Collection trong Java

Java cung cấp các class đặt biệt như **Dictionary, Vector, Stack** và **Properties** để lưu trữ và thao tác với một nhóm các đối tượng. Mặc dù những class này khá hữu dụng, nhưng lại thiếu sự tập trung, thống nhất. Do đó, cách sử dụng Vector trong Java khác với cách sử dụng Properties.

Để hiểu sâu hơn các khái niệm được trình bày trong chương này, mời bạn tham khảo loạt bài: **Ví dụ về Collection trong Java**.

Collection framework được thiết kế với mục đích như sau:

- Framework phải là hiệu năng cao. Sự triển khai cho các tập hợp cơ bản (các mảng động, linked list, tree và hashtable) được sử dụng với hiệu quả cao.
- Framework phải cho phép các kiểu tập hợp khác nhau để làm việc theo một cách tương tự như nhau với độ phân hóa ở mức cao.
- Kế thừa và/hoặc tìm hiểu với các tập hợp phải là dễ dàng.

Đến cuối cùng, toàn bộ collection framework này được thiết kế xung quanh một tập hợp các interface tiêu chuẩn. Vài class triển khai như **LinkedList, HashSet** và **TreeSet** của những interface này được cung cấp cho bạn có thể triển khai tập hợp nếu bạn chọn.

Một collections framework là một cấu trúc thống nhất để biểu diễn và thao tác các collection. Tất cả collections framework đều chứa:

- **Interface:** Đây là các kiểu dữ liệu abstract mà biểu diễn collection. Interface cho phép collection được thao tác một cách độc lập theo phép biểu diễn của chúng. Trong ngôn ngữ hướng đối tượng, các interface nói chung cấu tạo nên một hierarchy.
- **Sự triển khai, ví dụ như các Class** Đây là sự triển khai cụ thể của collection interface. Về bản chất, chúng là những cấu trúc dữ liệu có thể tái sử dụng.
- **Thuật toán:** Đây là các phương thức thực hiện các trình tính toán hữu ích, như tìm kiếm và xếp thứ tự phân loại, trên các đối tượng mà triển khai collection interface. Các thuật toán được xem như là đa hình: đó là, cùng một phương thức có thể được sử dụng trên nhiều sự triển khai khác nhau của collection interface thích hợp.

Ngoài ra, framework định nghĩa một số map interfaces và class. Map lưu giữ các cặp key/value. Mặc dù các map không là *collections* về khái niệm, nhưng chúng hoàn toàn tương thích với collection.

Collection Interface trong Java

Một Collection framework định nghĩa trước vài loại interface. Phần này cung cấp cho bạn tổng quan về mỗi interface:

STT	Interface và Miêu tả
1	**Collection Interface trong Java** Cho phép bạn có thể làm việc với các nhóm đối tượng, nó là phần interface cha của tất cả các collection interface khác
2	**List Interface trong Java** Kế thừa từ **Collection** và một kiểu List lưu trữ tập hợp các phần tử một cách có thứ tự
3	**Set Interface trong Java** Nó kế thừa Collection để thao tác với tập hợp, mà phải chứa các phần tử duy nhất
4	**SortedSet Interface trong Java** Kế thừa từ Set để thao tác với tập hợp được xếp thứ tự
5	**Map Interface trong Java** Liên kết giữa key duy nhất tới các value
6	**Map.Entry Interface trong Java** Mô tả một thành phần key/value trong map. Là một inner class của Map

7	**SortedMap Interface trong Java** Kế thừa từ Map để mà các key được duy trì một cách có thứ tự
8	**Enumeration Interface trong Java** Đây là một legacy interface và định nghĩa phương thức, theo đó bạn có thể liệt kê các thành phần của một tập hợp trong một tập hợp các đối tượng.

Các lớp Collection trong Java

Java cung cấp một tập hợp các lớp collection tiêu chuẩn có thể triển khai các Collection interface.

Bảng dưới đây tổng hợp các lớp collection chuẩn trong Java:

STT	Các lớp và miêu tả
1	**Lớp AbstractCollection trong Java** Triển khai tất cả các Collection interface
2	**Lớp AbstractList trong Java** Kế thừa AbtractCollection và triển khai tất cả phương thức List interface
3	**Lớp AbstractSequentialList trong Java** Kế thừa AbstractList để sử dụng bởi một Collection mà sử dụng liên tục thay vì truy cập ngẫu nhiên các phần tử của nó
4	**Lớp LinkedList trong Java** Triển khai một LinkedList bởi kế thừa AbstractSequentialList
5	**Lớp ArrayList trong Java**

