

BÀI 6: Lập trình hướng đối tượng nâng cao



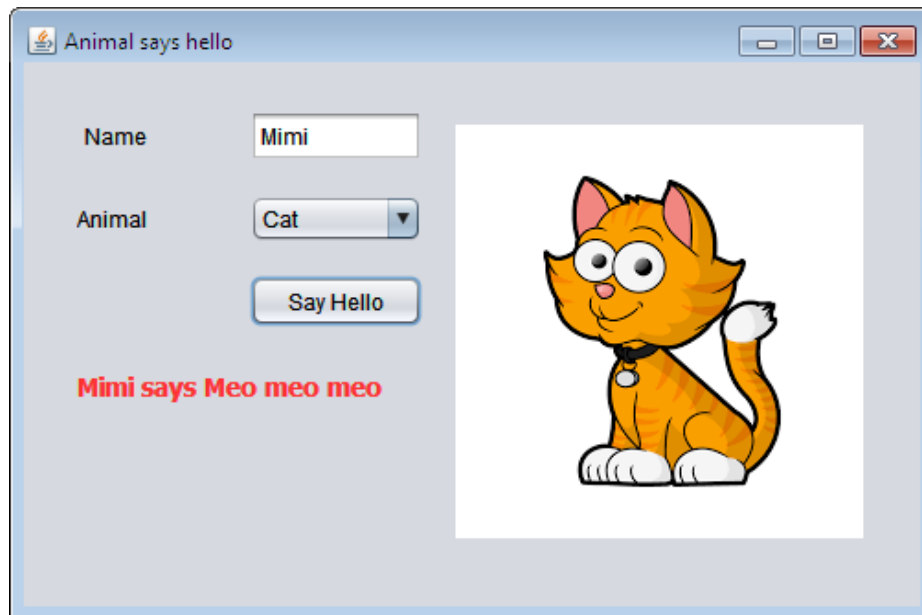
Mục tiêu chính:

Hiện thực ứng dụng áp dụng kế thừa

Áp dụng kiến thức về Abstract class và Interface để giải quyết các bài toán

6.1. Animal say hello

- ✓ **Yêu cầu:** Xây dựng chương trình hiển thị thông tin của một số con vật



- ✓ **Hướng dẫn sử dụng:**
 - Nhập tên và chọn một con vật, nhấn button "Say hello", chương trình xuất tên, tiếng kêu và hình ảnh của con vật được chọn.
- ✓ **Tóm tắt yêu cầu**
 - **Thiết kế giao diện người dùng:**
 - frmAnimalSayHello: FrmAnimalSayHello (extends từ JFrame)
 - * (Các thể hiện phía dưới đều nằm trong Frame)
 - txtName: JTextField (nhập liệu)
 - cmbAnimal: JComboBox (chọn Dog/Cat/Pig)
 - lblHello: JLabel (kết xuất)
 - lblImage: JLabel (kết xuất)
 - btnSayHello: JButton (Xử lý hiển thị thông tin)
 - **Nhập:**
 - Tên

- Chọn 1 con vật
- **Xuất:**
 - Tên, tiếng kêu, hình ảnh
- **Qui tắc xử lý:**
 - Không có
- ✓ **Thuật giải**
 - Không có
- ✓ **Hướng dẫn**
 - Xây dựng lớp Animal, có các thuộc tính name, image. Trong lớp này có phương thức sayHello() trả về chuỗi "Hello".

```
public class Animal {  
    String name;  
    String image;  
    public String getName() {  
        return name;  
    }  
    public String getImage() {  
        return image;  
    }  
    public Animal() {  
    }  
    public Animal(String name, String image) {  
        this.name = name;  
        this.image = image;  
    }  
    public String sayHello() {  
        return "Hello!";  
    }  
}
```

- Xây dựng các lớp Dog/ Cat/ Pig ... kế thừa từ lớp Animal có phương thức sayHello() trả về chuỗi chào tương ứng với từng loại, ví dụ như: "Gâu gâu gâu", "Meo meo meo", "Oii oii"

```
public class Dog extends Animal {  
    @Override  
    public String sayHello() {  
        return "Gâu gâu gâu";  
    }  
}
```

```

public Dog() {
}
public Dog(String name, String image) {
    super(name, image);
}
}
// Học viên làm tương tự cho lớp Cat/ Pig

