

Bài 32: Các thành phần static

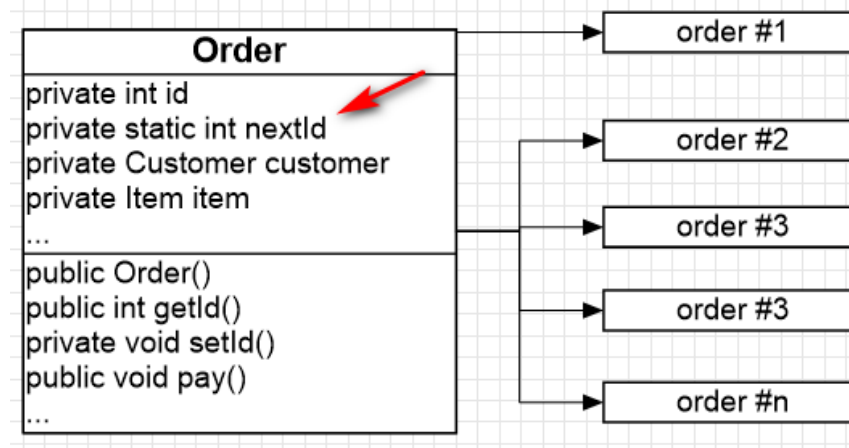
- ✓ Các trường static
- ✓ Các hằng số static
- ✓ Các phương thức static
- ✓ Phương thức main

Các trường static

- Trường static là các thuộc tính của lớp nhưng khai báo với keyword static
- Trường static là thành phần chia sẻ chung cho toàn bộ đối tượng của lớp
- Nếu access modifier của nó là public thì các thành phần bên ngoài lớp có thể truy cập và sử dụng qua tên lớp và tên thành phần static
- Thành phần static thuộc sở hữu lớp, không thuộc sở hữu của từng đối tượng riêng lẻ
- Khi tạo đối tượng của lớp, các đối tượng sẽ nhận một bản copy độc lập của các thuộc tính không static nhưng lại chia sẻ chung thành phần static

Ví dụ

- Trong hệ thống thương mại điện tử có lớp đơn hàng (Order). Ta giả định rằng mã đơn hàng là số nguyên tự tăng. Mỗi khi có khách hàng click đặt hàng thì đơn hàng sẽ được tạo và mỗi đơn sẽ có một mã mới là số kế tiếp của đơn trước đó. Để thực hiện điều này ta dùng một nextId static:



Các hằng số static

- Các trường static tương đối hiếm khi được sử dụng nhưng các hằng static lại thường xuyên xuất hiện
- Ví dụ điển hình là các hằng số của lớp Math như PI, E chẳng hạn
- Khi sử dụng các hằng static ta không cần đối tượng của lớp chứa nó mà chỉ cần cung cấp tên lớp và tên hằng cần sử dụng
- Ví dụ khác là hằng số **out** hay **in** của lớp System ta đã sử dụng rất nhiều lần từ bài đầu tiên đến nay
- Hằng số thường là thành phần dùng chung nên nếu muốn các lớp khác có thể sử dụng ta để access modifier của nó là public

Các phương thức static

- Phương thức static là phương thức không vận hành trên đối tượng
- Nếu sử dụng trong nội bộ lớp ta chỉ cần cung cấp tên phương thức và các đối số tương ứng
- Nếu sử dụng phương thức static của lớp khác, ta cung cấp tên lớp, tên phương thức và các đối số
- Các thành phần static nói chung không thể sử dụng được với từ khóa this
- Các phương thức static thường chỉ vận hành cùng các thành phần static
- Không dùng phương thức static để thao tác với các thành phần thuộc tính/phương thức non-static của lớp

Các phương thức static

- Ta sử dụng phương thức static trong hai trường hợp chính:
 - Khi phương thức không cần truy cập đến trạng thái của đối tượng vì tất cả những gì cần thiết đã được cung cấp qua các đối số của lời gọi phương thức
 - Khi một phương thức chỉ cần thao tác với các thành phần static của lớp

Ví dụ

```
public class StaticExample {  
    public static void main(String[] args) {  
        doSomeStatic(); // ok  
        doSomething(); // error  
    }  
  
    // phương thức vận hành không cần object  
    public static void doSomeStatic() {  
        System.out.println("Bên trong doSomeStatic()");  
    }  
  
    // phương thức vận hành trên object  
    public void doSomething(){  
        System.out.println("Bên trong doSomething()");  
    }  
}
```

Phương thức main

- Mọi chương trình Java muốn chạy được đều phải có phương thức main
- Phương thức này không vận hành dựa trên đối tượng
- Khi chương trình bắt đầu chạy không có đối tượng nào cả. Khi vào main, các đối tượng cần thiết khác mới bắt đầu được tạo ra
- Mỗi lớp đều có thể có một phương thức main. Điều này rất hữu ích cho việc thực hiện unit testing-kiểm thử đơn vị

Ví dụ

➤ Hình ảnh quen thuộc của phương thức main:

```
public static void main(String[] args) {  
    Order order1 = new Order(); // tạo đơn hàng thứ nhất  
    Order order2 = new Order(); // tạo đơn hàng thứ hai  
    // .. tạo đơn hàng thứ n...  
    System.out.println("order1 id = " + order1.getId());  
    System.out.println("order2 id = " + order2.getId());  
}
```

Nội dung tiếp theo

Inner class và nested class