Bài Học

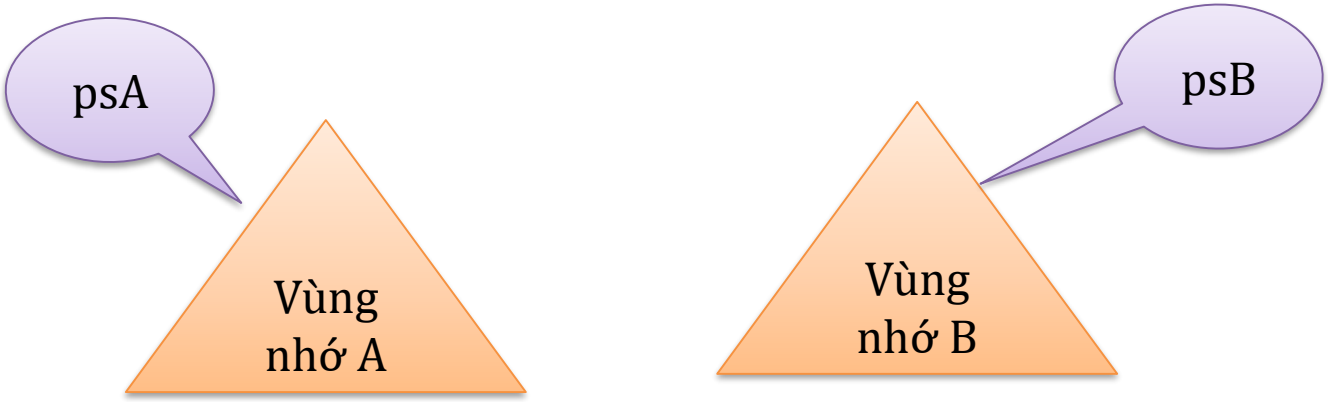
Alias
và
cơ chế gom rác tự động

Alias và cơ chế gom rác tự động

- Alias là đặc điểm mà trên một ô nhớ có nhiều biến đối tượng cùng trỏ tới.
- Ví dụ:

```
PhanSo psA=new PhanSo(1,5);
PhanSo psB=new PhanSo(3,7);
```

Lúc này trên thanh RAM sẽ có 2 ô nhớ cấp phát cho 2 đối tượng phân số được quản lý bởi 2 biến đối tượng psA và psB

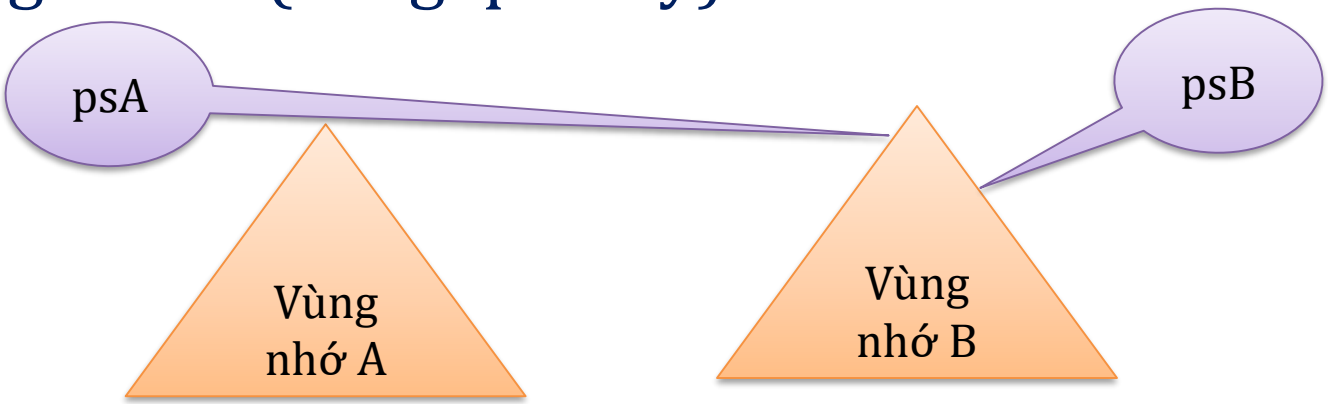


Alias và cơ chế gom rác tự động

- Giả sử ta thực hiện lệnh:

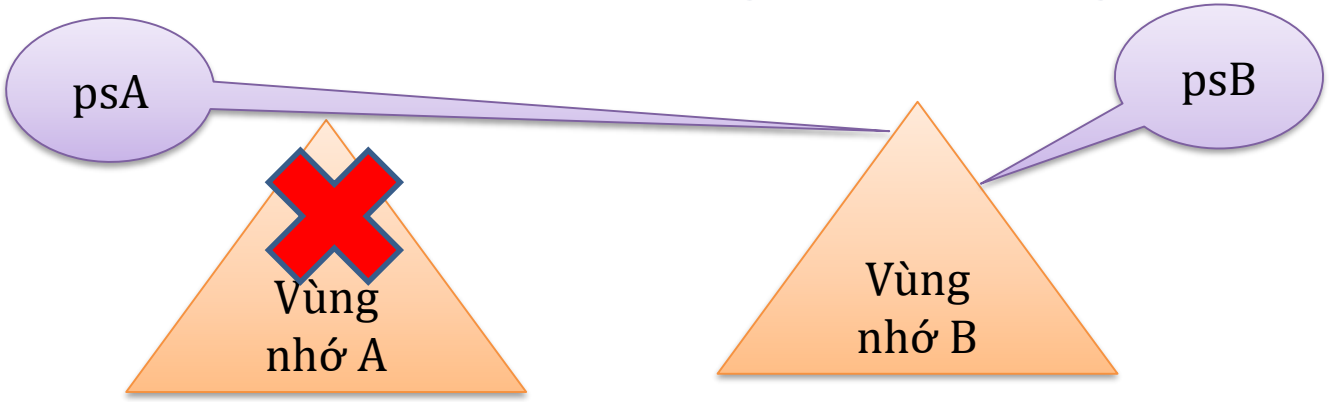
psA=psB;

➔ Ngôn ngữ nói "Phân số A bằng Phân số B", nhưng hệ thống máy tính sẽ làm việc theo cơ chế "Phân số A trỏ tới vùng nhớ mà phân số B đang quản lý". Hay nói cách khác "Vùng nhớ B" bây giờ có 2 biến đối tượng cùng trỏ tới(cùng quản lý)



Alias và cơ chế gom rác tự động

- Như vậy đã xuất hiện Alias ở "vùng nhớ B". Lúc này sẽ xảy ra 2 hiện tượng như sau:
  - Tại "vùng nhớ B", nếu psA thay đổi thông tin sẽ làm cho psB thay đổi thông tin (vì cả 2 đối tượng này cùng quản lý một vùng nhớ)
  - "Vùng nhớ A" không còn đối tượng nào tham chiếu tới, lúc này hệ thống sẽ tự động thu hồi bộ nhớ (hủy vùng nhớ A đã cấp trước đó), cơ chế này gọi là cơ chế gom rác tự động



Alias và cơ chế gom rác tự động

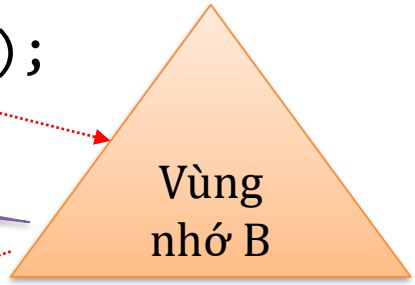
- Đôi khi trong quá trình thực hiện phần mềm ta có nhu cầu sao chép đối tượng ra (tạo thêm một đối tượng giống y xì đối tượng cũ nhưng nằm ở ô nhớ khác, để ta có thể tự do thay đổi thông tin trên đối tượng sao chép mà không làm ảnh hưởng tới đối tượng gốc). Java hỗ trợ chúng ta hàm clone trong interface Cloneable để sao chép đối tượng.

```
public class PhanSo implements Cloneable {
 public PhanSo copy()
       throws CloneNotSupportedException
 {
     return (PhanSo) this.clone();
 }
}
```

