

Bài 55: Một số interface sắp xếp

- ✓ Interface Comparable
- ✓ Interface Comparator
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Interface Comparable

- Được sử dụng nhằm cho phép hai đối tượng cùng lớp có thể so sánh với nhau theo 1 tiêu chí nào đó
- Thường dùng với mục đích sắp xếp danh sách đối tượng của lớp
- Chỉ có 1 phương thức `int compareTo(T other)`
- Giá trị trả về bởi `compareTo()` có thể là:
 - Số âm nếu giá trị hiện thời nhỏ hơn giá trị của `other`
 - 0 nếu hai giá trị của đối tượng hiện thời và `other` trùng nhau
 - Số dương nếu giá trị hiện thời lớn hơn giá trị của `other`

Interface Comparable

- Nếu muốn sắp xếp tăng dần ta lấy thuộc tính của đối tượng hiện thời so sánh với thuộc tính của other
- Ngược lại ta lấy thuộc tính của other so sánh với thuộc tính của đối tượng hiện thời
- Ta sử dụng phương thức `sort(list)` của Collections để sắp xếp
- Ví dụ sau sắp xếp tên theo thứ tự a-z

Ví dụ

➤ Lớp Movie thực thi interface Comparable:

```
public class Movie implements Comparable<Movie> {  
    private String name;  
    private int year;  
    private String director;  
    private float price;  
    private float star;  
  
    public Movie() {  
    }  
  
    public Movie(String name, int year,  
                  String director, float price, float star) {  
        this.name = name;  
        this.year = year;  
        this.director = director;  
        this.price = price;  
        this.star = star;  
    }  
    //...  
    @Override  
    public int compareTo(Movie o) {  
        return name.compareTo(o.name);  
    }  
}
```

Ví dụ

➤ Lớp Test:

```
public class Test {
    public static void main(String[] args) {
        ArrayList<Movie> movies = new ArrayList<>();
        movies.add(new Movie("Thor", 2011, "ABC", 59, 4.3f));
        movies.add(new Movie("Iron man 3", 2013, "DEF", 79, 4.5f));
        movies.add(new Movie("Ant man", 2015, "MNO", 75, 4.5f));
        movies.add(new Movie("Captain Marvel", 2019, "GHI", 59, 4.7f));
        movies.add(new Movie("Avengers: Infinity War", 2018, "ONL", 50,
4.8f));
        movies.add(new Movie("Avengers: End game", 2019, "XYZ", 99, 4.9f));

        System.out.println("==> Trước khi sắp xếp: ");
        showMovies(movies);
        System.out.println("==> Sau khi sắp xếp theo tên: ");
        Collections.sort(movies);
        showMovies(movies);
    }

    private static void showMovies(ArrayList<Movie> movies) {
        System.out.printf("%-25s%-10s%-10s%-10s%-10s\n",
            "Tên phim", "Năm SX", "Đạo diễn", "giá vé", "Số sao");
        for (var movie : movies) {
            System.out.printf("%-25s%-10d%-10s%-10.2f%-10.2f\n",
                movie.getName(), movie.getYear(), movie.getDirector(),
                movie.getPrice(), movie.getStar());
        }
    }
}
```

Kết quả

➤ Kết quả chạy chương trình:

==> Trước khi sắp xếp:

Tên phim	Năm SX	Đạo diễn	giá vé	Số sao
Thor	2011	ABC	59.00	4.30
Iron man 3	2013	DEF	79.00	4.50
Ant man	2015	MNO	75.00	4.50
Captain Marvel	2019	GHI	59.00	4.70
Avengers: Infinity War	2018	ONL	50.00	4.80
Avengers: End game	2019	XYZ	99.00	4.90

==> Sau khi sắp xếp theo tên:

Tên phim	Năm SX	Đạo diễn	giá vé	Số sao
Ant man	2015	MNO	75.00	4.50
Avengers: End game	2019	XYZ	99.00	4.90
Avengers: Infinity War	2018	ONL	50.00	4.80
Captain Marvel	2019	GHI	59.00	4.70
Iron man 3	2013	DEF	79.00	4.50
Thor	2011	ABC	59.00	4.30

Interface Comparator

- Để thuận tiện cho việc sắp xếp danh sách đối tượng theo nhiều tiêu chí khác nhau ta sử dụng Comparator
- Interface này cho phép ta tạo lớp thực thi độc lập bên ngoài lớp chứa đối tượng cần sắp xếp
- Ta override phương thức `compare(T t1, T t2)` để thiết lập quy chuẩn sắp xếp cho đối tượng trong danh sách của lớp mục tiêu

Interface Comparator

- Phương thức `compare(T t1, T t2)` trả về một số nguyên:
 - Âm nếu giá trị của thuộc tính của `t1` nhỏ hơn của thuộc tính của `t2`
 - 0 nếu hai giá trị thuộc tính của `t1`, `t2` trùng khớp
 - Dương nếu giá trị thuộc tính của `t1` lớn hơn giá trị thuộc tính của `t2`
- Nếu muốn sắp xếp tăng dần, ta so sánh thuộc tính của `t1` với `t2`
- Ngược lại ta so sánh thuộc tính của `t2` với `t1`

