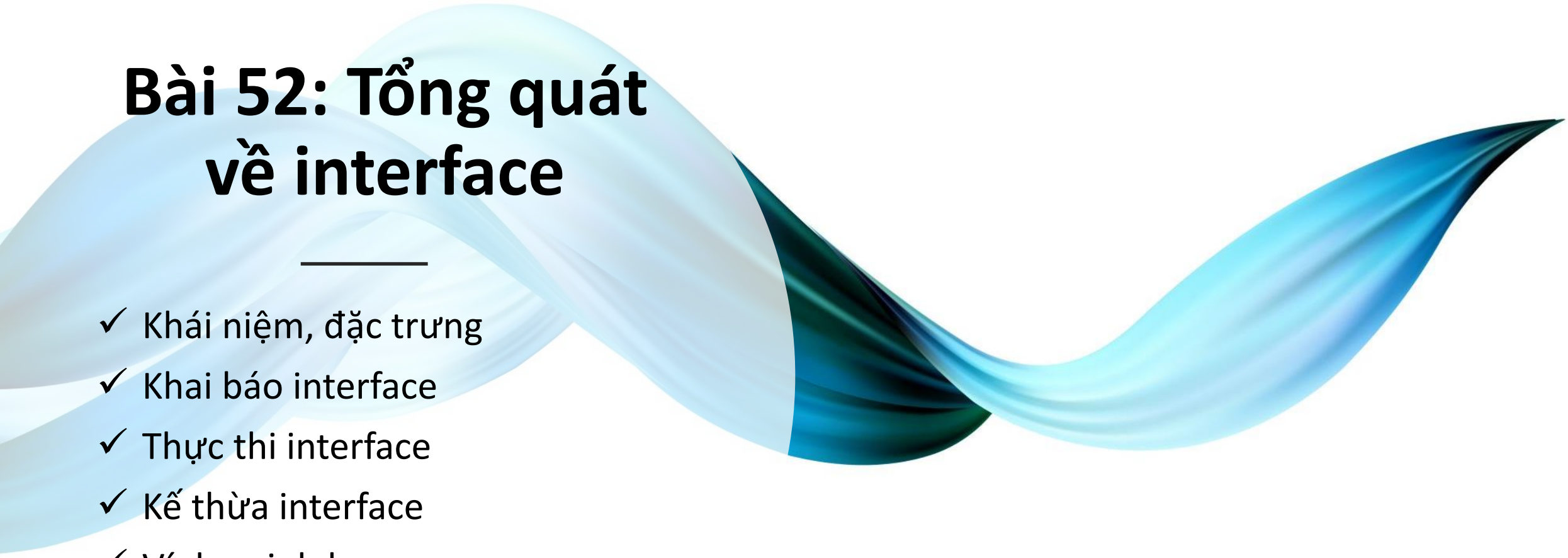
Bài 52: Tổng quát về interface



- ✓ Khái niệm, đặc trưng
- ✓ Khai báo interface
- ✓ Thực thi interface
- ✓ Kế thừa interface
- ✓ Ví dụ minh họa

Khái niệm, đặc trưng

- Interface cùng với class là một kiểu tham chiếu trong Java
- Interface là một kiểu tập hợp của các phương thức abstract và hằng số
- Các thành phần có thể có trong interface gồm: các hằng số, các private method, static methods, private static methods, default methods, abstract methods
- Interface mặc định là abstract nên ta bỏ qua keyword này khi khai báo interface mới

Khái niệm, đặc trưng

- Các trường khai báo trong interface mặc định là public static final(tức hằng số)
- Các phương thức trong interface mặc định là public abstract nên ta có thể bỏ qua hai keyword này trong khai báo abstract methods của interface
- Giống như abstract class, interface không thể khởi tạo đối tượng
- Interface không có constructors
- Một lớp có thể extends chỉ 1 lớp khác nhưng có thể implements nhiều interface

