**Tổng quan về Collections trong Java.**

**1. Khái niệm về Collections (tập hợp)**

**Collections là một tập các lớp dùng để lưu trữ danh sách và có khả năng tự co dãn khi danh sách đó thay đổi**, ví dụ như khi chúng ta thêm, sửa, xóa, chèn phần tử trong danh sách đó. Ngoài ra, Collections còn được dùng để lưu trữ, truy xuất, tương tác với dữ liệu và truyền dữ liệu giữa các phương thức với nhau (chi tiết về phương thức tôi sẽ giới thiệu trong chương Lập trình hướng đối tượng).

Một đặc điểm rất quan trong là khi sử dụng Collections đó là chúng ta không cần phải khai báo trước số lượng phần tử. Chính đặc điểm này đã khắc phục được hạn chế về kích thước khi khai báo mảng trong Java.

**2. Collections Framework**

Trong phần 1, tôi đã giới thiệu với các bạn sơ lược về Collections. Vậy Collections Framework là gì? Trước khi tìm hiểu về vấn đề này, các bạn cần phải hiểu về khái niệm **Framework**: "Framework là một tập hợp các thư viện (Library) đã được đóng gói để hỗ trợ phát triển ứng dụng dựa trên Framework đó. Đồng thời, Framework cung cấp các nguyên tắc, cấu trúc của ứng dụng mà chúng ta phải tuân thủ theo nó."

Ví dụ về Framework: Trong lập trình Java, chúng ta có Framework Spring, đây là một Framework phổ biến dùng để lập trình web với Java. Để sử dụng được Framework này thì chúng ta phải tuân thủ theo mô hình và cách hoạt động của nó, nghĩa là Spring Framework đều đã viết sẵn các thư viện để chúng ta sử dụng lại cùng với các khuôn mẫu (design pattern), cấu trúc của ứng dụng,... mà chúng ta phải tuân theo nó để có thể xây dựng được ứng dụng.

Sau khi đã tìm hiểu về Framework thì **Collections Framework** được hiểu như sau: "Một Java Collections Framework là một tập hợp các lớp (class) và các interface dùng để hỗ trợ việc thao tác trên tập các đối tượng" (chi tiết về interface tôi sẽ giới thiệu trong chương Lập trình hướng đối tượng, nhưng bạn có thể hiểu nôm na là 1 interface là 1 lớp rỗng chỉ chứa khai báo về tên phương thức, không có khai báo về thuộc tính hay thứ gì khác và các phương thức này cũng là phương thức rỗng).

Trong Java, các Collections Framework cung cấp những thành phần sau:

| **Loại thành phần** | **Mô tả** |
| --- | --- |
| Interfaces | Kiểu dữ liệu trừu tượng (abstract) biểu diễn Collections (chi tiết về abstract tôi sẽ giới thiệu trong chương Lập trình hướng đối tượng). |
| Implementations | Là sự triển khai các Interface, ví dụ như các Class. |
| Algorithms (các thuật toán) | Là các phương thức dùng để thực thi các phép toán như tìm kiếm và sắp xếp trên các đối tượng mà triển khai các Interface. |

**3. Interface Collections, Class Collections và Algorithms**

Collections Framework trong Java bao gồm 2 loại chính đó là **Interface Collections** và **Class Collections**, mỗi loại này sẽ có những đặc điểm và thành phần của riêng nó. Sau đây chúng ta sẽ cùng nhau tìm hiểu về 2 loại này.

**3.1. Interface Collections**

Là một tập hợp đại diện cho một nhóm các đối tượng, và được gọi là các phần tử (elements).

Một số Interface Collection cho phép lưu trữ các phần tử giống nhau, còn một số khác thì không. Ngoài ra, các phần tử này có thể có thứ tự hoặc không có thứ tự tùy theo từng loại Collection khác nhau.

Bao gồm các phương thức như thêm (add), xóa (clear), so sánh (compare) và duy trì (retainining) các đối tượng.

Trong Interface Collections chúng ta có các Interface chính như: List Interface, Set, SortedSet, Map và SortedMap. Bảng dưới đây sẽ mô tả khái quát về các Interface này:

