

Chương 07:

Sử dụng Collection trong JAVA

1. *Collections bạn là ai ?*
2. *Sử dụng List Interface*
3. *Sử dụng Set Interface*
4. *Sử dụng Map Interface*
5. *Static methods trong Collections*
6. *Sử dụng Comparable Interface*
7. *Sử dụng Comparator Interface*
8. *So sánh giữa các Collection*
9. *BookStore Version 5*

Chương 07:

Sử dụng Collection trong JAVA

1. *Collections bạn là ai ?*
2. *Sử dụng List Interface*
3. *Sử dụng Set Interface*
4. *Sử dụng Map Interface*
5. *Static methods trong Collections*
6. *Sử dụng Comparable Interface*
7. *Sử dụng Comparator Interface*
8. *So sánh giữa các Collection*
9. *BookStore Version 5*

Collections là gì ?

- *Nhược điểm của kiểu dữ liệu Array*
 - *Kích thước cố định*
 - *Các phần tử phải cùng kiểu dữ liệu và nằm liên kề nhau trong bộ nhớ*
- *Collection khắc phục tất cả các nhược điểm trên ngoài ra còn hỗ trợ sẵn các phương thức: tìm kiếm, sắp xếp, chèn, thao tác, xóa phần tử, ..*
- *Collection là một công cụ có thể chứa và thao tác với các đối tượng thuộc nhiều kiểu dữ liệu khác nhau*

