

Bài 31: Chuẩn thiết kế lớp

- ✓ Các chuẩn thiết kế lớp
- ✓ Minh họa

Chuẩn thiết kế lớp

- Luôn để dữ liệu private
- Luôn khởi tạo dữ liệu
- Không sử dụng quá nhiều kiểu nguyên thủy trong 1 lớp
- Không phải tất cả thuộc tính đều cần getter/setter
- Chia nhỏ lớp nếu nó thực hiện quá nhiều nhiệm vụ
- Đặt tên các thành phần sao cho tự bản thân nó mô tả ý nghĩa

Chuẩn 1

- Luôn để dữ liệu private. Đây là chuẩn đầu tiên và quan trọng nhất để đảm bảo tính đóng gói dữ liệu
- Dữ liệu có thể thường xuyên thay đổi nhưng cách mà nó được sử dụng thường ít thay đổi hơn
- Ta thường phân tách ra thành hai phần rõ ràng: dữ liệu private và để truy cập dữ liệu này ta phải thông qua phương thức get/set tương ứng
- Điều này vừa đảm bảo dữ liệu được an toàn vừa giảm thiểu các lỗi do trực tiếp thao tác với dữ liệu gây ra

Ví dụ

```
public class Pet {  
    private String name; // tên  
    private float age; // tuổi  
  
    // các getter/setter:  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public float getAge() {  
        return age;  
    }  
  
    public void setAge(float age) {  
        this.age = age;  
    }  
}
```

Chuẩn 2

- Luôn chủ động khởi tạo dữ liệu
- Java không chủ động khởi tạo giá trị cho các biến cục bộ. Do vậy bạn phải chủ động thực hiện.
- Với các thuộc tính, Java sẽ gán giá trị mặc định nhưng bạn không nên lệ thuộc vào điều này mà nên chủ động khởi tạo trong các constructors
- Điều này tạo thói quen tốt trong lập trình
- Và tránh được các lỗi không đáng có, ví dụ như NullPointerException

Ví dụ

➤ Ví dụ: chủ động khởi tạo giá trị

```
public class Pet {  
    private String name; // tên  
    private float age; // tuổi  
    private float weight; // cân nặng  
    // khởi tạo giá trị cho các thuộc tính trong constructor:  
    public Pet() {  
        age = 0;  
        name = "";  
        weight = 0;  
    }  
}
```

Chuẩn 3

- Không sử dụng quá nhiều kiểu nguyên thủy trong một lớp
- Ta thường gom nhóm các thuộc tính liên quan đến nhau vào một lớp nhằm đơn giản hóa vấn đề.
- Ví dụ:

```
public class Customer {  
    // thông tin về họ, đệm tên có thể thay thế không?  
    private String firstName;  
    private String lastName;  
    private String midName;  
  
    private String email;  
    private int age;  
    private String phoneNumber;  
    //...  
}
```

Chuẩn 3

➤ Chuyển thành:

```
public class Customer {  
    private FullName fullName;  
    private String email;  
    private int age;  
    private String phoneNumber;  
    //...  
}  
  
// tách Lớp FullName để lưu trữ các thông tin về tên  
class FullName {  
    private String firstName;  
    private String lastName;  
    private String midName;  
    //...  
}
```


Chuẩn 4

- Không phải tất cả các thuộc tính đều có getter/setter
- Ví dụ thông tin ngày bắt đầu làm việc tại công ty chẳng hạn. Kể từ sau khi thông tin đó được xác nhận, ta không cần thay đổi nó. Vậy không cần setter.
- Ví dụ thông tin về số điện thoại chẳng hạn, nếu nó là nhạy cảm và public với công chúng ta sẽ không để getter. Vậy sẽ không ai lấy được.

Chuẩn 5

- Chia nhỏ lớp đang thực hiện quá nhiều nhiệm vụ
- Một lớp mà chứa quá nhiều thông tin và vượt ngoài phạm vi ý nghĩa của lớp thì nên tách lớp
- Chia nhỏ lớp sao cho mỗi lớp chỉ thực hiện đúng nhiệm vụ mà nó mô tả mà thôi
- Không nên lạm dụng việc chia nhỏ lớp đến mức mỗi lớp chỉ có duy nhất một thuộc tính hay phương thức

Chuẩn 5

- Ví dụ một lớp mô tả thông tin sinh viên nhưng lại có cả thông tin môn học, bảng điểm, lớp học và các phương thức tính toán điểm trung bình, xét học lực, xét học bổng, thêm sửa môn học, khóa học... thì ta nên xem xét tách lớp ra cho tiện quản lý.

Chuẩn 6

- Đặt tên các thành phần sao cho tự nó mô tả ý nghĩa
- Phần này đã trình bày chi tiết trong từng bài học liên quan đến tên biến, tên hằng, tên phương thức, tên lớp...
- Với tên thuộc tính dùng danh từ, cụm danh từ hoặc tính từ kết hợp danh từ
- Với tên lớp cũng dùng danh từ, cụm danh từ, tính từ kết hợp với danh từ như Student, GraduatedStudent, BillingAddress, LearningProcess

Chuẩn 6

- Với tên phương thức thì dùng động từ hoặc cụm động từ trừ các lớp quá chuyên biệt như Math
- Với các getter và setter luôn có dạng getX hoặc setX với X là tên thuộc tính cần thao tác với
- Khi tuân thủ các chuẩn này bạn bắt đầu trở nên chuyên nghiệp và code chất lượng hơn



Ví dụ

- Ví dụ minh họa việc tách lớp khi một lớp thực hiện quá nhiều nhiệm vụ

Nội dung tiếp theo

Các thành phần static