

Bài 51: Lớp trừu tượng

- ✓ Khái niệm, mục đích sử dụng
- ✓ Các đặc điểm
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Khái niệm

- Lớp abstract là lớp có chứa keyword abstract trong khai báo lớp
- Phương thức abstract là phương thức có chứa keyword abstract trong khai báo phương thức
- Phương thức abstract chỉ chứa khai báo, không chứa phần thân định nghĩa.
- Ví dụ:

```
// Lớp abstract  
public abstract class Shape {  
    // phương thức abstract  
    public abstract void draw();  
}
```

Mục đích sử dụng

- Abstract methods nêu ra các hành động cần thực hiện nhưng không thực hiện
- Các lớp con kế thừa abstract class sẽ thực hiện điều này
- Lớp abstract dùng để làm lớp cha của một tập các lớp con có chung hành vi nào đó
- Lớp abstract thường chứa các phương thức abstract nhưng không bắt buộc phải chứa phương thức abstract
- Sử dụng lớp abstract nếu muốn đạt được tính trừu tượng: nhìn vấn đề một cách tổng thể thay vì cụ thể

Ví dụ

- Ví dụ về tính trừu tượng:
- Khi sử dụng smartphone người dùng chỉ cần cảm sạc để sạc, dùng vân tay, mặt khẩu, face id để mở khóa màn hình. Chọn ứng dụng và dùng
- Họ không hề hay biết các công đoạn cụ thể từ khi cắm sạc vào máy thì những gì xảy ra phía bên trong
- Họ không biết từ lúc quét vân tay, face id đến khi mở khóa màn hình thì trải qua những gì, thực hiện như thế nào
- Người sử dụng chỉ cần biết cách sử dụng không cần quan tâm cách thức thực thi của một chức năng

Các đặc điểm

- Phương thức abstract chỉ chứa khai báo không chứa định nghĩa
- Phương thức abstract không thể chứa access modifier là private
- Một lớp chứa phương thức abstract bắt buộc phải là lớp abstract nhưng lớp abstract không bắt buộc chứa phương thức abstract
- Lớp abstract không thể dùng để tạo đối tượng nhưng có thể dùng để làm kiểu tham chiếu đến đối tượng của các lớp con

Các đặc điểm

- Lớp abstract có thể chứa tất cả các thành phần đầy đủ của lớp: các trường, phương thức thông thường, các phương thức abstract và các thành phần khác
- Khi lớp con kế thừa từ lớp abstract có 2 tùy chọn:
 - Thực thi các phương thức abstract của lớp cha
 - Trở thành lớp abstract

```
package lesson51;
```

```
public class Shape { // error, thiếu abstract trong khai báo lớp  
    public abstract void draw();  
}
```

Ví dụ

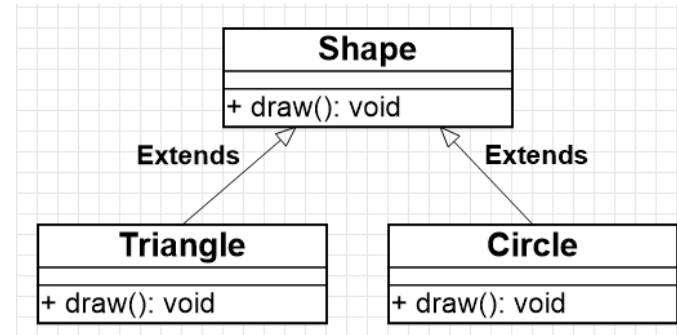
```
public abstract class Person {  
    private String id;        // số chứng minh thư/căn cước/hộ chiếu  
    private String name;     // họ tên  
    private String address;  // địa chỉ  
    private Date dateOfBirth; // ngày sinh  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    // hành động kiếm tiền của người nói chung  
    abstract void earnMoney();  
}  
  
class Student extends Person {  
    private String studentId;  
    private String major;  
    private float avgGrade;  
    //...  
    @Override  
    void earnMoney() {  
        System.out.println("Sinh viên kiếm tiền bằng cách đi làm thêm...");  
    }  
}
```

Ví dụ

```
public class Test {  
    public static void main(String[] args) {  
        Person person = new Person(); // error  
  
        Person nam = new Student(); // ok  
        nam.earnMoney(); // sinh viên Nam kiếm tiền bằng cách của mình  
    }  
}
```


Một ví dụ khác

➤ Lớp Shape và thực thi của nó



```
public abstract class Shape {
    public abstract void draw();
}

class Circle extends Shape {
    @Override
    public void draw() {
        System.out.println("Vẽ hình tròn...");
    }
}

class Triangle extends Shape {
    @Override
    public void draw() {
        System.out.println("Vẽ tam giác...");
    }
}
```



Minh họa

➤ Thực hiện trong công cụ lập trình

Nội dung tiếp theo

Tổng quan về interface