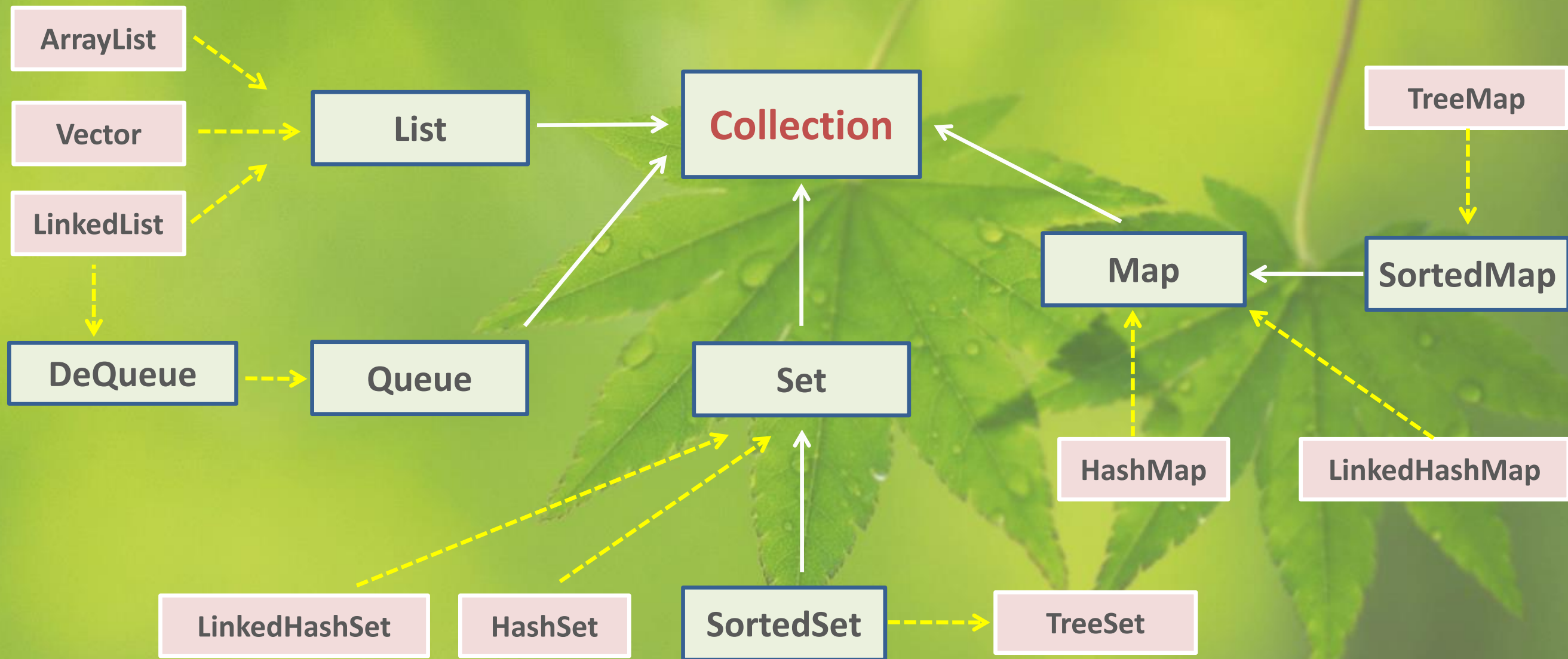
Cấu trúc Collection

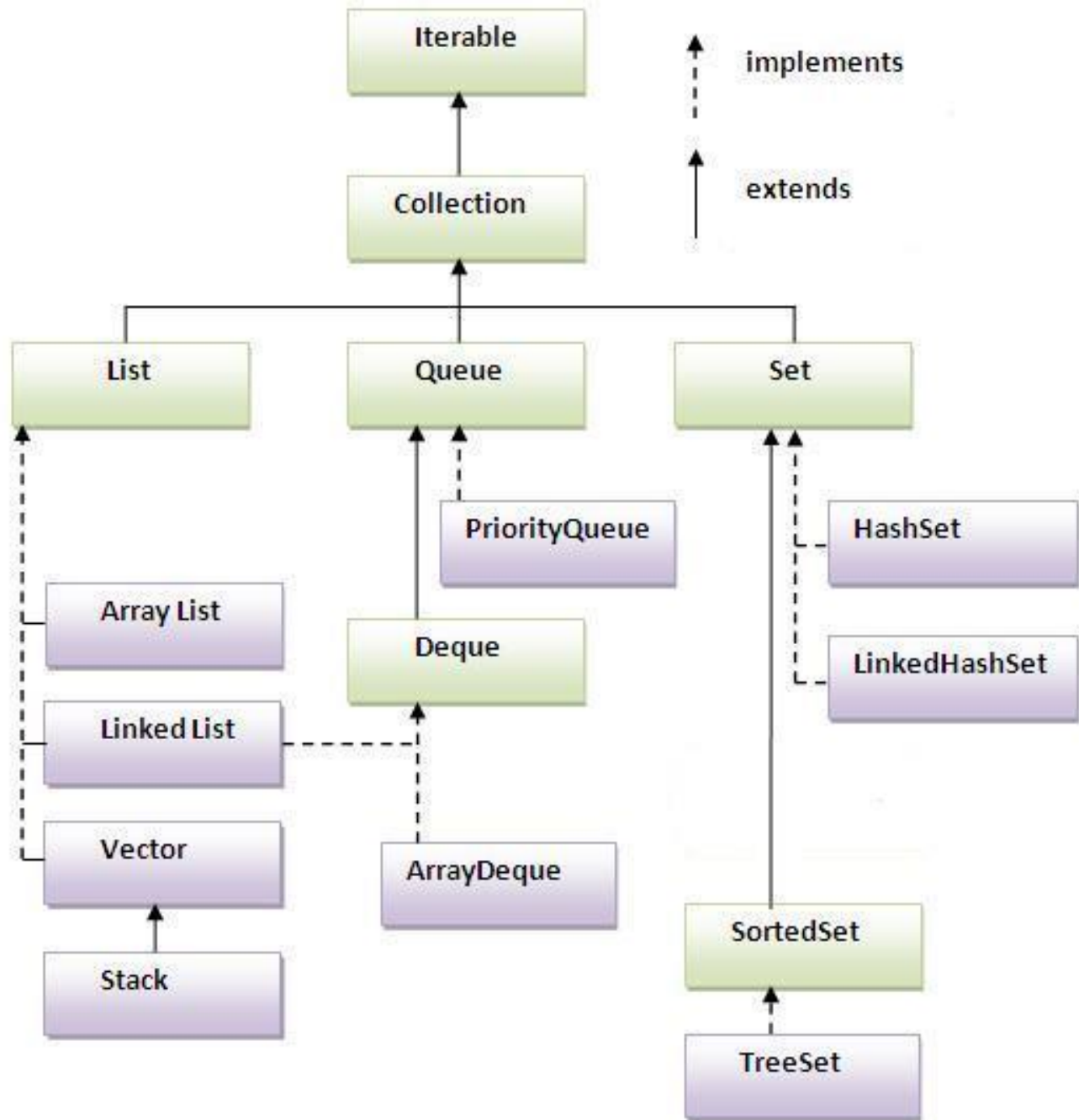
Interface

Class

extends

implements





Cấu trúc Collection

- Tất cả các class và interface của Collection đều nằm trong package *java.util*
- Các phương thức thường sử dụng trong Collection
 - *size()*
 - *remove()*
 - *add()*
 - *set()*
 - *isEmpty()*
 - *get()*