```

- Trên form, khởi tạo lớp và xuất kết quả tương ứng với lựa chọn của người dùng.

```

private void btnSayActionPerformed(java.awt.event.ActionEvent evt) {
    String name = txtName.getText();
    String imagePath = "src/images/";
    ImageIcon img = null;
    switch (cbbAnimal.getSelectedItem().toString()) {
        case "Dog":
            Dog d = new Dog(name, "1_dog.png");
            lblHello.setText(d.name + " says " + d.sayHello());
            img = new ImageIcon(imagePath + d.getImage());
            lblImage.setIcon(img);
            break;
        case "Cat":
            Cat c = new Cat(name, "1_cat.png");
            lblHello.setText(c.name + " says " + c.sayHello());
            img = new ImageIcon(imagePath + c.getImage());
            lblImage.setIcon(img);
            break;
        default:
            Pig p = new Pig(name, "1_pig.jpg");
            lblHello.setText(p.name + " says " + p.sayHello());
            img = new ImageIcon(imagePath + p.getImage());
            lblImage.setIcon(img);
            break;
    }
}

```

6.2. Tính lương nhân viên

- ✓ **Yêu cầu:** Xây dựng ứng dụng tính lương nhân viên

Lương được tính theo công thức sau:

- Lương cơ bản cố định: 1.550.000

- Nhân viên = lương cơ bản * hệ số
- Nhân viên hành chính = lương cơ bản * hệ số + phụ cấp
- Nhân viên kỹ thuật = lương cơ bản * hệ số + (số dự án hoàn thành trong tháng x 1.000.000)
- Nhân viên kinh doanh = lương cơ bản * hệ số + (số sản phẩm * 20.000)
- Nhân viên lãnh đạo = lương cơ bản * hệ số + 10% tổng lương (tiền trách nhiệm)

Tính lương nhân viên

Loại NV: Hành chính

Hệ số lương: 3

Phụ cấp: 1000000

Tính lương

Lương: 5650000.0

Khi chọn "Hành chính" và điền các số liệu

Tính lương nhân viên

Loại NV: Kỹ thuật

Hệ số lương: 3

Số dự án: 2

Tính lương

Lương: 6650000.0

Khi chọn "Kỹ thuật" và điền các số liệu

Tính lương nhân viên

Loại NV: Kinh doanh

Hệ số lương: 3

Số sản phẩm: 30

Tính lương

Lương: 5250000.0

Khi chọn "Kinh doanh" và điền các số liệu

Tính lương nhân viên

Loại NV: Lãnh đạo

Hệ số lương: 5.5

Số sản phẩm: 30

Tính lương

Lương: 9377500.0

Khi chọn "Lãnh đạo" và điền các số liệu

✓ **Hướng dẫn sử dụng:**

- Chọn loại nhân viên
- Nhập các giá trị theo loại nhân viên
- Nhấn nút "Tính lương"

✓ **Tóm tắt yêu cầu**

▪ **Thiết kế giao diện người dùng:**

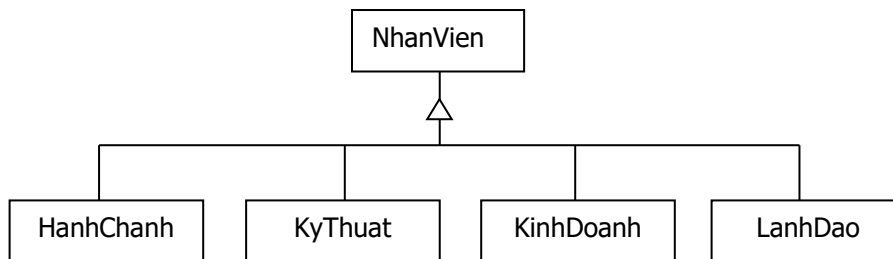
- frmTinhLuong: FrmTinhLuong

** Học viên tự xây dựng form như hình trên, lưu ý cần thay đổi nội dung label sau khi chọn loại nhân viên cho phù hợp*

- **Nhập:**
 - Các thông tin tương ứng với từng loại nhân viên
- **Xuất:**
 - Lương của nhân viên
- **Qui tắc xử lý:**
 - Tính lương như yêu cầu trên

✓ **Thuật giải**

Thiết kế lớp theo sơ đồ sau:



✓ **Hướng dẫn**

- Xây dựng lớp NhanVien có lương cơ bản = 1.550.000 là hằng số và phương thức tinhLuong()

```

public class NhanVien {
    final double LUONGCOBAN = 1550000;
    double hesoLuong;
    public NhanVien(double hesoLuong) {
        this.hesoLuong = hesoLuong;
    }
    public double getHesoLuong() {
        return hesoLuong;
    }
    public void setHesoLuong(double hesoLuong) {
        this.hesoLuong = hesoLuong;
    }
    public double tinhLuong(){
        return this.LUONGCOBAN * this.hesoLuong;
    }
}
  