	Triển khai một mảng động bởi kế thừa AbstractList
6	**Lớp AbstractSet trong Java** Kế thừa AbstractCollection và triển khai hầu hết Set interface
7	**Lớp HashSet trong Java** Kế thừa AbstractSet để sử dụng với một hash table
8	**Lớp LinkedHashSet trong Java** Kế thừa HashSet để cho phép lặp lại thứ tự chèn (insertion-order)
9	**Lớp TreeSet trong Java** Triển khai một tập hợp được lưu trong một tree. Kế thừa AbstractSet
10	**Lớp AbstractMap trong Java** Triển khai hầu hết Map interface
11	**Lớp HashMap trong Java** Kế thừa AbstractMap để sử dụng một hash table
12	**Lớp TreeMap trong Java** Kế thừa AbstractMap để sử dụng một tree
13	**Lớp WeakHashMap trong Java** Kế thừa AbstractMap để sử dụng một hash table với các khóa weak

14	**Lớp LinkedHashMap trong Java** Kế thừa HashMap để cho phép lặp lại thứ tự chèn (insertion-order)
15	**Lớp IdentityHashMap trong Java** Kế thừa AbstractMap và sử dụng tham chiếu ngang bằng khi so sánh các tài liệu

Các lớp *AbstractCollection, AbstractSet, AbstractList, AbstractSequentialList* và *AbstractMap*trong Java cung cấp sự triển khai xương sống của các Collection Interface lõi, để tối thiểu hóa nỗ lực cần để thi hành chúng.

Các legacy class sau được định nghĩa bởi java.util, đã được bàn luận trong chương trước:

STT	Các lớp và Miêu tả
1	**Lớp Vector trong Java** Lớp này triển khai một mảng động. Nó tương tự như ArrayList, nhưng có một số điểm khác nhau
2	**Lớp Stack trong Java** Stack là lớp phụ của lớp Vector mà triển khai last-in-first-out stack
3	**Lớp Dictionary trong Java** Dictionary là một abstract class mà biểu diễn một kho lưu giữ key/value và hoạt động khá giống Map
4	**Lớp Hashtable trong Java** Hashtable là một phần của java.util gốc và là một sự triển khai cụ thể của một Dictionary

5	**Lớp Properties trong Java** Properties là một lớp phụ của Hashtable. Nó được sử dụng để duy trì các danh sách giá trị trong đó key là một String và value cũng là một String
6	**Lớp BitSet trong Java** Một lớp BitSet tạo một kiểu mảng đặc biệt mà giữ các giá trị bit. Mảng này có thể tăng kích cỡ nếu cần

Các thuật toán Collection trong Java

Collection Framework định nghĩa một số thuật toán có thể được áp dụng cho các Collection và Map. Những thuật toán này được định nghĩa như là các phương thức tĩnh (static) bên trong lớp Collection.

Một số phương thức có thể ném một **ClassCastException**, mà xảy ra khi cố gắng so sánh các kiểu không tương thích, hoặc một **UnsupportedOperationException**, mà xảy ra khi cố gắng sửa đổi một Unmodifiable Collectioon.

Các Collection định nghĩa 3 biến static là: EMPTY_SET, EMPTY_LIST, và EMPTY_MAP. Tất cả là không thể thay đổi.

STT	Thuật toán và Miêu tả
1	**Thuật toán Collection trong Java** Đây là danh sách tất cả các thuật toán.

Cách sử dụng một Iterator trong Java

Thường thì, bạn sẽ muốn tuần hoàn qua các phần tử trong một tập hợp. Ví dụ, có thể bạn muốn hiển thị mỗi phần tử.

Cách đơn giản nhất để thực hiện điều này là thuê một Iterator, là một đối tượng mà triển khai hoặc Iterator hoặc Listlerator interface.

Iterator cho bạn khả năng để tuần hoàn qua một tập hợp, kiếm được và gỡ bỏ các phần tử. ListIterator kế thừa Iterator để cho phép "vọc" song hướng một danh sách và sửa đổi các phần tử.

STT	Các phương thức Iterator và miêu tả
1	**Sử dụng Iterator trong Java** Đây là danh sách tất cả phương thức và ví dụ về Iterator và ListIterator interface.

Cách sử dụng một Comparator trong Java

Cả TreeSet và TreeMap đều lưu giữ các phần tử trong thứ tự đã được xếp thứ tự. Tuy nhiên, nó là comparator mà định nghĩa chính xác ý nghĩa của *sorted order*.

Interface này giúp chúng ta xếp thứ tự một tập hợp với bất kỳ số lượng mảng nào đã cho. Ngoài ra, interface này cũng có thể được sử dụng để xếp thứ tự bất kỳ instance nào của bất kỳ lớp nào (ngay cả các lớp chúng ta không thể chỉnh sửa).

STT	Phương thức Comparator và Miêu tả
1	**Sử dụng Comparator trong Java** Đây là danh sách tất cả các phương thức và ví dụ về Comparator Interface.

Tổng kết

Collection Framework trong Java cung cấp cho lập trình viên truy cập tới các cấu trúc dữ liệu đã đóng gói trước (Prepackage) cũng như các thuật toán để thao tác chúng.

Một Collection là một đối tượng mà có thể giữ các tham chiếu tới các đối tượng khác. Collection Interface khai báo các hoạt động mà có thể được thực hiện trên mỗi kiểu Collection.

Các Class và Interface của Collection Framework là trong gói java.util.