

Huỳnh Thanh Tâm  
D16HT01  
MSSV: 1624801040051

## 15 BÀI TẬP JAVA

### PHẦN 1: 5 BÀI TẬP CƠ BẢN

#### Bài 1:

**Yêu cầu:** Giải phương trình bậc 2 với các hệ số nhập từ bàn phím

PTBac2.java

```
public class PTBac2 {
    public static void GiaiPTBac1(double a, double b){
        if(a==0)
        {
            if(b==0)
            {
                System.out.println("PT có vô số nghiệm");
            }
            else {
                System.out.println("PT vô nghiệm");
            }
        }
        else {
            System.out.println("PT có nghiệm x= "+ -b/a);
        }
    }
}
/**
 * Giải pt bậc 2:  $ax^2 + bx + c = 0$ 
 * @param a - hệ số a
 * @param b - hệ số b
 * @param c - hệ số c
 */
public static void GiaiPTBac2(double a, double b, double c) {
    if(a==0) {
        GiaiPTBac1(b,c);
    }
    else {
        double delta = b*b-4*a*c;
        if(delta<0) {
            System.out.println("PT vô nghiệm");
        }
        else if(delta>0) {
            double x1=(-b+Math.sqrt(delta))/2*a;

```

```

        double x2=(-b-Math.sqrt(delta))/2*a;
        System.out.println("PT có 2 nghiệm phân biệt x1=
"+ x1 +"; x2="+x2);
    }
    else
        System.out.println("PT có 1 nghiệm kép: x= "+ -
b/2*a);
    }
}

public static void main(String[] args) {
    System.out.println("Giải pt bậc 2: ax^2 + bx + c = 0 ");
    Scanner sc=new Scanner(System.in);
    do {
        System.out.println("Nhập hệ số a: ");
        double a=sc.nextDouble();
        System.out.println("Nhập hệ số b: ");
        double b=sc.nextDouble();
        System.out.println("Nhập hệ số c: ");
        double c=sc.nextDouble();
        GiaiPTBac2(a, b, c);
        System.out.println("Tiếp tục? (c/k): ");
        String line=new Scanner(System.in).nextLine();
        if(line.equalsIgnoreCase("k"))
            break;

    }while(true);
    System.out.println("Kết thúc!");
}
}

```

## Bài 2:

Yêu cầu: Nhập n, tính tổng số chẵn nhỏ hơn n.

TongNSo.java

```

public class TongNSo {
    public static void main(String[] args) {
        do {
            try {
                int n=0;
                do {

                    System.out.println("Nhập n: ");
                    Scanner sc=new Scanner(System.in);
                    n=sc.nextInt();
                    if(n<0) {
                        System.out.println("Không được nhập
số âm!");
                    }
                }while(n<0);
            }
        }
    }
}

```

```

        System.out.println("Tổng số chẵn tự nhiên: "+
tongSoChan(n));

    }
    catch(Exception e) {
        System.out.println("Lỗi định dạng");
    }
    System.out.println("Tiếp tục? (c/k): ");
    String line=new Scanner(System.in).nextLine();
    if(line.equalsIgnoreCase("k"))
        break;

    }while(true);
    System.out.println("Kết thúc!");

}

private static int tongSoChan(int n) {
    int sum=0;
    for(int i=1;i<=n;i++) {
        if(i%2==0) {
            sum+=i;
        }
    }
    return sum;
}
}

```

### Bài 3:

Yêu cầu: Xử lý ngoại lệ bằng try..catch

XuLyNgoaiLe.java

```

public class XuLyNgoaiLe {

    public static void main(String[] args) {
        try {
            int kq=6/0;
            System.out.println("Kết quả 6/0:"+kq);

        } catch (ArithmeticException e) {
            System.out.println("Phép chia 0 không thể xử lý");
        }
        finally {
            System.out.println("Kết thúc!");
        }
    }

}

