

# Môn học lý thuyết ngôn ngữ hướng đối tượng

## Bài thực hành số 9: Đa hình trong hướng đối tượng

**Bài 1)** So sánh hai đối tượng với phương thức equals(). Hãy chạy các chương trình sau, cho biết kết quả và giải thích:

**a) Chương trình 1**

```
public class EqualsMethod1 {  
    public static void main(String[] args) {  
        Integer n1 = new Integer(47);  
        Integer n2 = new Integer(47);  
        System.out.println(n1 == n2);  
        System.out.println(n1 != n2);  
        System.out.println(n1.equals(n2));  
    }  
}
```

CT1:.....  
.....  
.....  
.....  
.....

**b) Chương trình 2**

```
class MyValue {  
    int i;  
}  
  
public class EqualsMethod2 {  
    public static void main(String[] args) {  
        MyValue v1 = new MyValue();  
        MyValue v2 = new MyValue();  
        v1.i = v2.i = 100;  
        System.out.println(v1==v2);  
        System.out.println(v1.equals(v2));  
    }  
}
```

CT2:.....  
.....  
.....  
.....  
.....

**c) Chương trình 3**

```
public class EqualsMethod3 {  
    public static void main(String[] args) {  
        String s1 = "Hello";  
        String s2 = "Hello";  
        System.out.println(s1==s2);  
        String s3 = new String ("Goodbye");  
        String s4 = new String ("Goodbye");  
        System.out.println(s3==s4);  
        String name = "Joe";  
        System.out.println("Joe".equals(name));  
        System.out.println("Joe".equalsIgnoreCase("joE"));  
    }  
}
```

CT3:.....  
 .....  
 .....  
 .....  
 .....

**Bài 2)** Viết lại (ghi đè) phương thức **equals()** cho lớp **MyValue** (phương thức này kế thừa từ lớp **Object** tuy nhiên muốn sử dụng cần viết lại để chỉ rõ cách so sánh như thế nào, trong trường hợp này hai đối tượng của lớp **MyValue** được xem là bằng nhau nếu chúng có giá trị của thuộc tính **i** bằng nhau).

```
class MyValue {
    int i;
    //Chú ý tham số của phương thức equals() là một đối tượng của lớp Object
    //Khi viết lại cần ép kiểu cho phù hợp
    public boolean equals(Object obj) {
        return (this.i == ((MyValue) obj).i);
    }
}

public class EqualsMethod4 {
    public static void main(String[] args) {
        MyValue v1 = new MyValue();
        MyValue v2 = new MyValue();
        v1.i = v2.i = 100;
        System.out.println(v1.equals(v2));
        System.out.println(v1==v2);
    }
}
```

Kết quả:

CT4:.....  
 .....  
 .....  
 .....  
 .....

**Bài 3)** Xây dựng chương trình như mô tả như sau:

- a) Viết mã nguồn cho lớp nhân viên theo sơ đồ UML sau: Định nghĩa các phương thức setA/getA. Tạo 2 phương thức constructors cho đối tượng, một phương thức không tham số và một phương thức đầy đủ tham số.

NhanVien
-maNhanVien: String
-tenNhanVien: String
-luongCoBan: double
-heSoLuong: double
+LUONG_MAX: double
+tangLuong(double):boolean
+tinhLuong(): double
+inTTin()
+equals(Object obj):boolean

