

# Collection

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

# Nội dung

1. Collections
2. List Interface
3. ArrayList
4. Vector
5. LinkedList

# Collections

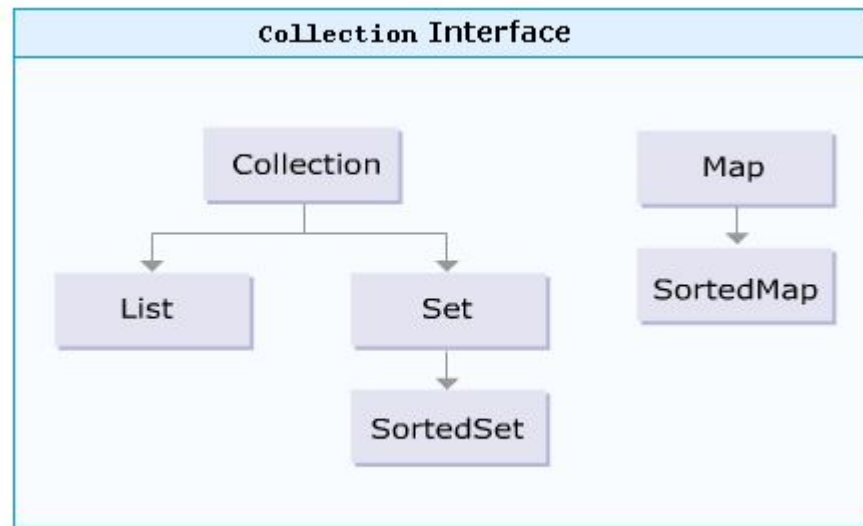
**Collections** là một gói các thư viện được cung cấp sẵn bởi **Java** trong gói **java.util**.

- 1 collection giống như 1 container (thùng chứa) có thể gom tất cả các đối tượng vào để quản lý bằng 1 thể hiện duy nhất
- cung cấp cho ta rất nhiều công cụ, lớp để lưu trữ theo nhiều dạng khác nhau.
- tăng hiệu năng làm việc (lưu trữ, truy vấn)

# Collections

## Collection Interface

- Là Interface ở mức trừu tượng, định nghĩa ra các thành phần cơ bản cho các Collection
- Một số Lớp kế thừa từ Collection
  - + List
  - + Set
  - + Queue



# Collections

## Một số phương thức cơ bản của Collection

- `add(E obj)`
- `contains(Object obj)`
- `isEmpty()`
- `size()`

# List Interface

- **List interface** là lớp kế thừa lại từ **Collection Interface**
- Cho phép chúng ta lưu trữ các dữ liệu có thứ tự
- Dữ liệu lưu trữ có thể trùng lặp.
- Cho phép thêm phần tử vào vị trí xác định

# List Interface

## Một số phương thức cơ bản của List

- `add(int index, Object obj)`
- `addAll(int index, Collection c)`
- `get(int index)`
- `set(int index, Object obj)`
- `remove(int index)`
- `subList(int start, int end)`

# ArrayList

- **ArrayList** là lớp thực thi của **interface List**
- Có thể chứa tất cả các kiểu dữ liệu, và không nhất thiết các phần tử phải có dữ liệu trùng nhau giống như mảng

## Contrucstor

- `ArrayList()`
- `ArrayList(Collection c)`
- `ArrayList(int initialCapacity)`



# ArrayList

## Một số phương thức cơ bản của List

- `add(E obj)`
- `trimToSize()`
- `clear()`
- `contains(Object obj)`
- `size()`

# ArrayList

## Ví dụ

```
List list = new ArrayList();  
list.add(1);  
list.add("Hello");  
list.add("World");
```

# Vector

- **Vector** là lớp thực thi của **interface List**
- Tính năng tương tự như ArrayList
- Điểm khác duy nhất là các phương thức của Vector đều là các phương thức synchronised

## Contrucstor

- `Vector()`
- `Vector(Collection c)`
- `Vector(int initialCapacity)`

# Vector

## Một số phương thức cơ bản của Vector

- `addElement(E obj)`
- `capacity()`
- `toArray()`
- `elementAt(int pos)`
- `removeElement(Object obj)`
- `clear()`

# ArrayList

## Ví dụ

```
List list = new Vector();  
list.add(1);  
list.add("Hello");  
list.add("World");
```

# LinkedList

- **LinkedList** là lớp thực thi của **interface List**
- Là một ds đặc biệt
- Các phần tử được móc nối với nhau bằng 1 liên kết
- Phần tử trước giữ liên kết đến phần tử sau

## Contrucstor

- **LinkedList()**
- **LinkedList(Collection c)**

# LinkedList

## Một số phương thức cơ bản của Vector

- `addFirst(E obj)`
- `addLast(E obj)`
- `getFirst()`
- `getLast()`
- `removeFirst()`
- `removeLast()`

# LinkedList

## Ví dụ

```
List list = new LinkedList();  
list.add(1);  
list.add("Hello");  
list.add("World");
```