBase cài đặt phương thức để phân biệt rõ ràng cho các phần tử.

- Demo và giới thiệu cài đặt lập collection riêng trên VB.NET (trong SGK)

Lập CollectionBase

- Là lớp định nghĩa abstract, cung cấp khung xây dựng các lớp implementer để hỗ trợ các collection

Phương thức	Mô tả
Add	Thêm một đối tượng vào cuối collection.
Insert	Chèn thêm một phần tử vào collection tại vị trí chỉ định.
Remove	Xóa bỏ phần tử đầu tiên tìm thấy trong collection giống với đối tượng được truyền vào.
Contains	Xác định vị trí mà collection chứa phần tử được truyền vào.
IndexOf	Tìm kiếm phần tử được truyền vào và trả về chỉ số (tính từ 0) của phần tử đầu tiên trong collection thỏa mãn điều kiện.
CopyTo	Sao chép toàn bộ collection từ vị trí bắt đầu đến vị trí kết thúc vào mảng một chiều.

Sử dụng Lớp CollectionBase

- Bao gồm các phương thức public chính sau:
 - **Clear:** Xóa bỏ tất cả các đối tượng collection
 - **RemoveAt:** Xóa một phần tử tại vị trí chỉ định trong collection
 - **Equals:** Xác minh xem đối tượng này có bằng với đối tượng kia hay không.
 - **ToString:** Trả về một chuỗi thể hiện đối tượng này.
 - **GetEnumerator:** Trả về một đối tượng liệt kê để duyệt toàn bộ collection.

Demo v CollectionBase

- Demo và giới thiệu cách cài đặt lớp StudentCollection kế thừa từ lớp CollectionBase (trong SGK)

Namespace System.Collection

- Ngoài lớp `CollectionBase`, trong namespace `System.Collection` còn có nhiều lớp khác collection, ví dụ dưới đây là một số lớp như sau:

Tên lớp	Ý nghĩa
<u>ArrayList</u>	Biểu diễn tập các đối tượng có thể thay đổi
<u>Hashtable</u>	Biểu diễn tập hợp theo cặp khóa/giá trị
<u>Queue</u>	Biểu diễn cấu trúc queue, vào-trước-ra-trước
<u>SortedList</u>	Biểu diễn tập hợp theo cặp khóa/giá trị, đã sắp xếp theo khóa, truy cập theo khóa hoặc chỉ số
<u>ReadOnlyCollectionBase</u>	Lớp abstract để cài đặt collection chỉ đọc
<u>Stack</u>	Biểu diễn cấu trúc stack, vào-sau-ra-trước

Chi tiết xem thêm tại:

<http://msdn.microsoft.com/en-us/library/system.collections%28v=VS.110%29.aspx>

- **Collection** là một kiểu dữ liệu có cấu trúc một nhóm không ghi vị trí và kiểu phần tử
- Hai loại collection:
 - Tập hợp tuyến tính: các phần tử sắp xếp theo vị trí.
 - Tập hợp phi tuyến tính: các phần tử không sắp xếp theo vị trí.
- Phần tử trong collection có kiểu là Object

- Hai cách cài đặt collection trong VB.Net:
 - Sử dụng ngay lớp **Collection** để tạo ra collection
 - Tạo lớp mới kế thừa từ **CollectionBase**. kế thừa các phương thức do người dùng định nghĩa.