- Lương = Lương cơ bản \* Hệ số lương

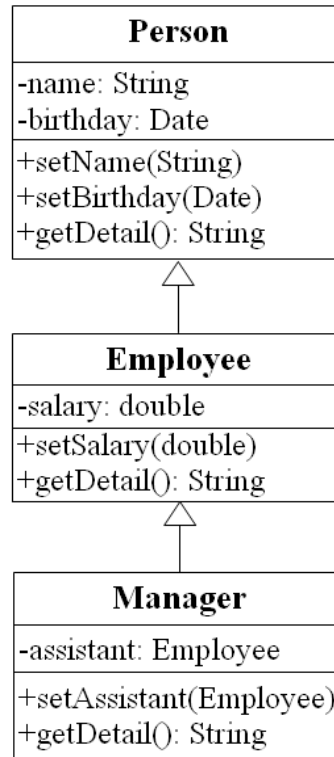
- Phương thức `inTTin()` hiển thị thông tin của đối tượng `NhanVien` tương ứng.
- Phương thức `tangLuong(double)` tăng hệ số lương hiện tại lên một lượng bằng giá trị tham số `double` truyền vào. Nếu điều này làm cho lương của nhân viên  $>$  lương tối đa cho phép thì không cho phép thay đổi, in ra thông báo và trả về `false`, ngược lại trả về `true`.
- Phương thức `inTTin()`: in thông tin của đối tượng nhân viên hiện thời
- Phương thức `equals()` so sánh hai đối tượng nhân viên: hai đối tượng nhân viên là “bằng nhau” nếu có mã nhân viên và họ tên trùng nhau.

b) Xây dựng lớp Phòng ban như sơ đồ UML sau:

PhòngBan
<b>+maPhong: String</b> <b>+tenPhong: String</b> <b>+DSNhanVien: NhanVien[]</b>
<b>+demNV(NhanVien nv): int</b> <b>+tongLuong(): double</b>

- Mỗi phòng ban có tối đa 15 nhân viên.
  - Phương thức **`demNV(NhanVien nv): int`** đếm xem trong phòng có bao nhiêu nhân viên bằng với nhân viên `nv` được truyền vào trong tham số.
  - Phương thức **`tongLuong(): double`** tính tổng lương của nhân viên trong phòng.
- c) Viết lớp cho phần thử nghiệm của các lớp vừa tạo.

**Bài 4)** Xây dựng các lớp theo sơ đồ UML như sau:



- Phương thức getDetail() in ra thông tin của loại đối tượng tương ứng.  
Cho biết kết quả khi chạy các đoạn chương trình dưới đây và giải thích?

a) Chương trình 1

```

public class Test1 {
    public static void main(String arg[]){
        Person p;
        Employee e = new Employee();
        p = e;
        p.setName("Hoa");
        p.setSalary(350000);
    }
  
```

CT1:.....  
 .....  
 .....  
 .....  
 .....

b) Chương trình 2

```

public class Test2 {
    public static void main(String arg[]){
        Employee e = new Employee();
        Person p = e;
        Employee ee = (Employee) p;
        Manager m = (Manager) ee;
        Person p2 = new Manager();
        Employee e2 = (Employee) p2;
    }
}
  
```

CT2:.....  
.....  
.....  
.....  
.....

- c) Chương trình 3: hoàn thành nốt đoạn chương trình dưới đây và cho biết kết quả thực thi và giải thích

```
public class Test3 {  
    String static teamInfo(Person p1, Person p2){  
        return "Leader: " + p1.getName() +  
            ", member: " + p2.getName();  
    }  
  
    public static void main(String arg[]){  
        Employee e1, e2;  
        Manager m1, m2;  
        // ... thiết lập thông tin cho các đối tượng e1, e2, m1, m2  
        . . .  
        . . .  
        System.out.println(teamInfo(e1, e2));  
        System.out.println(teamInfo(m1, m2));  
        System.out.println(teamInfo(m1, e2));  
    }  
}
```

CT3:.....  
.....  
.....  
.....  
.....

- d) Chương trình 4: hoàn thành nốt đoạn chương trình dưới đây và cho biết kết quả thực thi và giải thích về hoạt động của phương thức **getDetail()**

```
public class Test4 {  
    public static void main(String arg[]){  
        Person p = new Person();  
        // ... thiết lập thông tin cho các đối tượng p  
        Employee e = new Employee();  
        // ... thiết lập thông tin cho các đối tượng e  
        Manager m = new Manager();  
        // ... thiết lập thông tin cho các đối tượng m  
        Person pArr[] = {p, e, m};  
        for (int i=0; i< pArr.length; i++){  
            System.out.println(  
                pArr[i].getDetail());  
        }  
    }  
}
```

CT4:.....  
.....  
.....

.....  
.....  
.....  
.....  
.....