# LAB3. LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (BUỔI 2)

1. **Hướng dẫn**

***Bài tập 2***. Xây dựng lớp *SinhVien* kế thừa từ lớp *ConNguoi* đã xây dựng ở bài 1. Lớp *SinhVien* có thêm các thuộc tính sau: mã sinh viên, lớp, điểm toán, điểm lý, điểm hóa .

1. Khai báo các thuộc tính và viết các phương thức *nhapTTSinhVien()* và *hienThiTTSinhVien()* để nhập và hiển thị đầy đủ thông tin của lớp SinhVien.
2. Ghi đè phương thức *nhapTT()* và *hienThiTT()* của lớp cha.
3. Viết phương thức tính điểm trung bình cho các môn học với yêu cầu:
   * Tính điểm trung bình cho cả 3 môn Toán, Lý, Hóa.
   * Tính điểm trung bình của 2 môn học bất kỳ. Ví dụ Toán và Lý, Toán và Hóa, Lý và Hóa.

***Hướng dẫn:***

1. Để kế thừa từ một lớp, ta sử dụng từ khóa *extends,* sử dụng như sau:

**class** SinhVien **extends** ConNguoi

{

**public** String maSV;

**public** String lop;

**public** double diemToan;

**public** double diemLy;

**public** double diemHoa;

**public** **void** nhapTTSinhVien(){

// Có thể gọi tới phương thức nhapTT()

// từ lớp cha (lớp ConNguoi) qua toán tử super.

super.nhapTT();

System.out.println("Nhập mã sinh viên: ");

maSV = nhapString();

System.out.println("Nhập lớp: ");

lop = nhapString();

System.out.println("Nhập điểm Toán: ");

diemToan = nhapDouble();

System.out.println("Nhập điểm Lý: ");

diemLy = nhapDouble();

System.out.println("Nhập điểm Hóa: ");

diemHoa = nhapDouble();

}

**public** **void** hienThiTTSinhVien(){

// Gọi tới phương thức hiển thị thông tin từ lớp cha

super.hienThiTT();

System.out.println("Mã sinh viên: "+ maSV);

System.out.println("Lớp: "+ lop);

System.out.println("Điểm Toán: "+ diemToan);

System.out.println("Điểm Lý: "+ diemLy);

System.out.println("Điểm Hóa: "+ diemHoa);

}

}

Trong hàm ***main*** ta viết như sau để kiểm tra kết quả:

public static void main(String[] args) {

// Khai báo đối tượng SinhVien có tên là sv

SinhVien sv = new SinhVien();

// Gọi tới phương thức nhapTTSinhVien

// để nhập thông tin cho đối tượng sv

sv.nhapTTSinhVien();

// Hiển thị thông tin vừa nhập

// cho đối tượng sv ra ngoài màn hình

sv.hienThiTTSinhVien();

System.out.print("Tuổi của sinh viên là: " + sv.tinhTuoi());

}

1. Trong phần a, ta có thể thấy rằng, ta vẫn có thể gọi tới phương thức *nhapTTSinhVien* để thực hiện nhập thông tin cho đối tượng bằng cách gọi như sau:

sv.nhapTTSinhVien();

Tuy nhiên, ta vẫn có thể gọi tới phương thức nhập thông tin của lớp *ConNguoi* mà lớp *SinhVien* đã kế thừa. Nhưng khi ta gọi tới phương thức này, ta chỉ có thể nhập được các thông tin cơ bản của lớp *ConNguoi* (bao gồm họ tên, năm sinh, địa chỉ, giới tính) như đã định nghĩa mà thôi:

sv.nhapTT();

Như vậy, giả sử ta có lớp *CanBo* kế thừa từ lớp *ConNguoi*, mà việc nhập thông tin cho lớp *CanBo* sẽ khác với việc nhập thông tin của *SinhVien*. Ngoài ra, ta có đến 2 phương thức thực hiện chức năng nhập thông tin. Điều đó có thể dẫn tới trường hợp nhầm lẫn cho người lập trình. Lúc này ta có thể sử dụng tới ghi đè để thay đổi chức năng thực hiện của một phương thức nằm trong lớp cha mà lớp đã kế thừa. Cụ thể là thay đổi chức năng thực hiện của phương thức *nhapTT* của lớp *ConNguoi*.

