

## LAB 1: KẾ THỪA

### MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Xây dựng ứng dụng với nhiều lớp được tổ chức theo sự phân cấp thừa kế trong Java
- ✓ Sử dụng lại những gì đã có ở một lớp khác
- ✓ Ghi đè để hiệu chỉnh lại nội dung của phương thức ở lớp con
- ✓ Ứng dụng lớp trừu tượng và phương thức trừu tượng
- ✓ Ứng dụng interface

### PHẦN I

#### BÀI 1 (2 ĐIỂM)

Tạo lớp A theo hướng dẫn sau

```
public class A{
    protected int x, y;
    public A(int x, int y){
        this.x = x;
        this.y = y;
    }
    public A(int x){
        this(x, x);
    }
    public void print(){
        System.out.printf("x=%d, y=%d", x, y);
    }
}
```

Tạo lớp B kế thừa từ lớp A và override phương thức print() theo hướng dẫn sau

```
public class B{
```

```
int z;  
public B(int x, int y, int z){  
    super(x, y);  
    this.z = z;  
}  
public void print(){  
    System.out.printf("x=%d, y=%d, z=%d", x, y, z);  
}  
}
```

Tạo lớp C chứa phương thức main() và tạo 3 đối tượng sau đó gọi phương thức print() để xuất thông tin ra màn hình theo hướng dẫn sau:

```
A o1 = new A(5);  
A o2 = new A(5, 9);  
A o3 = new B(5, 6, 7);  
o1.print();  
o2.print();  
o3.print();
```

## **BÀI 2 (2 ĐIỂM)**

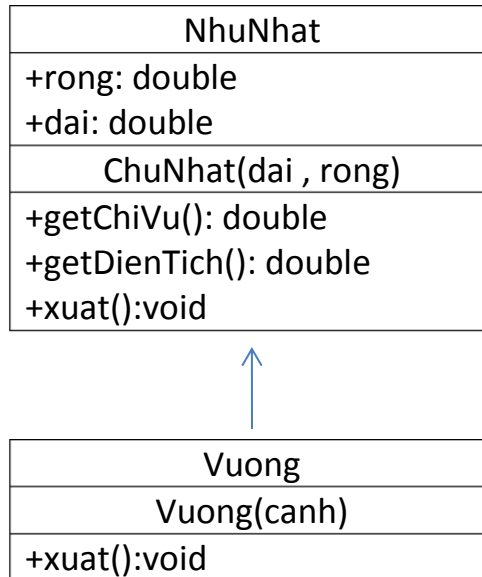
Xây dựng lớp ChuNhat gồm 2 thuộc tính là rong và dai và các phương thức getChuVi() và getDienTich() để tính chu vi và diện tích. Phương thức xuất() sẽ xuất ra màn hình chiều rộng, chiều dài, diện tích và chu vi.

Xây dựng lớp Vuong kế thừa từ lớp ChuNhat và ghi đè phương thức xuất() để xuất thông tin cạnh, diện tích và chu vi.

Viết chương trình nhập 2 hình chữ nhật và một hình vuông sau đó xuất ra màn hình.

### **HƯỚNG DẪN**

- ✓ Xây dựng các lớp theo kiến trúc phân cấp kế thừa như sau



- ✓ Hàm tạo của lớp **Vuong** gọi lại hàm tạo của lớp **ChuNhat** và truyền cạnh của hình vuông cho 2 tham số chiều dài và chiều rộng
- ✓ Ghi đè phương thức xuất để xuất thông tin của hình vuông
- ✓ Tạo lớp chứa phương thức `main()` và nhập chiều dài, chiều rộng của hình chữ nhật và cạnh của hình vuông. Sau đó sử dụng các lớp **ChuNhat** và **Vuong** để tạo các đối tượng và gọi phương thức `xuat()` để xem thông tin:
 

```

ChuNhat cn = new ChuNhat(dai, rong)
ChuNhat vu = new Vuong(canh)
cn.xuat();
vu.xuat()
      
```

## PHẦN II

### BÀI 3 (2 ĐIỂM)

Tạo lớp **SinhVienPoly** gồm 2 thuộc tính họ tên và ngành cùng với phương thức trừu tượng là `getDiem()`. Thêm phương thức `getHocLuc()` để xếp loại học lực. Lớp cũng bao gồm một phương thức `xuat()` để xuất họ tên, ngành, điểm và học lực ra màn hình.

