Note: Sắp xếp theo họ tên:

//Y7: Sắp theo họ tên

# Duyệt list

## Cách 1 – for i

for (int i = 0; i < listNV.size(); i++) {

listNV.get(i).xuat();

}

## Cách 2 – for each

for(NhanVien x:listNV){

x.xuat();

}

## Cách3 – Dùng biểu thức lambda

listNV.forEach(x -> {

x.xuat();

});

# Sắp theo chuỗi

## Cách 1

public void sapTheoHoTen() {

Comparator<NhanVien> com = new Comparator<NhanVien>() {

@Override

public int compare(NhanVien o1, NhanVien o2) {

return o1.getHoTen().compareTo(o2.getHoTen());

}

};

listNV.sort(com);

System.out.println("DSNV sắp theo họ tên là:");

xuatDSNV();

}

## Cách 2- dùng dạng lambda

public void sapTheoHoTen() {

Collections.sort(listNV, (NhanVien o1, NhanVien o2)

-> {

return o1.getHoTen().compareToIgnoreCase(o2.getHoTen());

});

System.out.println("DSNV sắp theo họ tên là:");

xuatDSNV();

}

## Cách 3- dùng dạng lambda

public void sapTheoHoTen() {

listNV.sort((o1, o2)-> {

return o1.getHoTen().compareTo(o2.getHoTen());

});

System.out.println("DSNV sắp theo họ tên là:");

xuatDSNV();

}

# Sắp theo số thực

//C1:

public void sapTheoThuNhapGiamDan() {

Comparator<NhanVien> com = new Comparator<NhanVien>() {

@Override

public int compare(NhanVien o1, NhanVien o2) {

Double d1 = o1.getThuNhap();

Double d2 = o2.getThuNhap();

return d2.compareTo(d1);

}

};

listNV.sort(com);

}

//C2:

//Y8: Sắp theo thu nhập - Dùng biểu thức lambda:

public void sapTheoThuNhapGiamDan\_Lambda() {

Collections.sort(listNV, (NhanVien o1, NhanVien o2) -> {

Double d1 = o1.getThuNhap();

Double d2 = o2.getThuNhap();

return d2.compareTo(d1);

});

}

//C3:

public void sapTheoThuNhapGiamDan\_Lambda() {

listNV.sort((o1, o2) -> {

Double d1 = o1.getThuNhap();

Double d2 = o2.getThuNhap();

return d2.compareTo(d1);

});

}

# In đẹp trong java

https://gpcoder.com/2352-huong-dan-su-dung-string-format-trong-java/

### ****Định dạng kiểu chuỗi (String)****

* **%s** : sẽ in chuỗi tham số đầu vào.
* **%15s** : sẽ in chuỗi như tham số đầu vào. Nếu số chữ số nhỏ hơn 15, đầu ra sẽ được thêm khoảng trắng ở bên trái.
* **%-6s** : sẽ in chuỗi như tham số đầu vào. Nếu số chữ số nhỏ hơn 6, đầu ra sẽ được thêm khoảng trắng ở bên phải.
* **%.8s** : in tối đa tối chuỗi 8 ký tự.

### ****Định dạng kiểu số nguyên (Integer)****

* **%d** : sẽ in số nguyên như tham số đầu vào.
* **%6d** : sẽ in số nguyên như tham số đầu vào. Nếu số chữ số nhỏ hơn 6, đầu ra sẽ được thêm khoảng trắng ở bên trái.
* **%-6d** : sẽ in số nguyên như tham số đầu vào. Nếu số chữ số nhỏ hơn 6, đầu ra sẽ được thêm khoảng trắng ở bên phải.
* **%06d** : sẽ in số nguyên như tham số đầu vào. Nếu số chữ số nhỏ hơn 6, đầu ra sẽ được thêm ký tự số 0 ở bên trái.
* **%.2d** : in tối đa 2 chữ số của số nguyên.

### ****Định dạng kiểu số thực (Float)****

* **%f** : sẽ in số thực như tham số đầu vào.
* **%15f** : sẽ in số thực như tham số đầu vào. Nếu số chữ số nhỏ hơn 15, đầu ra sẽ được thêm khoảng trắng ở bên trái.
* **%.8f** : hiển thị tối đa 8 chữ số thập phân của số.
* **%9.4f** : hiển thị tối đa 4 chữ số thập phân của số. Đầu ra sẽ chiếm ít nhất 9 ký tự. Nếu số chữ số không đủ, nó sẽ được đệm khoảng trắng.

**VD: Xuất NhanVien:**

public void xuat() {

System.out.printf("%-15s %-20s %-20.1f %-20.1f %-20.1f %-20.1f %-20.1f %-20.1f \n",

maNv,hoTen,

luong,getThuNhap(),getThueThuNhap(),

0.0,0.0,0.0);

}

**VD: xuất DSNV:**

public void xuatDSNV() {

System.out.printf("%-15s %-20s %-20s %-20s %-20s %-20s %-20s %-20s \n",

"MaNV", "HoTen",

"Luong", "Thu Nhap", "Thue Thu Nhap", "Doanh So", "Hoa Hong", "Trach Nhiem");

for(NhanVien x:listNV){

x.xuat();

}

}

**Xuất Trưởng Phòng:**

@Override

public void xuat() {

System.out.printf("%-15s %-20s %-20.1f %-20.1f %-20.1f %-20.1f %-20.1f %-20.1f \n",

super.getMaNv(),super.getHoTen(),

super.getLuong(),getThuNhap(),super.getThueThuNhap(),

0.0,0.0,trachNhiem);

}

**Xuất Tiếp Thị:**

@Override

public void xuat() {

System.out.printf("%-15s %-20s %-20.1f %-20.1f %-20.1f %-20.1f %-20.1f %-20.1f \n",

super.getMaNv(),super.getHoTen(),

super.getLuong(),getThuNhap(),super.getThueThuNhap(),

doanhSo,hueHong,0.0);

}