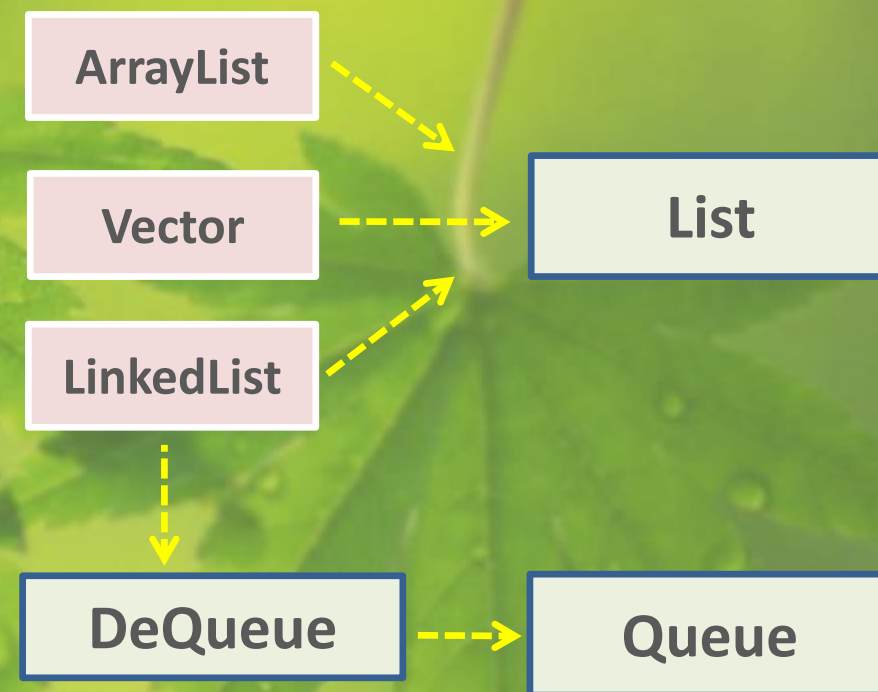
Chương 07:

Sử dụng Collection trong JAVA

1. Collections bạn là ai ?
2. **Sử dụng List Interface**
3. Sử dụng Set Interface
4. Sử dụng Map Interface
5. Static methods trong Collections
6. Sử dụng Comparable Interface
7. Sử dụng Comparator Interface
8. So sánh giữa các Collection
9. BookStore Version 5