Interface Comparator

- Sử dụng phương thức `sort(list, comparator)` để sắp xếp danh sách các đối tượng của kiểu mục tiêu, trong đó:
 - Tham số thứ nhất là danh sách các đối tượng của kiểu mục tiêu cần sắp xếp
 - Tham số thứ hai là đối tượng của lớp đã thực thi interface `Comparator` cho kiểu mục tiêu của các đối tượng trong tham số thứ nhất



Ví dụ

➤ Lớp Movie:

```
public class Movie {  
    private String name;  
    private int year;  
    private String director;  
    private float price;  
    private float star;  
  
    public Movie() {  
    }  
  
    public Movie(String name, int year,  
                  String director, float price, float star) {  
        this.name = name;  
        this.year = year;  
        this.director = director;  
        this.price = price;  
        this.star = star;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

Ví dụ

➤ Lớp thực thi Comparator cho kiểu Movie:

```
/**
 * Lớp này thực thi việc sắp xếp theo tên phim
 */
class SortByName implements Comparator<Movie> {
    @Override
    public int compare(Movie o1, Movie o2) {
        return o1.getName().compareTo(o2.getName());
    }
}

/**
 * Lớp này thực thi việc sắp xếp theo năm sản xuất
 */
class SortByYear implements Comparator<Movie> {
    @Override
    public int compare(Movie o1, Movie o2) {
        return o1.getYear() - o2.getYear();
    }
}
```

Ví dụ

➤ Lớp chạy chương trình:

```
public class Test {
    public static void main(String[] args) {
        ArrayList<Movie> movies = new ArrayList<>();
        movies.add(new Movie("Thor", 2011, "ABC", 59, 4.3f));
        movies.add(new Movie("Iron man 3", 2013, "DEF", 79, 4.5f));
        movies.add(new Movie("Ant man", 2015, "MNO", 75, 4.5f));
        movies.add(new Movie("Captain Marvel", 2019, "GHI", 59, 4.7f));
        movies.add(new Movie("Avengers: Infinity War", 2018, "ONL", 50,
4.8f));
        movies.add(new Movie("Avengers: End game", 2019, "XYZ", 99, 4.9f));

        System.out.println("==> Trước khi sắp xếp: ");
        showMovies(movies);
        System.out.println("==> Sau khi sắp xếp theo tên: ");
        Collections.sort(movies, new SortByName());
        showMovies(movies);
        System.out.println("==> Sau khi sắp xếp theo năm sản xuất: ");
        Collections.sort(movies, new SortByYear());
        showMovies(movies);
    }

    private static void showMovies(ArrayList<Movie> movies) {
        System.out.printf("%-25s%-10s%-10s%-10s%-10s\n",
            "Tên phim", "Năm SX", "Đạo diễn", "giá vé", "Số sao");
        for (var movie : movies) {
            System.out.printf("%-25s%-10d%-10s%-10.2f%-10.2f\n",
                movie.getName(), movie.getYear(), movie.getDirector(),
                movie.getPrice(), movie.getStar());
        }
    }
}
```

Kết quả

➤ Kết quả khi chạy chương trình:

==> Trước khi sắp xếp:

Tên phim	Năm SX	Đạo diễn	giá vé	Số sao
Thor	2011	ABC	59.00	4.30
Iron man 3	2013	DEF	79.00	4.50
Ant man	2015	MNO	75.00	4.50
Captain Marvel	2019	GHI	59.00	4.70
Avengers: Infinity War	2018	ONL	50.00	4.80
Avengers: End game	2019	XYZ	99.00	4.90

==> Sau khi sắp xếp theo tên:

Tên phim	Năm SX	Đạo diễn	giá vé	Số sao
Ant man	2015	MNO	75.00	4.50
Avengers: End game	2019	XYZ	99.00	4.90
Avengers: Infinity War	2018	ONL	50.00	4.80
Captain Marvel	2019	GHI	59.00	4.70
Iron man 3	2013	DEF	79.00	4.50
Thor	2011	ABC	59.00	4.30

==> Sau khi sắp xếp theo năm sản xuất:

Tên phim	Năm SX	Đạo diễn	giá vé	Số sao
Thor	2011	ABC	59.00	4.30
Iron man 3	2013	DEF	79.00	4.50
Ant man	2015	MNO	75.00	4.50
Avengers: Infinity War	2018	ONL	50.00	4.80
Avengers: End game	2019	XYZ	99.00	4.90
Captain Marvel	2019	GHI	59.00	4.70



Minh họa

➤ Thực hiện trong công cụ lập trình

Nội dung tiếp theo

**So sánh giữa abstract class vs
interface**