```

- Xây dựng các lớp HanhChanh có thêm thuộc tính phụ cấp và phương thức tinhLuong() trả về lương cơ bản * hệ số + phụ cấp

```
public class HanhChinh extends NhanVien {
    double phuCap;
    public HanhChinh(double phuCap, double hesoLuong) {
        super(hesoLuong);
        this.phuCap = phuCap;
    }
    public double getPhuCap() {
        return phuCap;
    }
    public void setPhuCap(double phuCap) {
        this.phuCap = phuCap;
    }
    @Override
    public double tinhLuong() {
        return super.tinhLuong() + this.phuCap;
    }
}
```

- Thực hiện tương tự cho các lớp KyThuat, KinhDoanh, LanhDao kế thừa từ lớp NhanVien có thuộc tính riêng và phương thức tinhLuong() trả về lương tương ứng với từng loại nhân viên theo công thức tính lương trên.
- Trên form, khởi tạo lớp và tính lương ứng với lựa chọn của người dùng.

```
private void btnTinhActionPerformed(java.awt.event.ActionEvent evt) {
    double hesoLuong = Double.parseDouble(txtHeSo.getText());
    int thongSo = Integer.parseInt(txtThongSo.getText());

    switch (cbbLoaiNV.getSelectedIndex()) {
        case 0: // hanh chinh
            HanhChinh hc = new HanhChinh(thongSo, hesoLuong);
            txtLuong.setText(String.valueOf(hc.tinhLuong()));
            break;
    }
}
```

```

case 1: // ky thuat
    Kythuat kt = new Kythuat(thongSo, hesoLuong);
    txtLuong.setText(String.valueOf(kt.tinhLuong()));
    break;
case 2: // kinh doanh
    KinhDoanh kd = new KinhDoanh(thongSo, hesoLuong);
    txtLuong.setText(String.valueOf(kd.tinhLuong()));
    break;
case 3: // lanh dao
    LanhDao ld = new LanhDao(hesoLuong);
    txtLuong.setText(String.valueOf(ld.tinhLuong()));
    break;
}
}

```

6.3. Quản lý sách thư viện

- ✓ **Yêu cầu: Xây dựng ứng dụng quản lý sách thư viện**

The screenshot shows a Java Swing window titled "Quản lý sách". It features a form with the following fields and controls:

- Thông tin chung:**
 - Mã sách: gk002
 - Tên sách: PHP & MySQL tập 1
 - Ngày nhập: 26/03/2015
 - NXB: ĐHQG
 - Đơn giá: 36000
 - Số lượng: 50
 - Loại sách: ☒ Sách giáo khoa, ☐ Sách tham khảo
 - Tình trạng: (dropdown)
 - Thuế (1-100):
 - Buttons: and
- Summary Section:**
 - Buttons: and
 - Tổng thành tiền: 5100000.0
 - List of books:
 - gk001 - Quy trình phát triển phần mềm - 26/03/2015 - ĐHQG - 100 - 42000.0 - 4200000.0
 - gk002 - PHP & MySQL tập 1 - 26/03/2015 - ĐHQG - 50 - 36000.0 - 900000.0

Quản lý các loại sách – tab Sách giáo khoa

Quản lý các loại sách – tab Sách tham khảo

Mô tả: Thư viện X quản lý danh sách các loại sách. Thông tin về các loại sách:

- Sách giáo khoa: Mã sách, tên sách, ngày nhập (ngày/tháng/năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ)
 - Nếu tình trạng sách là mới thì: thành tiền = số lượng * đơn giá
 - Nếu tình trạng sách là cũ thì: thành tiền = số lượng * đơn giá * 50%
- Sách tham khảo: Mã sách, tên sách, ngày nhập (ngày/tháng/năm), đơn giá, số lượng, nhà xuất bản, thuế
 - Thành tiền = (số lượng * đơn giá) + (số lượng * đơn giá) * thuế (với thuế có giá trị từ 1% đến 100%)

Dựa vào mô tả trên, hãy:

- Tạo lớp Sach với các thuộc tính và phương thức chung.
- Tạo lớp SachGiaoKhoa và lớp SachThamKhao kế thừa từ lớp sách với các thuộc tính riêng và phương thức cần thiết.
- Nhập xuất danh sách các loại sách.
- Tính tổng thành tiền cho từng loại.
- Tính trung bình cộng đơn giá của các sách tham khảo.