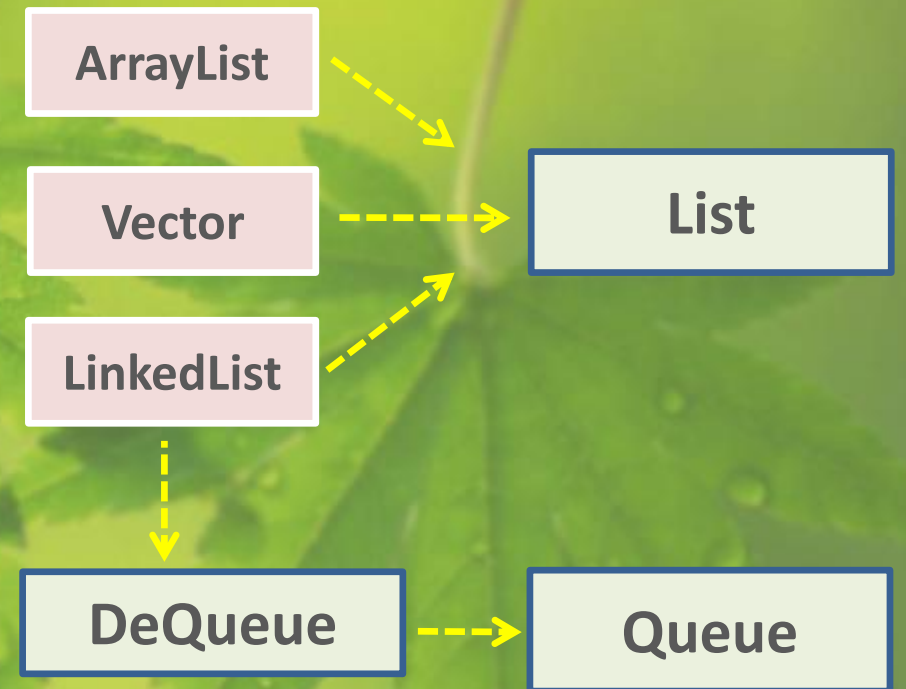
ArrayList

- Sử dụng một mảng động để lưu trữ các phần tử
- Giá trị các phần tử có thể trùng nhau
- Thứ tự thêm vào các element cũng chính là thứ tự các phần tử trong ArrayList
- Tìm hiểu các phương thức thông dụng và phương pháp duyệt các phần tử *for* và *Iterator*



LinkedList

- Sử dụng “**doubly linked list**” để lưu trữ các phần tử
- Giá trị các phần tử có thể trùng nhau
- Thứ tự thêm vào các element cũng chính là thứ tự các phần tử trong *LinkedList*
- Tìm hiểu phương pháp duyệt các phần tử: *for* và *Iterator*



ArrayList vs LinkedList

ArrayList

- *Sử dụng mảng động để lưu trữ các phần tử*
- *Implement: List*
- *Better for storing and accessing data*

LinkedList

- *Sử dụng “doubly linked list” để lưu trữ các phần tử*
- *Implement: List và Dequeue*
- *Better for manipulating*

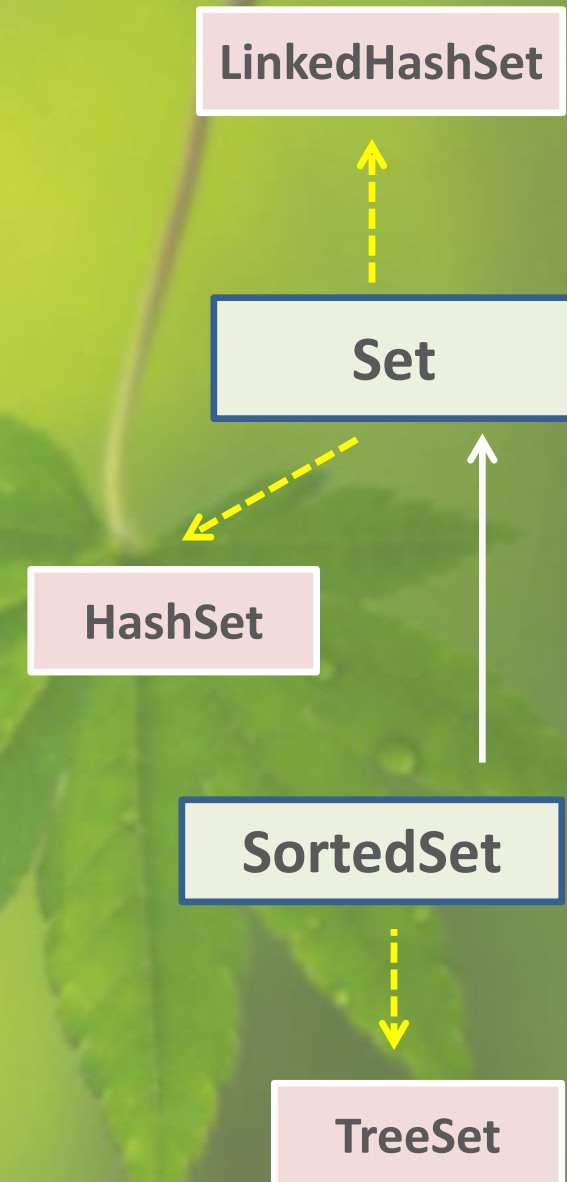
Chương 07:

Sử dụng Collection trong JAVA

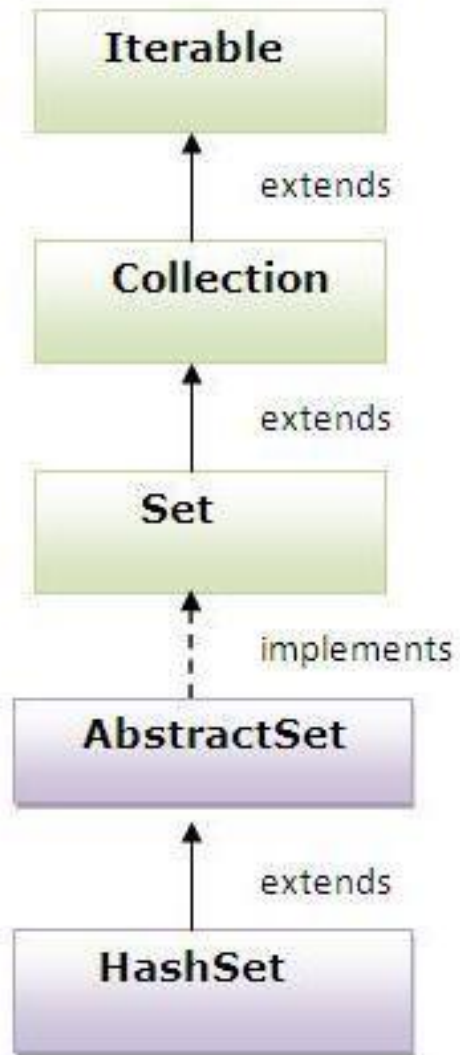
1. Collections bạn là ai ?
2. Sử dụng List Interface
3. **Sử dụng Set Interface**
4. Sử dụng Map Interface
5. Static methods trong Collections
6. Sử dụng Comparable Interface
7. Sử dụng Comparator Interface
8. So sánh giữa các Collection
9. BookStore Version 5

Set Interface

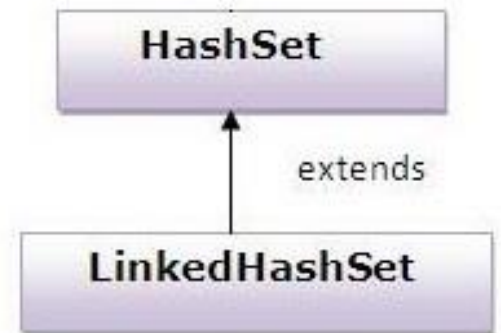
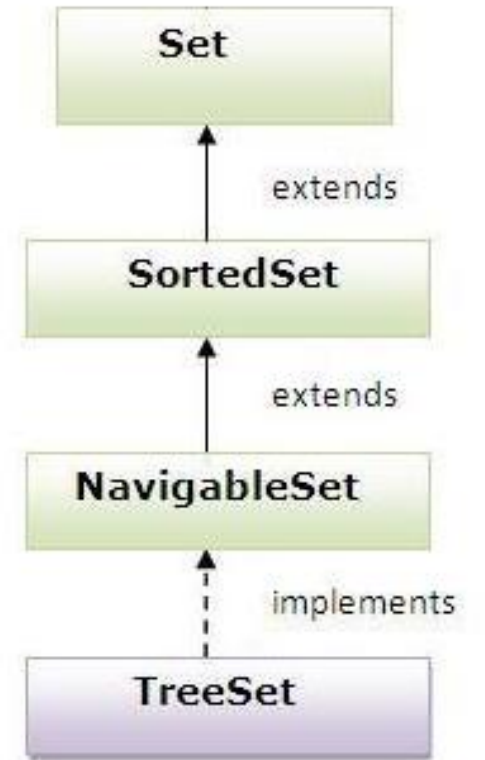
- Lưu trữ các phần tử thuộc nhiều kiểu dữ liệu khác nhau **nhưng** các phần tử **không** được lưu trữ theo chỉ số
- Không chứa các phần tử trùng nhau
- **Sự khác biệt giữa List và Set:** List có thể chứa những phần tử có giá trị giống nhau còn Set thì không
- Các class tiến hành tìm hiểu: HashSet, LinkedHashSet, TreeSet



HashSet – LinkedHashSet – TreeSet



- *HashSet* thứ tự các phần tử được thêm vào bị thay đổi
- *LinkedSet* thứ tự các phần tử được thêm vào đảm bảo được giữ nguyên
- *TreeSet* thứ tự các phần tử được thêm vào được sắp xếp theo thứ tự tăng dần



