**BÀI TẬP**

**Nội dung**

**Bài 1:**

Với Collection Framework, Các phần tử trong Set không bị trùng lặp

Ví dụ 1 khi convert List sang Set:

|  |
| --- |
| List<Integer> nums = **new** ArrayList<Integer>();  nums.add(4);  nums.add(4);  nums.add(4);  Set<Integer> setNums = **new** HashSet<Integer>(nums);  System.***out***.println(setNums.size()); // kết quả 1 |

Ví dụ 2 Nhưng khi convert List Employee sang Set thì lại không như vậy

|  |
| --- |
| List<Employee> empls = **new** ArrayList<Employee>();  empls.add(**new** Employee("Huy", 13));  empls.add(**new** Employee("Huy", 14));  empls.add(**new** Employee("Huy", 13));  Set<Employee> set = **new** HashSet<Employee>(empls);  System.***out***.println(set.size()); // Kết quả 3 |

Tìm hiểu nguyên nhân là do việc so sánh 2 đối tượng Employee sẽ khác so sánh 2 Integer

1. Việc cần làm là viết lại hàm equals và hashCode trong class Employee để kết quả Ví dụ 2 là 2 chứ không phải là 3
2. Cũng tương tự hãy đổi HashSet thành TreeSet và sử dụng Comparable để đạt kết quả như bước 1

**Bài 2**

Tạo class Employee có các biến tối thiểu: Id, name, salary,age

1. Tạo ra 1 list danh sách nhân viên tùy ý
2. Sắp xếp theo thứ tự của name trong employee
3. Sắp xếp theo mức lương tư cao tới thấp
4. Lọc ra 3 người trẻ nhất vào có mức lương cao nhất
5. Tìm ra các Employee có tên chứa kí tự N

**Gợi ý:** Sử dụng cách viết Java 8 với ý 4,5 sẽ nhanh hơn, sử dụng các hàm filter, group….

Sử dụng Comparator hoặc Comparable để làm cho các ý còn lại