### HƯỚNG DẪN

- ✓ Xây dựng lớp có mô hình như sau

SinhVienPoly
+hoTen: String
+nganh: String
SinhVienPoly (hoTen , nganh)
+getDiem(): double
+getHocLuc(): String
+xuat():void

- ✓ Vì chưa biết sinh viên này học những môn nào nên chưa tính được điểm vì vậy phương thức getDiem() phải là phương thức trừu tượng
- ✓ Chú ý lớp SinhVienPoly phải là lớp trừu tượng vì có phương thức getDiem() là phương thức trừu tượng
- ✓ Phương thức getHocLuc() được viết bình thường vẫn sử dụng phương thức getDiem() để lấy điểm của sinh viên mặc dù hiện tại vẫn chưa biết điểm được tính thế nào. Học lực được tính như sau
  - Yếu: điểm < 5
  - Trung bình: 5 <= điểm < 6.5
  - Khá: 6.5 <= điểm < 7.5
  - Giỏi: 7.5 <= điểm < 9
  - Xuất sắc: điểm >= 9

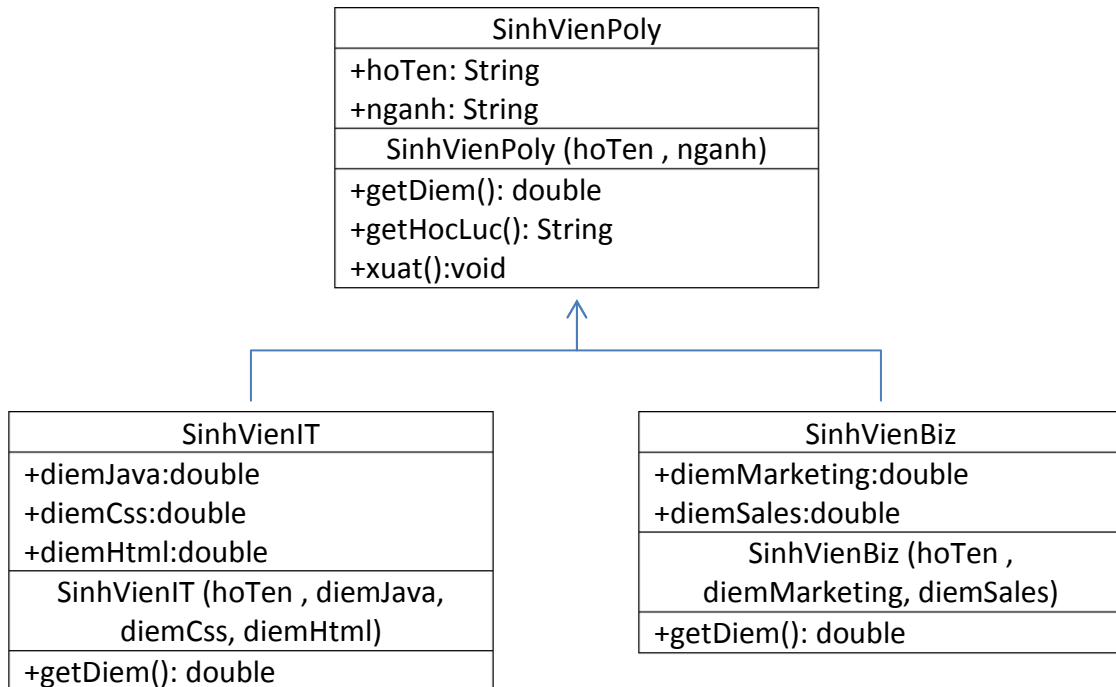
#### **BÀI 4 (2 ĐIỂM)**

Tạo lớp SinhVienIT và SinhVienBiz kế thừa từ lớp SinhVienPoly của bài 3.

- ✓ SinhVienIT gồm các thuộc tính điểm java, html, css. Ghi đè phương thức getDiem() để tính điểm cho sinh viên IT theo công thức  $(2 * \text{java} + \text{html} + \text{css}) / 4$
- ✓ SinhVienBiz gồm các thuộc tính điểm marketing, sales. Ghi đè phương thức getDiem() để tính điểm cho sinh viên Biz theo công thức  $(2 * \text{marketing} + \text{sales}) / 3$

#### **HƯỚNG DẪN**

- ✓ Tổ chức các lớp theo kiến trúc phân cấp kế thừa như sau



- ✓ Ghi đè phương thức getDiem() trên 2 lớp SinhVienIT và SinhVienBiz theo yêu cầu của đề để tính điểm cho các sinh viên của các ngành.

### BÀI 5 (2 ĐIỂM)

Tạo một interface có tên là DbAction và khai báo 4 phương thức insert(), update(), delete() và select()theo cú pháp sau:

```

void insert();
void update();
void delete();
void select();
    
```

Tạo 2 lớp Product và Order thực thi theo interface DbAction. Viết mã cho các phương thức insert(), update(), delete() và select() cho các lớp bằng cách xuất tên của phương thức kết hợp với tên lớp ra màn hình. Ví dụ trong phương thức insert() của lớp Product viết mã là System.out.println("Insert product");

Tạo lớp DbManager chứa phương thức main(). Tạo 2 đối tượng từ 2 lớp Product và Order và gọi phương thức insert() của mỗi đối tượng.

```
DbAction db1 = new Product();  
DbAction db2 = new Order();  
db1.insert();  
db2.insert();
```

**Chú ý:**

- ✓ Phần I và Phần II chỉ áp dụng cho dạy tích hợp. Sinh viên làm phần 1 và phần 2 theo 2 buổi học khác nhau tương ứng với 2 phần lý thuyết đã dạy trong bài học.
- ✓ Nếu giảng dạy theo phương pháp truyền thống thì sinh viên phải thực hiện tất cả các bài trong một buổi.