Và điều đó tương tự với phương thức *hienThiTT()*

Ta viết như sau:

class SinhVien extends ConNguoi

{

@Override

public void nhapTT()

{

**super**.nhapTT();

System.out.println("Nhập mã sinh viên: ");

maSV = nhapString();

System.out.println("Nhập lớp: ");

lop = nhapString();

System.out.println("Nhập điểm Toán: ");

diemToan = nhapDouble();

System.out.println("Nhập điểm Lý: ");

diemLy = nhapDouble();

System.out.println("Nhập điểm Hóa: ");

diemHoa = nhapDouble();

}

@Override

public void hienThiTT()

{

System.out.println("Mã sinh viên: "+ maSV);

System.out.println("Tên sinh viên: "+ hoTen);

System.out.println("Lớp: "+ lop);

System.out.println("Địa chỉ: " + diaChi);

System.out.println("Tuổi: " + tinhTuoi());

System.out.println("Điểm Toán: "+ diemToan);

System.out.println("Điểm Lý: "+ diemLy);

System.out.println("Điểm Hóa: "+ diemHoa);

}

}

Như vậy, khi ta gọi tới phương thức nhapTT(), việc thực hiện nhập thông tin sẽ được thực hiện theo chức năng được đã được ghi đè ở lớp SinhVien chứ không còn thực hiện theo chức năng đã được định nghĩa ở lớp ConNguoi.

* Chú ý: Trong phương thức nhập thông tin, ta cần dùng tới toán tử ***super*** để phân biệt giữa việc gọi phương thức *nhapTT()* của lớp *ConNguoi* và lớp *SinhVien*. Cụ thể, trong lớp *SinhVien*, nếu ta gọi *super.nhapTT()* thì sẽ gọi tới phương thức *nhapTT()* được định nghĩa ở lớp *ConNguoi*, nếu chỉ gọi *nhapTT()* thì sẽ gọi tới phương thức *nhapTT()* ta đang ghi đè ở lớp *SinhVien*.
* Đối với phương thức tinhTuoi() được định nghĩa ở lớp ConNguoi nhưng không được ghi đè ở lớp SinhVien, ta có thể gọi *super.tinhTuoi()* hay *tinhTuoi()* đều giống nhau.

1. Các phương thức tính điểm trung bình ta sử dụng nạp chồng như sau:

**class** SinhVien extends ConNguoi

{

// Thực hiện tính điểm trung bình của 3 môn học

public double tinhDiemTB()

{

return (diemToan + diemLy + diemHoa)/3;

}

// Thực hiện tính điểm trung bình của 2 môn bất kỳ

public double tinhDiemTB(double diem1, double diem2)

{

return (diem1 + diem2)/2;

}

}

Trong hàm ***main*** ta sử dụng như sau:

public static void main(String[] args) {

SinhVien sv = new SinhVien();

sv.nhapTT();

System.out.print("Điểm trung bình 3 môn học là: "

+ sv.tinhDiemTB());

System.out.print("Điểm trung bình 2 môn Toán, Lý là: "

+ sv.tinhDiemTB(sv.diemToan, sv.diemLy));

}

1. **Bài tập**

**Bài 2**. Xây dựng lớp Máy in gồm các thông tin: Trọng lượng máy, năm sản xuất, hãng sản xuất. Sau đó, xây dựng lớp dẫn xuất: Máy in kim, ngoài các thuộc tính của máy in ra còn có thêm thuộc tính : số kim (int), tốc độ in (trang/ phút - int).

Xây dựng lớp Máy in Laser ngoài các thuộc tính của máy in còn có thêm các thuộc tính: Độ phân giải (int), tốc độ in (int).

Hai lớp dẫn xuất này có các phương thức: Nhập: nhập các thông tin của máy in, Xuất: xuất các thông tin của máy in ra màn hình.

Xây dựng chương trình chính nhập vào thông tin của n máy in kim và m máy in Laser. Xuất các thông tin đó lên màn hình.

**Bài 3.** Xây dựng lớp PERSON gồm các thông tin sau: Hoten (char[50]), Ngaysinh (char[12]), Quequan (char[100]) và xây dựng lớp DIEM gồm: Diểmtoan (int), Diemly (int), Điểmhoá (int).

Xây dựng lớp HOCSINH kế thừa từ 2 lớp trên có thêm dữ liệu: Lop (char [30]), Tongdiem (int) và các phương thức nhập dữ liệu từ bàn phím và xuất dữ liệu ra màn hình. Yêu cầu cả 3 lớp trên đều toán tử khởi tạo để khởi tạo các dữ liệu là số thì giá trị = 0, dữ liệu là xâu thì giá trị = “”. Phải viết chương trình chính để minh hoạ sử dụng lớp vừa xây dựng.

**Bài 4**. Xây dựng lớp **Canh** với các thuộc tính: tên cạnh, chiều dài và các toán tử tạo lập cho lớp.

Xây dựng lớp **DaGiac** với các thuộc tính: số cạnh, danh sách các cạnh.

* Viết toán tử tạo lập với tham số truyền vào là số cạnh của đa giác.
* Viết các phương thức cho lớp:

|  |  |
| --- | --- |
| **Tên phương thức** | **Chức năng** |
| void nhapDaGiac() | Cho phép nhập thông tin cho các cạnh của đa giác |
| double tinhChuVi(); | Tính chu vi của đa giác |
| void hienThi(); | Hiển thị thông tin của đa giác bao gồm:   * Số cạnh của đa giác. * Chu vi của đa giác * Danh sách các cạnh của đa giác |