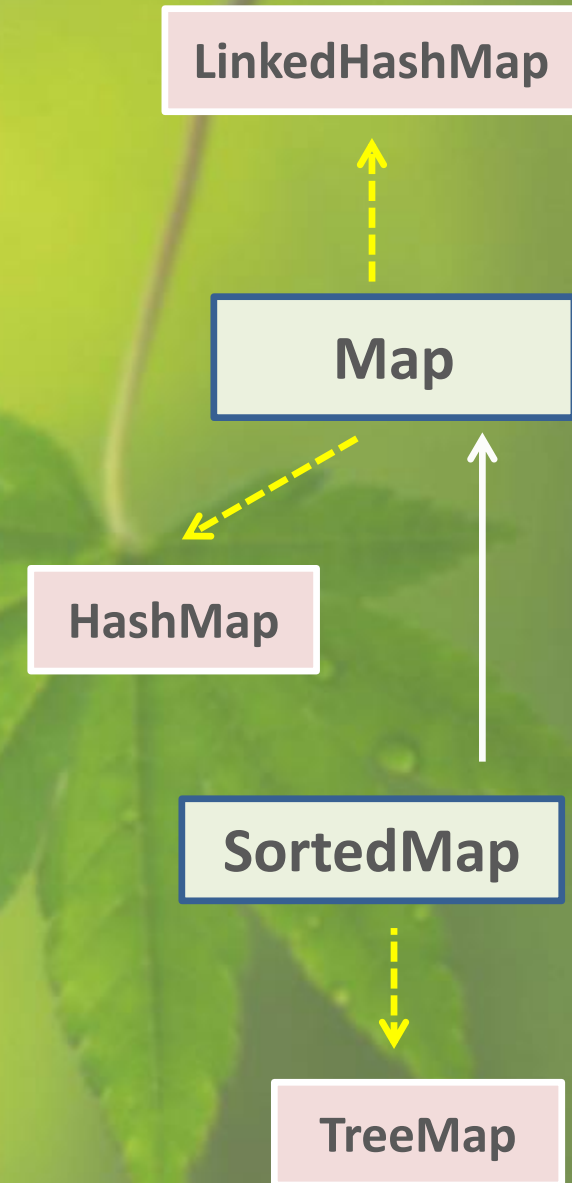
Chương 07:

Sử dụng Collection trong JAVA

1. *Collections bạn là ai ?*
2. *Sử dụng List Interface*
3. *Sử dụng Set Interface*
4. ***Sử dụng Map Interface***
5. *So sánh giữa các Collection*
6. *Static methods trong Collections*
7. *Sử dụng Comparable Interface*
8. *Sử dụng Comparator Interface*
9. *BookStore Version 5*