Mục đích sử dụng

- Sử dụng interface để:
  - Đạt được tính trừu tượng hoàn toàn
  - Đạt được mục đích đa kế thừa trong Java
  - Đạt được tính chất kết nối lỏng lẻo(loose coupling) trong Java
  - Để giao tiếp giữa các lớp không cùng trong chuỗi kế thừa nhưng có chung một số hành động nào đó

Khai báo interface

- Sử dụng keyword interface để khai báo mới một interface.
- Cú pháp tổng quát:

```
interface name {
    // các hằng số
    // các phương thức private
    // các phương thức static
    // các phương thức mặc định
    // các phương thức trừu tượng
}
```

Khai báo interface

- Trong đó:
  - interface là keyword bắt buộc phải có
  - Name là tên interface. Tên của interface có thể là danh từ nếu đại diện cho họ các lớp như List, Map
  - Tên interface thường là tính từ viết hoa các chữ cái đầu từ: Clonable, Comparable, Iterable…
  - Dữ liệu của interface là các hằng số
  - Các phương thức private thường là các helper method nội bộ của interface
  - Các phương thức static nhằm chỉ định sẵn một chức năng cụ thể không đổi của interface

Khai báo interface

- Trong đó:
  - Các phương thức default có sẵn thực thi đơn giản ban đầu trong interface và có thể override
  - Các abstract method là thành phần chính của interface để các lớp con implements interface đó override

Ví dụ

- Sau đây là ví dụ một interface của thư viện:

```
public interface MouseListener extends EventListener {
    public void mouseClicked(MouseEvent e);

    public void mousePressed(MouseEvent e);

    public void mouseReleased(MouseEvent e);

    public void mouseEntered(MouseEvent e);

    public void mouseExited(MouseEvent e);
}
```

- Interface tự viết:

```
package lesson52;

interface Drawable {
    void draw(); // vẽ
    void erase(); // tẩy xóa
}
```

Thực thi interface

- Một lớp khi muốn sử dụng tính năng của interface nào thì phải implements interface đó và override các phương thức cần thiết
- Nếu một lớp implements nhiều interface thì các interface đó sẽ phân tách nhau bằng dấu phẩy và đứng sau keyword implements
- Sử dụng tính đa hình của interface như của lớp cha thông thường
- Nếu 1 lớp implements interface thì:
  - Hoặc là thực thi tất cả các phương thức của interface
  - Hoặc thực thi một phần các phương thức của interface và trở thành abstract class

Ví dụ

➢Ví dụ sau thực thi 1 phần:

```
package lesson52;

interface Drawable {
    void draw(); // vẽ

    void erase(); // tẩy xóa
}

abstract class Shape implements Drawable {
    @Override
    public void draw() {
        System.out.println("Drawing a shape...");
    }
}
```

Ví dụ

➢Ví dụ sau thực thi toàn phần interface:

```
interface Drawable {
    void draw(); // vẽ

    void erase(); // tẩy xóa
}

class Shape implements Drawable {
    @Override
    public void draw() {
        System.out.println("Drawing a shape...");
    }

    @Override
    public void erase() {
        System.out.println("Erasing a shape...");
    }
}
```

Kế thừa interface

- Interface có thể kế thừa các interface đã có trước đó qua keyword extends
- Nếu một interface kế thừa nhiều interface khác thì các interface cha sẽ được phân tách nhau bởi dấu phẩy và nằm sau keyword extends

Ví dụ

➢Ví dụ kế thừa nhiều interface:

```
interface Drawable {
    void draw(); // vẽ

    void erase(); // tẩy xóa
}

interface Movable {
    void up(); // di chuyển lên theo trục y

    void down(); // di chuyển xuống theo trục y

    void left(); // di chuyển sang trái theo trục x

    void right(); // di chuyển sang phải theo trục x
}

interface ShapeDrawer extends Drawable, Movable {
    void moveIn(); // di chuyển vào

    void moveOut(); // di chuyển ra

    void moveDiagonally(); // di chuyển theo đường chéo
}
```

Minh họa

- Thực hiện trong công cụ lập trình

Nội dung tiếp theo

Các tính năng mới của interface trong Java 8, 9