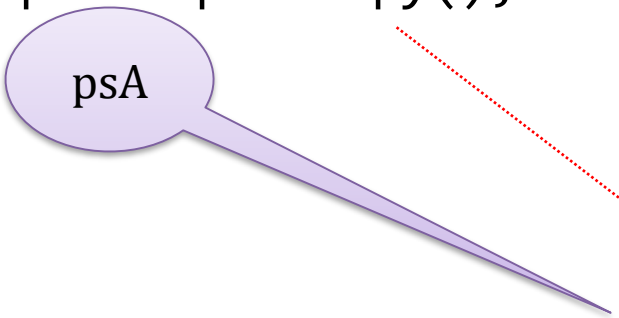
Alias và cơ chế gom rác tự động

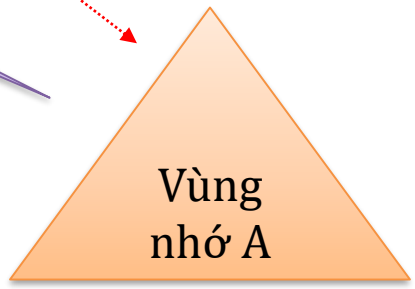
Ví dụ:

```
PhanSo psB = new PhanSo(1,4);
```



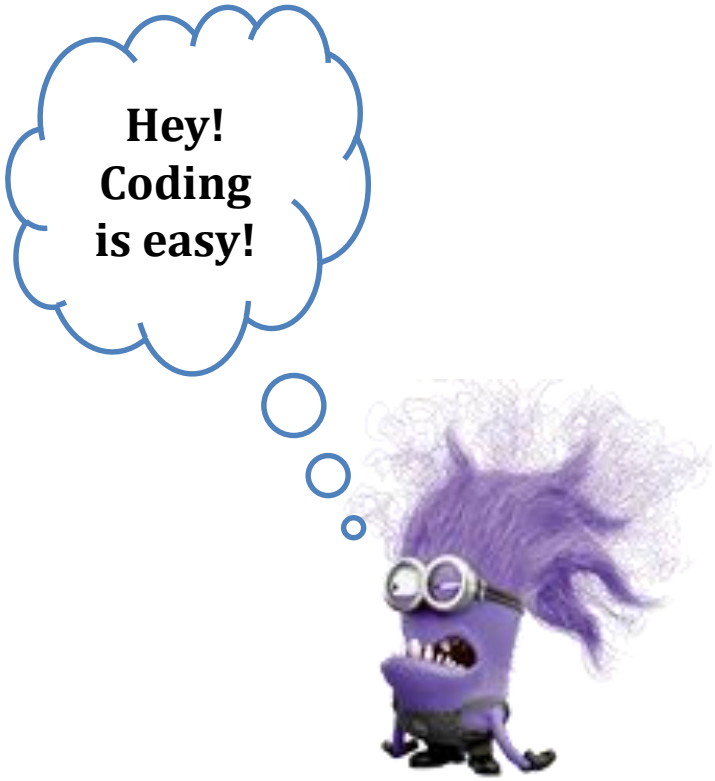

```
PhanSo psA = psB.copy();
```





Sao chép toàn bộ thông tin trong Vùng nhớ B vào vùng nhớ A ➔Tức là ta có 2 đối tượng có thông tin giống nhau y xì nhưng nằm trên 2 ô nhớ hoàn toàn khác nhau

psA thay đổi không ảnh hưởng gì tới psB và ngược lại



END