✓ **Hướng dẫn sử dụng:**

- Nhập thông tin chung về sách, sau đó chọn loại sách và thông tin chi tiết cho loại sách được chọn > Nhấn “Nhập sách” => Hiển thị thông tin các sách trong thư viện với tổng thành tiền của từng loại và đơn giá trung bình của sách tham khảo.
- Nhấn “Tiếp tục” => Các ô nhập liệu sẽ được xóa nội dung để người dùng nhập dữ liệu mới.

✓ Tóm tắt yêu cầu

▪ Thiết kế giao diện người dùng:

- frmQuanLySach: JFrame

* (Các thể hiện phía dưới đều nằm trong form)

- txtMaSach: Text Field (nhập liệu)
- txtTenSach: Text Field (nhập liệu)
- txtNgayNhap: Text Field (kết xuất)
- txtNXB: Text Field (nhập liệu)
- txtDonGia: Text Field (nhập liệu)
- txtSoLuong: Text Field (kết xuất)
- rbTgLoaiSach: Radio Button Group (nhóm 2 radio button Sách giáo khoa và Sách tham khảo)
 - rbSGK: Radio Button (chọn)
 - cbbTinhTrang: Combo Box (chọn)
 - rbSTK: Radio Button (chọn)
 - txtThue: Text Field (nhập liệu)
- btnNhap: Button (xử lý tính toán và in kết quả)
- btnTiepTuc: Button (xử lý xóa nội dung của các điều khiển trên màn hình để người dùng tiếp tục nhập liệu)
- tpThongKe: Tab Panel (chứa thông tin thống kê)
 - pnlSGK: Panel (chứa thông tin Sách giáo khoa)
 - txtTienSGK: Text Field (kết xuất)
 - lstSGK: List (kết xuất)
 - pnlSTK: Panel (chứa thông tin Sách tham khảo)
 - txtTienSTK: Text Field (kết xuất)
 - txtTBCDonGia: Text Field (kết xuất)
 - lstSTK: List (kết xuất)

▪ Nhập:

- Mã sách
- Tên sách
- Ngày nhập

- Nhà xuất bản
- Đơn giá
- Số lượng
- Sách giáo khoa (Tình trạng)/ Sách tham khảo (Thuế)
- **Xuất:**
 - Danh sách sách
 - Tổng thành tiền sách giáo khoa/ sách tham khảo
 - Đơn giá trung bình sách tham khảo
- ✓ **Hướng dẫn**
 - Tạo một lớp có tên là Sach gồm có
 - Các thuộc tính: maSach, tenSach, ngayNhap, NXB, donGia, soLuong
 - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thông tin thuộc tính trên
 - Phương thức tinhThanhTien() với thành tiền = số lượng * đơn giá
 - Tạo một lớp có tên là SachGiaoKhoa kế thừa lớp Sach gồm có
 - Các thuộc tính: tinhTrang
 - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thông tin thuộc tính ứng với các thuộc tính
 - Phương thức tinhThanhTien() với thành tiền = số lượng * đơn giá nếu sách mới và thành tiền = số lượng * đơn giá * 50% nếu là sách cũ
 - Tạo một lớp có tên là SachThamKhao kế thừa từ lớp Sach gồm có:
 - Các thuộc tính: thue
 - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thông tin thuộc tính ứng với các thuộc tính
 - Phương thức tinhThanhTien() với thành tiền = số lượng * đơn giá + (số lượng * đơn giá * thuế) /100
 - Thực hiện những hàm xử lý sự kiện cho các nút nhấn:
 - Khi chọn nút “Nhập sách”, thực hiện:
 - Gọi sử dụng class sách tương ứng với thông tin nhập
 - Đưa sách vừa nhập vào danh sách Sách giáo khoa hoặc Sách tham khảo
 - Tính toán
 - Xuất kết quả ra các điều khiển
 - Khi chọn nút “tiếp tục”, thực hiện:
 - Đặt lại giá trị rỗng cho các điều khiển để người dùng tiếp tục nhập sách

6.4. Tính chu vi & diện tích các hình

- ✓ **Yêu cầu:** Viết chương trình tính chu vi và diện tích của một số hình như sau:

Abstract Class

Hình tròn Hình chữ nhật Hình vuông

Bán kính 10 Tính chu vi

Chi vi 62.83 Tính diện tích

Diện tích 314.16

Hình tròn

Abstract Class

Hình tròn Hình chữ nhật Hình vuông

Chiều dài 10

Chiều rộng 5

Chu vi 30.00 Tính chu vi

Diện tích 50.00 Tính diện tích

Hình chữ nhật

Abstract Class

Hình tròn Hình chữ nhật Hình vuông

Cạnh 20

Chu vi 80.00 Tính chu vi

Diện tích 400.00 Tính diện tích

Hình vuông

- ✓ **Hướng dẫn sử dụng:**

- Nhập bán kính hình tròn hoặc chiều dài và chiều rộng hình chữ nhật hoặc cạnh hình vuông, nhấn "Tính chu vi"/ "Tính diện tích" => hiển thị kết quả tương ứng

✓ Tóm tắt yêu cầu

▪ Thiết kế giao diện người dùng:

- frmAbstractClass: JFrame (chính thuộc tính defaultCloseOperation của JFrame là DISPOSE)

* (Các thể hiện phía dưới đều nằm trong form)

- pnlHình: Tab Panel (chứa 3 panel hình tròn, hình chữ nhật, hình vuông)
 - pnlCircle: Panel (chứa thông tin hình tròn)
 - txtR: Text Field (nhập liệu)
 - txtPTron: Text Field (kết xuất)
 - txtSTron: Text Field (kết xuất)
 - btnTinhPTron (xử lý tính chu vi hình tròn và in kết quả)
 - btnTinhSTron (xử lý tính diện tích hình tròn và in kết quả)
 - pnlRectangle: Panel (chứa thông tin hình chữ nhật)
 - txtCanhA: Text Field (nhập liệu)
 - txtCanhB: Text Field (nhập liệu)
 - txtPCN: Text Field (kết xuất)
 - txtSCN: Text Field (kết xuất)
 - btnTinhPCN (xử lý tính chu vi hình chữ nhật và in kết quả)
 - btnTinhSCN (xử lý tính diện tích hình chữ nhật và in kết quả)
 - pnlSquare: Panel (chứa thông tin hình vuông)
 - txtCanh: Text Field (nhập liệu)
 - btnTinhPV (xử lý tính chu vi hình vuông và in kết quả)
 - btnTinhSV (xử lý tính diện tích hình vuông và in kết quả)

▪ Nhập:

- Bán kính/ chiều dài & chiều rộng/ cạnh

▪ Xuất:

- Chu vi
- Diện tích

✓ Hướng dẫn

- Xây dựng lớp trừu tượng Shape như sau:

```
public abstract class Shape {
    public abstract double perimeter();
}
```

```
public abstract double area();  
}
```

- Xây dựng lớp Circle kế thừa lớp Shape

```
public class Circle extends Shape {  
    public double r;  
    public void setR(double r) {  
        this.r = r;  
    }  
    public double getR() {  
        return r;  
    }  
    @Override  
    public double perimeter() {  
        return 2* Math.PI * r;  
    }  
    @Override  
    public double area() {  
        return Math.PI * Math.pow(r,2);  
    }  
}
```

- Làm tương tự cho các lớp còn lại
- Khởi tạo và gọi sử dụng các lớp này trong form để tính chu vi, diện tích và xuất kết quả

```
// tính chu vi hình tròn  
private void btnTinhPTronActionPerformed(java.awt.event.ActionEvent evt) {  
    double r = Double.parseDouble(txtR.getText());  
    Circle circle = new Circle();  
    circle.r = r;    txtPTron.setText(String.format("%.2f",square.perimeter()));  
}  
  
// tính diện tích hình tròn  
private void btnTinhSTronActionPerformed(java.awt.event.ActionEvent evt) {
```

```

double r = Double.parseDouble(txtR.getText());

Circle circle = new Circle();

circle.r = r;

txtSTron.setText(String.format("%.2f",circle.area()));

}

```

- Làm tương tự cho các hình còn lại

6.5. Chơi nhạc

- ✓ **Yêu cầu: Viết chương trình chơi nhạc cụ như sau:**

Khởi động chương trình

Nhạc cụ có dây

Nhạc cụ không dây

Mô tả: Chương trình có 2 loại nhạc cụ:

- Nhạc cụ có dây: tên nhạc cụ, tên hình, số dây.
 - Chơi nhạc được thực hiện như sau: Tên đàn + số dây
- Nhạc cụ không dây: tên nhạc cụ, tên hình, sử dụng
 - Chơi nhạc được thực hiện như sau: "Phát ra âm thanh khi tác động vào" + sử dụng

Dựa vào mô tả trên, hãy:

- Tạo lớp trừu tượng `NhacCu` với thuộc tính chung và phương thức trừu tượng `choiNhac()`
- Tạo lớp `NhacCuCoDay` kế thừa từ lớp nhạc cụ với các thuộc tính riêng và hiện thực phương thức `choiNhac()`.
- Tạo lớp `NhacCuKhongDay` kế thừa từ lớp nhạc cụ với các thuộc tính riêng và hiện thực phương thức `choiNhac()`
- Hiển thị hình ảnh nhạc cụ và thông tin chơi nhạc

✓ **Hướng dẫn sử dụng:**

- Nhập Tên nhạc cụ, Tên hình
- Chọn Loại nhạc cụ "Có dây", nhập Số dây
- Hoặc
- Chọn Loại nhạc cụ "Không dây", nhập Sử dụng
- Nhấn "Chơi nhạc" -> hiển thị kết quả

✓ Tóm tắt yêu cầu

▪ Thiết kế giao diện người dùng:

- frmChoiNhac: JFrame

* (Các thể hiện phía dưới đều nằm trong form)

- txtTen: Text Field (nhập liệu)
- txtHinh: Text Field (nhập liệu)
- cbbLoai: Combo Box (chọn Có dây/ không dây)
- txtSoDay: Text Field (nhập liệu)
- txtSuDung: Text Field (nhập liệu)
- lblHinh: Label (kết xuất)
- txtaChoiDan: Text Area (kết xuất)
- btnChoi: Button (xử lý và in kết quả)

▪ Nhập:

- Tên nhạc cụ, Tên hình
- Loại nhạc cụ “Có dây”, Số dây
Hoặc
- Loại nhạc cụ “Không dây”, Sử dụng

▪ Xuất:

- Hình ảnh nhạc cụ
- Thông tin chơi nhạc

✓ Hướng dẫn

- Tạo một trừu tượng có tên là NhạcCu

```
public abstract class NhạcCu {  
    String ten;  
    String hinh;  
    abstract public String chơiNhac();  
    public NhạcCu() {  
    }  
    public NhạcCu(String ten, String hinh) {  
        this.ten = ten;  
        this.hinh = hinh;  
    }  
}
```

```

public String getTen() {
    return ten;
}

public void setTen(String ten) {
    this.ten = ten;
}

public String getHinh() {
    return hinh;
}

public void setHinh(String hinh) {
    this.hinh = hinh;
}
}

```

- Tạo lớp NhạcCuCoDay kế thừa lớp NhạcCu

```

public class NhạcCuCoDay extends NhạcCu {
    int soDay;

    public NhạcCuCoDay(int soDay, String ten, String hinh) {
        super(ten, hinh);
        this.soDay = soDay;
    }

    public int getSoDay() {
        return soDay;
    }

    public void setSoDay(int soDay) {
        this.soDay = soDay;
    }

    @Override
    public String chơiNhac() {
        return this.getTen() + " có " + this.soDay + " dây";
    }
}

```

```
}
```

- Làm tương tự cho lớp NhạcCuKhongDay
- Gọi sử dụng các lớp trong form, xử lý và hiển thị kết quả

```
private void btnChoiActionPerformed(java.awt.event.ActionEvent evt) {  
    String choi = "";  
    String hinh = "src/images/" + txtDuongDan.getText();  
  
    if(cbbLoai.getSelectedIndex()==0){ // nhạc cu co day  
        NhạcCuCoDay nccd = new NhạcCuCoDay(Integer.parseInt(txtSoDay.getText()),  
txtTen.getText(), hinh);  
        choi = nccd.choiNhac();  
    }  
    else{  
        NhạcCuKhongDay nckd = new NhạcCuKhongDay(txtSuDung.getText(),  
txtTen.getText(), hinh);  
        choi = nckd.choiNhac();  
    }  
    txtaChoiDan.setText(choi);  
    ImageIcon iconLogo = new ImageIcon(hinh);  
    lblHinh.setIcon(iconLogo);  
}
```

- Chú ý: cần tạo thư mục images trong thư mục src của project và chuẩn bị sẵn một số hình ảnh cần thiết cho bài tập này.