| **Tên Interface** | **Đặc điểm khái quát** |
| --- | --- |
| [List Interface](https://freetuts.net/list-interface-trong-java-1077.html" \o "list interface trong java 1077 html" \t "https://freetuts.net/_blank) | Các phần tử trong List Interface được sắp xếp có thứ tự và có thể có giá trị giống nhau. |
| [Set](https://freetuts.net/set-interface-trong-java-1080.html" \o "set interface trong java 1080 html" \t "https://freetuts.net/_blank) | Các phần tử trong Set là duy nhất (nghĩa là giá trị của các phần tử này không được giống nhau). |
| [SortedSet](https://freetuts.net/sortedset-interface-trong-java-1086.html" \o "sortedset interface trong java 1086 html" \t "https://freetuts.net/_blank) | Là 1 dạng riêng của Set Interface, trong đó giá trị của các phần tử mặc định được sắp xếp tăng dần. |
| [Map](https://freetuts.net/map-interface-trong-java-1087.html" \o "map interface trong java 1087 html" \t "https://freetuts.net/_blank) | Giá trị của mỗi phần tử trong Map bao gồm 2 phần đó là khóa (key) và giá trị tương ứng của key đó (value) và khóa của các phần tử này là duy nhất. |
| [SortedMap](https://freetuts.net/sortedmap-interface-trong-java-1088.html" \o "sortedmap interface trong java 1088 html" \t "https://freetuts.net/_blank) | Là 1 dạng riêng của Map Interface, trong đó giá trị key được sắp xếp tăng dần. |

### **3.2. Class Collections**

Java cung cấp một tập hợp các lớp tiêu chuẩn dùng để triển khai các Interface Collection. Trong Class Collections chúng ta có rất nhiều loại nhưng trong phạm vi của series Lập trình Java căn bản này thì chúng ta chỉ cần nắm 6 loại chính sau: LinkedList, ArrayList, HashSet, TreeSet, HashMap và TreeMap. Bảng dưới đây sẽ mô tả khái quát về các Class này:

| **Tên Class** | **Đặc điểm khái quát** |
| --- | --- |
| [LinkedList (Danh sách liên kết)](https://freetuts.net/linkedlist-danh-sach-lien-ket-trong-java-1089.html" \o "linkedlist danh sach lien ket trong java 1089 html" \t "https://freetuts.net/_blank) | Là 1 cấu trúc dữ liệu lưu trữ các phần tử dưới dạng danh sách. Các phần tử trong LinkedList được sắp xếp có thứ tự và có thể có giá trị giống nhau. |
| [ArrayList](https://freetuts.net/arraylist-trong-java-1091.html" \o "arraylist trong java 1091 html" \t "https://freetuts.net/_blank) | Là kiểu danh sách sử dụng cấu trúc mảng để lưu trữ phần tử. Thứ tự các phần tử dựa theo thứ tự lúc thêm vào và giá trị của các phần tử này có thể trùng nhau. |
| [HashSet](https://freetuts.net/hashset-trong-java-1094.html" \o "hashset trong java 1094 html" \t "https://freetuts.net/_blank) | Thứ tự các phần tử trong HashSet không dựa theo thứ tự lúc thêm vào và giá trị của các phần tử này là duy nhất. |
| [TreeSet](https://freetuts.net/treeset-trong-java-1096.html" \o "treeset trong java 1096 html" \t "https://freetuts.net/_blank) | Các phần tử trong TreeSet mặc định được sắp xếp tăng dần và giá trị của các phần tử này là duy nhất. |
| [HashMap](https://freetuts.net/hashmap-trong-java-1099.html" \o "hashmap trong java 1099 html" \t "https://freetuts.net/_blank) | Giá trị của mỗi phần tử trong HashMap bao gồm 2 phần đó là khóa (key) và giá trị tương ứng của key đó (value) và khóa của các phần tử này là duy nhất. HashMap cho phép truy xuất trực tiếp dữ liệu bằng khóa duy nhất của nó. |
| TreeMap | Giá trị của mỗi phần tử trong TreeMap bao gồm 2 phần đó là khóa (key) và giá trị tương ứng của key đó (value) và khóa của các phần tử này là duy nhất. Giá trị của các phần tử trong TreeMap được sắp xếp tăng dần. |

### **3.3. Algorithms (các thuật toán)**

Collections Framework trong Java cung cấp các thuật toán để các Collection có thể sử dụng được. Chi tiết về các thuật toán này tôi sẽ trình bày trong các bài sau.

**4. Iterator (lặp) trong Java**

Trong các bài trước, khi muốn duyệt qua các phần tử của một đối tượng nào đó thì thông thường chúng ta hay sử dụng các loại vòng lặp như for, while hoặc do - while. Nhưng đối với Collections thì chúng ta sử dụng một cách mới để duyệt qua các phần tử của một Collection đó là **Iterator**.

Vậy Iterator dùng để làm gì? Đối với Collections, **Iterator**là một Interface cung cấp một số các phương thức để duyệt (lặp) qua các phần tử của bất kỳ tập hợp nào. Ngoài ra, Iterator còn có khả năng xóa những phần tử của một tập hợp trong quá trình lặp.

**5. Interface Comparable và Comparator**

Java cung cấp 2 Interface **Comparable** và **Comparator** để so sánh và sắp xếp vị trí 2 đối tượng của một Collection. Chi tiết về 2 Interface này chúng ta sẽ tìm hiểu khi bước sang chương Lập trình hướng đối tượng.