* **Comparable** (interface)
  + Để so sánh chỉ 1 option
  + 1 class chỉ implement 1 comparable
  + Chỉ chứa method **CompareTo()**
  + Không tách được các tiêu chí so sánh ra
* **CompareTo(Object obj)** 
  + Dùng để so sánh đối tượng hiện tại với 1 đối tượng khác, sắp xếp list đối tượng
  + Phải implement ở đối tượng cần so sánh
  + Note: Check Object = null
  + Using: Collections.sort(list);
* **Comparator** (interface)
  + Định nghĩa so sánh multi-option
  + Chứa 2 method **Compare()** và **equals()**
* **Compare(Object obj1, Object obj2)**
  + - Compare() định nghĩa chính xác cách sắp xếp, so sánh giữa nhiều đối tượng
    - Không nhất thiết phải implement ở đối tượng cần so sánh, có thể tách ra inner class

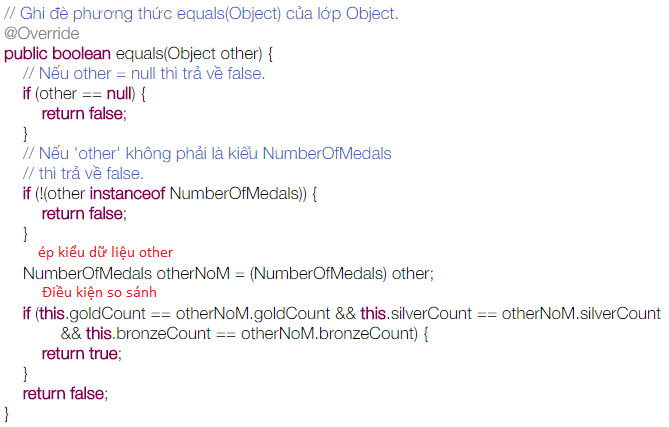


* Default Object.compare()
  + = 0 khi obj1 = obj2
  + > 0 khi obj1 > obj2
  + < 0 khi obj1 < obj2
* Chú ý: Check Object = null
* Order: Collections.sort(list, new Object());
* VD:

Collections.sort(list, new CompareName());

Sau đó so sánh tiếp: Collections.sort(list, new CompareAge());

* **Equals(Object obj)**
  + Equals() là so sánh giá trị (== là so sánh địa chỉ)



* **HashCode()**
  + Luôn luôn override method hashCode() khi override equals() trong Hash Collection
    - Nếu chỉ override equal() mà không override HashCode thì khi thêm các Object có giá trị giống nhau vào trong HashSet

Ảnh hưởng tới việc sắp sếp Object

* + Được sử dụng khi chúng ta insert object vào Hash Collection (HashTable, HashMap hoặc HashSet)
    - Khi insert một object vào hashtable, phương thức hashCode() sẽ trả về giá trị hash code của object, và giá trị này được sử dụng để xác định vị trí của object được lưu trữ trong hashtable.
    - Tương tự khi chúng ta get object từ hashtable, giá trị hash code này được sử dụng để tìm nơi lưu trữ của object trong hashtable.
  + HashCode()sẽ trả về kiểu int (đại diện cho mã băm của đối tượng )

2 đối tượng bằng nhau khi chúng có HashCode() bằng nhau và equal() = true (có nghĩa là chúng trỏ tới cùng 1 đối tượng trên vùng nhớ HEAP)

* **Implement**

<https://www.javatpoint.com/Comparable-interface-in-collection-framework>

<https://www.javatpoint.com/Comparator-interface-in-collection-framework>