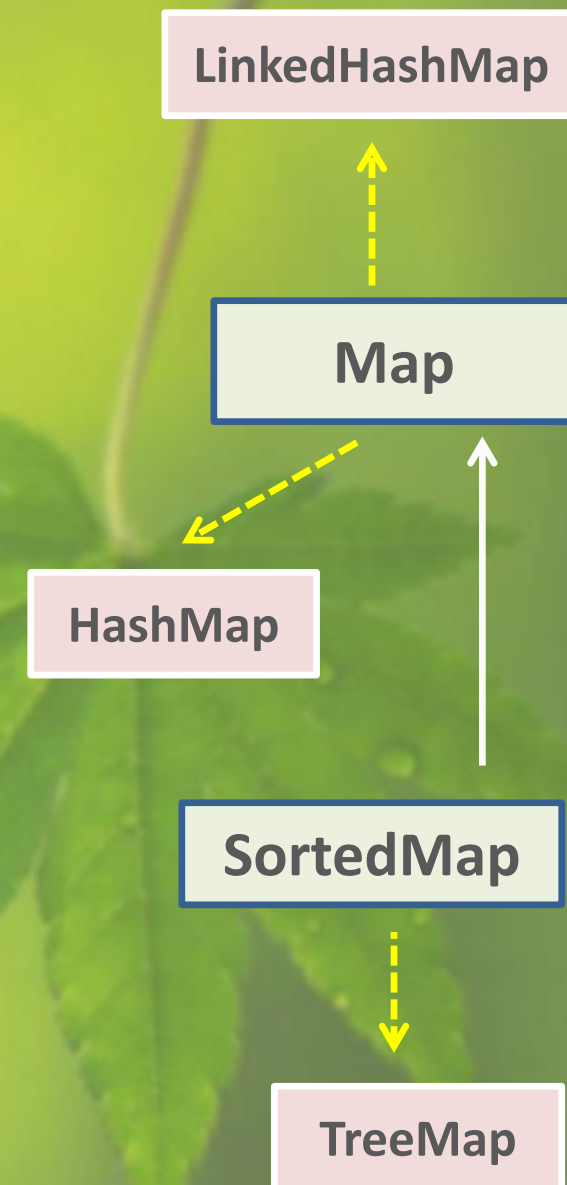
Map Interface

- Lưu trữ các phần tử thuộc nhiều kiểu dữ liệu khác nhau **nhưng** các phần tử là một đối tượng bao gồm **key** và **value**
- Không chứa các phần tử có khóa (key) giống nhau
- Các class tiến hành tìm hiểu: *HashMap*, *LinkedHashMap*, *TreeMap*



HashMap – LinkedHashMap - TreeMap

- *HashMap* thứ tự các phần tử thay đổi so với thứ tự insert vào
- *LinkedHashMap* giữ nguyên thứ tự các phần tử
- *Treemap* thứ tự các phần tử được sắp xếp theo thứ tự tăng dần của key



Chương 07:

Sử dụng Collection trong JAVA

1. *Collections bạn là ai ?*
2. *Sử dụng List Interface*
3. *Sử dụng Set Interface*
4. *Sử dụng Map Interface*
5. ***So sánh giữa các Collection***
6. *Static methods trong Collections*
7. *Sử dụng Comparable Interface*
8. *Sử dụng Comparator Interface*
9. *BookStore Version 5*

So sánh giữa các Collection

- **List** có thể tồn tại các phần tử có giá trị trùng nhau
- **Set** không tồn tại các phần tử có giá trị trùng nhau
- **Map** không tồn tại các phần tử có **key** trùng nhau

Tình huống luyện tập

- *ArrayList* – *LinkedList*
- *HashSet* – *LinkHashSet* – *TreeSet*
- *HashMap* – *LinkHashMap* – *TreeMap*

- Difference between ... and ... in Java
- Java ... vs ...

Chương 07:

Sử dụng Collection trong JAVA

1. *Collections bạn là ai ?*
2. *Sử dụng List Interface*
3. *Sử dụng Set Interface*
4. *Sử dụng Map Interface*
5. *So sánh giữa các Collection*
6. ***Static methods trong Collections***
7. *Sử dụng Comparable Interface*
8. *Sử dụng Comparator Interface*
9. *BookStore Version 5*

Sử dụng Static methods trong Collections

Sử dụng phương thức static của Collections để thực hiện các yêu cầu sau:

- *Sắp xếp danh sách theo thứ tự tăng dần*
- *Đảo ngược vị trí các phần tử trong một danh sách*
- *Sắp xếp danh sách theo thứ tự giảm dần*
- *Tìm kiếm phần tử trong danh sách*
- *Trộn ngẫu nhiên vị trí các phần tử nằm trong danh sách*

Chương 07:

Sử dụng Collection trong JAVA

1. *Collections bạn là ai ?*
2. *Sử dụng List Interface*
3. *Sử dụng Set Interface*
4. *Sử dụng Map Interface*
5. *So sánh giữa các Collection*
6. *Static methods trong Collections*
7. ***Sử dụng Comparable Interface***
8. *Sử dụng Comparator Interface*
9. *BookStore Version 5*

Sử dụng Collections.sort đối với object

- *Exception in thread "main" java.lang.ClassCastException: **chapo6.collection.Course** cannot be cast to java.lang.Comparable*
- Khắc phục bằng cách **implements Comparable** và định nghĩa phương thức **compareTo()**

```
public class Course implements Comparable{  
    @Override  
    public int compareTo(Object obj) {  
        // your code here  
    }  
}
```

Chương 07:

Sử dụng Collection trong JAVA

1. *Collections bạn là ai ?*
2. *Sử dụng List Interface*
3. *Sử dụng Set Interface*
4. *Sử dụng Map Interface*
5. *So sánh giữa các Collection*
6. *Static methods trong Collections*
7. *Sử dụng Comparable Interface*
8. ***Sử dụng Comparator Interface***
9. *BookStore Version 5*

Sử dụng Comparator Interface

- Định nghĩa class *XComparator* implements từ *Comparator*
- Sử dụng tham số thứ hai trong *Collections.sort()*