6.6. Phương tiện giao thông và động cơ

- ✓ **Yêu cầu:** Xây dựng ứng dụng tính vận tốc và trung bình tiêu thụ nhiên liệu như sau:

Phương tiện giao thông - động cơ

Thông tin chung

Quãng đường (km) 150 Thời gian (giờ) 2 Nhiên liệu (lít) 3

☒ Xe hơi ☐ Tàu thủy

Tên xe Civic Trọng lượng (kg)

Hãng sản xuất Honda Sức tải (kg)

Vận tốc nước xuôi dòng (km/h)

Vận tốc nước ngược dòng (km/h)

Nhiên liệu tiêu hao (lít)

Tính vận tốc - Tiêu thụ nhiên liệu

Vận tốc (km/giờ) 75.0 Tiêu thụ nhiên liệu (km/ lít) 50.0

Xe hơi

Phương tiện giao thông - động cơ

Thông tin chung

Quãng đường (km) 600 Thời gian (giờ) 5 Nhiên liệu (lít) 10

☐ Xe hơi ☒ Tàu thủy

Tên xe Trọng lượng (kg) 1000

Hãng sản xuất Sức tải (kg) 2000

Vận tốc nước xuôi dòng (km/h) 20

Vận tốc nước ngược dòng (km/h) 10

Nhiên liệu tiêu hao (lít) 2

Tính vận tốc - Tiêu thụ nhiên liệu

Vận tốc (km/giờ) 130.0 Tiêu thụ nhiên liệu (km/ lít) 75.0

Tàu thủy

Mô tả: Chương trình có 2 loại phương tiện giao thông:

- Xe hơi: tên xe, hãng sản xuất.
 - Vận tốc xe hơi (km/giờ) = Quãng đường/ Thời gian
 - Tiêu hao nhiên liệu xe hơi (km/lít) = Quãng đường/ Số lít
- Tàu thủy: Trọng lượng, Sức tải, Vận tốc nước xuôi dòng, Vận tốc nước ngược dòng, Nhiên liệu tiêu hao
 - Vận tốc tàu thủy (km/giờ)= Quãng đường/ Thời gian + Vận tốc nước xuôi dòng – Vận tốc nước ngược dòng

- Tiêu thụ nhiên liệu tàu thủy (km/lít) = Quãng đường / (Số lít – nhiên liệu tiêu hao)

Dựa vào mô tả trên, hãy:

- Tạo interface DongCo có phương thức trừu tượng tinhTieuThuNhiemLieu(double QuangDuong, double SoLit)
- Tạo interface PhuongTienGiaoThong có phương thức trừu tượng tinhVanToc(double QuangDuong, double ThoiGian)
- Tạo lớp XeHoi implements 2 interface trên và hiện thực các phương thức trừu tượng.
- Tạo lớp TauThuy implements 2 interface trên và hiện thực các phương thức trừu tượng.
- Tính toán và hiển thị vận tốc, tiêu thụ nhiên liệu tương ứng với từng loại

✓ Hướng dẫn sử dụng:

- Nhập Quãng đường, Thời gian, Nhiên liệu
- Chọn “Xe hơi” => nhập Tên xe, Hãng sản xuất => Nhấn “Tính vận tốc – Tiêu thụ nhiên liệu” => Hiển thị kết quả

Hoặc

- Chọn “Tàu thủy” => nhập Trọng lượng, Sức tải, Vận tốc nước xuôi dòng, Vận tốc nước ngược dòng, Nhiên liệu tiêu hao => Nhấn “Tính vận tốc – Tiêu thụ nhiên liệu” => Hiển thị kết quả

✓ Tóm tắt yêu cầu

▪ Thiết kế giao diện người dùng:

- frmInterface: JFrame (chỉnh thuộc tính defaultCloseOperation của JFrame là DISPOSE)

** (Các thể hiện phía dưới đều nằm trong form)*

- txtQuangDuong: Text Field (nhập liệu)
- txtThoiGian: Text Field (nhập liệu)
- txtNhiemLieu: Text Field (nhập liệu)
- btnGTauXe: Button Group
 - rbtXe: RadioButton (chọn)
 - txtTenXe: Text Field (nhập liệu)
 - txtHSX: Text Field (nhập liệu)
 - rbtTau: Radio Button (chọn)
 - txtTrongLuong: Text Field (nhập liệu)
 - txtSucTai: Text Field (nhập liệu)
 - txtXuoi: Text Field (nhập liệu)
 - txtNguoc: Text Field (nhập liệu)
 - txtTieuHao: Text Field (nhập liệu)
- txtVanToc: Text Field (kết xuất)

- txtTieuThuNL: Text Field (kết xuất)
- btnVanToc_TieuThuNL: Button (xử lý tính toán và in kết quả)
- **Nhập:**
 - Quãng đường
 - Thời gian
 - Nhiên liệu
 - Chọn “Xe hơi”
 - Tên xe
 - Hãng sản xuất
 - Chọn “Tàu thủy”
 - Trọng lượng
 - Sức tải
 - Vận tốc nước xuôi dòng
 - Vận tốc nước ngược dòng
 - Nhiên liệu tiêu hao
- **Xuất:**
 - Vận tốc
 - Tiêu thụ nhiên liệu

