

BÁO CÁO LAB 02

Question: If you create a constructor method to build a **DVD** by title then create a constructor method to build a **DVD** by category. Does JAVA allow you to do this?

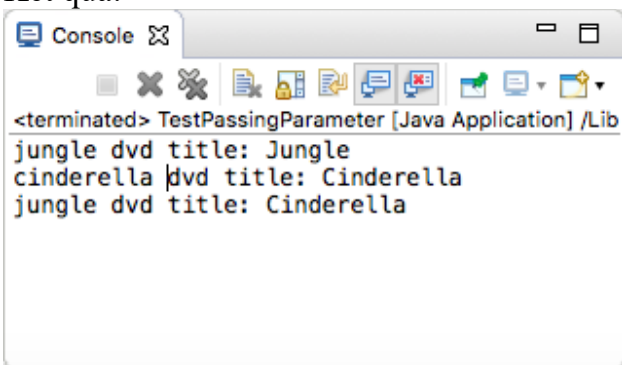
- Có, Java cho phép bạn tạo nhiều constructor với các tham số khác nhau trong cùng một lớp. Đây được gọi là constructor overloading (quá tải constructor).

Question: *Is JAVA a Pass by Value or a Pass by Reference programming language?*

Code:

```
public class TestPassingParameter {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");  
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");  
  
        swap(jungleDVD, cinderellaDVD);  
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());  
        System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());  
  
        changeTitle(jungleDVD, cinderellaDVD.getTitle());  
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());  
    }  
  
    public static void swap(Object o1, Object o2) {  
        Object tmp = o1;  
        o1 = o2;  
        o2 = tmp;  
    }  
  
    public static void changeTitle(DigitalVideoDisc dvd, String title) {  
        String oldTitle = dvd.getTitle();  
        dvd.setTitle(title);  
        dvd = new DigitalVideoDisc(oldTitle);  
    }  
}
```

Kết quả:



1. Sau khi gọi swap(jungleDVD, cinderellaDVD), tại sao tiêu đề của hai đối tượng vẫn không đổi?

- Trong phương thức swap(Object o1, Object o2), hai tham số o1 và o2 chỉ là bản sao của tham chiếu gốc.
- Khi thực hiện phép gán o1 = o2; và o2 = tmp;, nó chỉ thay đổi giá trị của tham số trong phạm vi phương thức.
- Tuy nhiên, tham chiếu gốc bên ngoài phương thức không bị thay đổi.
- Do đó, sau khi gọi swap(jungleDVD, cinderellaDVD), hai đối tượng gốc vẫn giữ nguyên tiêu đề của chúng.

```
void swap(Object o1, Object o2) {  
    Object tmp = o1;  
    o1 = o2; // Chỉ thay đổi giá trị của o1 trong phương thức
```

```
    o2 = tmp; // Chỉ thay đổi giá trị của o2 trong phương thức  
}
```

2. Sau khi gọi `changeTitle(jungleDVD, cinderellaDVD.getTitle())`, tại sao tiêu đề của `jungleDVD` lại thay đổi?

- Khi gọi `changeTitle(jungleDVD, cinderellaDVD.getTitle())`, tham chiếu `jungleDVD` được truyền vào phương thức.
- Trong `changeTitle()`, dòng `dvd.setTitle(title);` thay đổi nội dung bên trong đối tượng `jungleDVD` vì nó truy cập trực tiếp vào vùng nhớ của đối tượng này.
- Mặc dù `dvd = new DigitalVideoDisc(oldTitle);` tạo một đối tượng mới, nhưng nó chỉ thay đổi tham chiếu của biến `dvd` trong phương thức (không ảnh hưởng đến `jungleDVD` bên ngoài `main()`).
- Vì thế, `jungleDVD` ngoài `main()` vẫn trỏ đến cùng đối tượng, nhưng tiêu đề đã bị thay đổi.

```
void changeTitle(DigitalVideoDisc dvd, String title) {  
    dvd.setTitle(title); // Thay đổi tiêu đề của đối tượng gốc  
    dvd = new DigitalVideoDisc("Old Title"); // dvd trỏ sang đối tượng mới nhưng  
    không ảnh hưởng đến jungleDVD bên ngoài  
}
```

Kết luận:

- Java truyền tham trị (pass by value), không phải pass by reference.
- Khi truyền một đối tượng, Java truyền bản sao của tham chiếu đến đối tượng (tức là pass-by-value của reference).
- `swap()` không đổi giá trị vì thay đổi tham chiếu chỉ có tác dụng trong phương thức.
- `changeTitle()` thay đổi nội dung của đối tượng gốc vì nó tác động lên vùng nhớ của đối tượng đó.