CÂU HỎI PHỎNG VẤN LẬP TRÌNH JAVA

Phân biệt tham chiếu và tham trị trong java

IT Phú Trần 16 Tháng Ba 2017

Phân biệt tham chiếu và tham trị trong java tham chiếu là gì tham trị là gì ví dụ về tham chiếu và tham trị trong java

Tham chiếu và tham trị trong java rất dễ gây nhầm lẫn cho các bạn mới học,kể cả các lập trình viên code lâu năm cũng đôi lúc có sự nhầm lẫn về sự phân biệt khái niệm tham chiếu và tham trị trong java.Trong bài này,tôi sẽ chia sẻ theo kinh nghiệm cũng như hiểu biết của mình để các bạn tham khảo cùng thảo luận về vấn đề trên.

Trong JAVA có 2 loại kiểu dữ liệu: cơ bản và tham chiếu :
– Kiểu cơ bản : Số nguyên, số thực, kí tự , logic
– Kiểu tham chiếu : Kiểu mảng, kiểu lớp đối tượng Class, kiểu Interface

Truyền tham trị là gì?

ĐÔI NÉT VỀ TÔI?

Tôi là Trần Phú

Hiện đang là Senior tại công ty 인조이웍스: EnjoyWorks và Vinaenter.

Tôi có 6 năm kinh nghiệm chuyên về lập trình web, trong đó có 5 năm kinh nghiệm làm việc với Java. Tôi yêu thích và muốn tìm hiểu chuyên sâu về java.

Đọc thêm về tôi

Khi các bạn truyền tham trị thì trình biên dịch sẽ tạo 1 bản sao và sau đó truyền bản sao đó vào nơi muốn truyền, vì vậy mà mọi thay đổi sẽ sảy ra trên bản sao này và ko ảnh hưởng đến biến gốc.

Truyền tham chiếu là gì?

Truyền tham chiếu thì các bạn sẽ truyền ngay địa chỉ hay gọi là chính cái biến đó vào nơi cần truyền, nên ở trong hàm mà mình truyền vào có thay đổi gì thì biến ngoài cũng thay đổi theo.

Để làm rõ hơn về vấn đề này,tôi sẽ vào một ví dụ để các bạn dễ hình dùng như sau :

1.Ví dụ về truyền kiểu dữ liệu nguyên thủy :

Class **TestDemo1.java :**

```
1  package demo;
2
3  public class TestDemo1 {
4      private static int number;
5
6      public void showNumber(){
7          System.out.println(number);
8      }
9
10     public void changeNumber(int number){
11         number = 4;
12     }
13
14     public static void main(String[] args) {
15         TestDemo1 demo1 = new TestDemo1();
16         demo1.number  = 10;
17         demo1.changeNumber(number);
18         demo1.showNumber();
19     }
20 }
```

Khi bạn truyền tham số cho hàm trong Java, Java sẽ tạo ra một bản sao của biến và truyền vào phương thức đó. Vậy nên khi thực thi hàm chỉ có bản sao bị ảnh hưởng mà biến không hề bị ảnh hưởng.Vì lý do vậy nên khi chạy trường trình thì giá trị của biến **number** vẫn bằng 10.

2. Java không có kiểu truyền tham chiếu (reference)

Thường thì chúng ta sẽ nghĩ rằng ~~**Kiểu dữ liệu cơ sở được truyền theo tham trị, kiểu object được truyền theo tham chiếu**~~. Ok.Nhưng đó là một sự nhầm lẫn.

Khi các bạn truyền một biến có kiểu object vào một hàm thì lúc đó có nghĩa ta đã **truyền giá trị** của biến đó để sử dụng trong hàm, chứ không phải truyền đối tượng được biến đó tham chiếu tới. Vậy nên việc thay đổi giá trị của biến có kiểu object này bằng cách gán cho một biến kiểu object khác thì object được tham chiếu đến lúc đầu không bị ảnh hưởng, chỉ khi sử dụng các method của chính các object được tham chiếu này thì dữ liệu của object mới được thay đổi sau khi ra khỏi hàm.

Các bạn hãy xem ví dụ dưới đây :

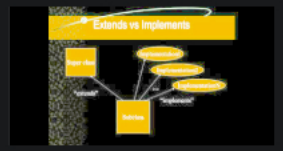
```
1  package demo;
2
3  import bean.SinhVien;
4
5  public class TestDemo2 {
6      public static void main(String[] args) {
7          TestDemo2 demo2 = new TestDemo2();
8          // tạo đối tượng sinh viên
9          SinhVien sinhVien = new SinhVien("Lan Anh");
10         // lấy tên trước khi gọi phương thức
```

```
        System.out.println("Trước khi gọi method change1: "+sinhVien.getName());
        demo2.change1(sinhVien);
        System.out.println("Sau khi gọi method change1 : "+sinhVien.getName());

        System.out.println("Trước khi gọi method change2: "+sinhVien.getName());
        demo2.change2(sinhVien);
        System.out.println("Sau khi gọi method change2: "+sinhVien.getName());
    }

    public void change1(SinhVien sinhVien){
        SinhVien objSV = new SinhVien();
        sinhVien = objSV;
    }

    public void change2(SinhVien sinhVien){
        sinhVien.setName("Phú Trần IT");
    }
}
```

Qua ví dụ này,các bạn thấy được rằng với Java không có tryền tham chiếu như thấy ở trường hợp 1 sử dụng phương thức change1,chúng ta vẫn lấy được name là "Lan Anh" .Giá trị của biến kiểu object là địa chỉ của object mà biến đó tham chiếu đến đến. Cho nên lúc này việc thay đổi giá trị của biến kiểu object này bằng cách gán cho một biến kiểu object khác thì object được tham chiếu đến lúc đầu không bị ảnh hưởng.

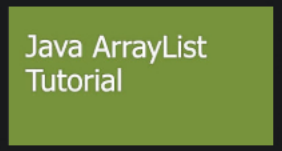
Khi dùng change2 thì lúc này,khi biến tham chiếu đến chúng ta đã thay đổi giá trị ở trong phương thức bằng cách setName lại mà không thông qua một đối tượng tham chiếu tạo ra mới nào,nên lúc này chỉ khi sử dụng các method của chính các object được tham chiếu này thì dữ liệu của object mới được thay đổi sau khi ra khỏi hàm.Vậy kết quả lần lượt sẽ là : "Lan Anh" , "Phú Trần IT".

Các bài viết cùng chủ đề:



Phân biệt extends và implements trong java

Hướng dẫn tạo menu trong Java

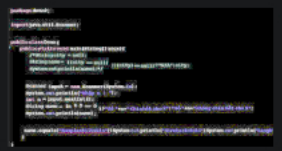


Ví dụ và cách sử dụng ArrayList trong Java

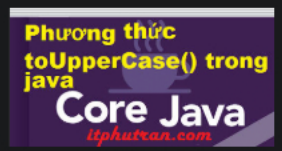
Xử lý trôi lệnh khi dùng Scanner trong Java bằng nhiều cách

Toàn tập kế thừa (extends) trong Java

Nhập xuất dữ liệu sử dụng Scanner trong Java



Cấu trúc if else ? : trong Java



Phương thức toUpperCase() trong java