✓ **Hướng dẫn**

- Tạo Interface `PhuongTienGiaoThong`

```
public interface PhuongTienGiaoThong {
    double tinhVanToc(double QuangDuong, double ThoiGian);
}
```

- Tạo interface `DongCo`:

```
public interface DongCo {
    double tinhTieuThuNhiemLieu(double QuangDuong, double SoLit);
}
```

- Tạo class `XeHoi` implements 2 interface trên:
 - Thuộc tính: `tenXe`, `hangSanXuat`
 - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thuộc tính trên

- Phương thức: hiện thực các phương thức của 2 interface trên

```
public class XeHoi implements PhuongTienGiaoThong, DongCo{
    String tenXe;
    String hangSanXuat;
    public XeHoi(String tenXe, String hangSanXuat) {
        this.tenXe = tenXe;
        this.hangSanXuat = hangSanXuat;
    }
    public String getTenXe() {
        return tenXe;
    }
    public void setTenXe(String tenXe) {
        this.tenXe = tenXe;
    }
    public String getHangSanXuat() {
        return hangSanXuat;
    }
    public void setHangSanXuat(String hangSanXuat) {
        this.hangSanXuat = hangSanXuat;
    }
    /**
     *
     * @param QuangDuong
     * @param ThoiGian
     * @return
     */
    @Override
    public double tinhVanToc(double QuangDuong, double ThoiGian) {
        return QuangDuong/ThoiGian;
    }
    /**
```

```

*
* @param QuangDuong
* @param SoLit
* @return
*/
@Override
public double tinhTieuThuNhiemLieu(double QuangDuong, double SoLit) {
    return QuangDuong/SoLit;
}
}

```

- Tạo class `TauThuy` implements 2 interface trên:
 - Thuộc tính: `trongLuong`, `sucTai`, `vanTocNuocXuoiDong`, `vanTocNuocNguocDong`, `nhiemLieuTieuHao`
 - Phương thức khởi tạo mặc định và khởi tạo có tham số truyền vào là các thuộc tính trên
 - Phương thức: hiện thực các phương thức của 2 interface trên

```

public class TauThuy implements PhuongTienGiaoThong, DongCo{
    double trongLuong;
    int sucTai;
    double vanTocNuocXuoiDong;
    double vanTocNuocNguocDong;
    double nhiemLieuTieuHao;

    public TauThuy(double trongLuong, int sucTai, double vanTocNuocXuoiDong, double
vanTocNuocNguocDong, double nhiemLieuTieuHao) {
        this.trongLuong = trongLuong;
        this.sucTai = sucTai;
        this.vanTocNuocXuoiDong = vanTocNuocXuoiDong;
        this.vanTocNuocNguocDong = vanTocNuocNguocDong;
        this.nhiemLieuTieuHao = nhiemLieuTieuHao;
    }
}

```



```
public double getTrongLuong() {
    return trongLuong;
}
public void setTrongLuong(double trongLuong) {
    this.trongLuong = trongLuong;
}
public int getSucTai() {
    return sucTai;
}
public void setSucTai(int sucTai) {
    this.sucTai = sucTai;
}
public double getVanTocNuocXuiDong() {
    return vanTocNuocXuiDong;
}

public void setVanTocNuocXuiDong(double vanTocNuocXuiDong) {
    this.vanTocNuocXuiDong = vanTocNuocXuiDong;
}
public double getVanTocNuocNguocDong() {
    return vanTocNuocNguocDong;
}
public void setVanTocNuocNguocDong(double vanTocNuocNguocDong) {
    this.vanTocNuocNguocDong = vanTocNuocNguocDong;
}
public double getNhienLieuTieuHao() {
    return nhienLieuTieuHao;
}
public void setNhienLieuTieuHao(double nhienLieuTieuHao) {
    this.nhienLieuTieuHao = nhienLieuTieuHao;
}
```

```
@Override  
public double tinhVanToc(double QuangDuong, double ThoiGian) {  
    double vanToc = (QuangDuong/ThoiGian) + this.vanTocNuocXuoaiDong -  
this.vanTocNuocNguocDong;  
    return vanToc;  
}  
@Override  
public double tinhTieuThuNhiemLieu(double QuangDuong, double SoLit) {  
    return QuangDuong/(SoLit - this.nhiemLieuTieuHao);  
}  
}
```

- Khởi tạo và sử dụng 2 lớp vừa xây dựng trong form, tính toán và hiển thị kết quả