```

#### Bài 4:

Yêu cầu: Nhập vào mảng số nguyên và sắp xếp mảng tăng dần

SapXepMang.java

```
public class SapXepMang {
    public static void main(String[] args) {
        int n=0;
        Scanner sc=new Scanner(System.in);
        do {
            try {
                do {
                    System.out.println("Nhập n: ");

                    n=sc.nextInt();
                    if(n<0) {
                        System.out.println("Không được nhập
số âm!");
                    }
                }while(n<0);
            }
            catch(Exception e) {
                System.out.println("Lỗi định dạng");
            }
        }while(n<=0);

        int arr[]=new int[n];
        //Nhập phần tử vào mảng
        System.out.println("Nhập phần tử cho mảng:");

        for(int i=0;i<n;i++) {
            System.out.print("Nhập phần tử thứ "+ (i+1) + " :");
            arr[i]=sc.nextInt();
        }
        System.out.println("Thông tin mảng: ");
        for(int i=0;i<n;i++) {
            System.out.print(arr[i] + "\t");
        }
        //sắp xếp mảng
        Arrays.sort(arr);
        System.out.println("\nMảng sau khi sắp xếp: ");
        for(int i=0;i<n;i++) {
            System.out.print(arr[i]+ "\t");
        }
    }
}
```

#### Bài 5:

Yêu cầu: Tìm số hoàn hảo nhỏ hơn n(nhập từ bàn phím), tiến hành ghi mảng này xuống file.

DocGhiFile.java

```
public class DocGhiFile {
    private static boolean checkPerfectNumber(int n) {
        int sum=0;
        for(int i=1;i<=n/2;i++) {
            if(n%i==0) {
                sum+=i;
            }
        }
        if(sum==n) return true;
        return false;
    }
    public static boolean ghiFile(ArrayList<String>dsData, String
path) {
        try(OutputStreamWriter fos= new OutputStreamWriter(new
FileOutputStream(path), "UTF-8")) {
            for(int i=0;i<dsData.size();i++) {
                fos.write(dsData.get(i).toString());
            }
            fos.close();
            return true;
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }
    public static void main(String[] args) {
        String name="sohoanhao.txt";
        Scanner sc=new Scanner(System.in);
        int n;
        System.out.println("Nhập n: ");
        n=sc.nextInt();
        ArrayList<String> list=new ArrayList<String>();
        for(int i=1;i<=n;i++) {
            if(checkPerfectNumber(i))
                list.add(i+"\t");
        }
        try(OutputStreamWriter fos= new OutputStreamWriter(new
FileOutputStream(name), "UTF-8")) {
            for(int j=0;j<list.size();j++) {
                fos.write(list.get(j).toString());
            }
            fos.close();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## PHẦN 2: 5 BÀI TẬP OOP

### Bài 6:

**Yêu cầu:** Xây dựng lớp trừu tượng **Khach**, 2 lớp **KhachThuong** và **KhachVip** kế thừa lớp **Khach**, override lại phương thức  **tinhTien()**

#### LopTruuTuong.java

```
abstract class Khach{
    abstract void tinhTien();
}
class KhachThuong extends Khach{

    @Override
    void tinhTien() {
        System.out.println("Tính tiền cho khách thường");
    }
}
class KhachVip extends Khach{

    @Override
    void tinhTien() {
        System.out.println("Tính tiền cho khách VIP");
    }
}
public class LopTruuTuong {
    public static void main(String[] args) {
        try {
            System.out.println("Chọn khách hàng (1-VIP 2-
Thường): ");
            int choose=new Scanner(System.in).nextInt();
            switch(choose) {
                case 1:
                {
                    Khach k=new KhachVip();
                    k.tinhTien();
                    break;
                }

                case 2: {
                    Khach k=new KhachThuong();
                    k.tinhTien();
                    break;
                }

                default:
                    System.out.println("Sai lựa chọn");
                    break;
            }
        }
        catch(Exception e) {
```

```

        System.out.println("Lựa chọn không hợp lệ");
    }

}

```

## Bài 7: Tính kế thừa

**Yêu cầu:** Xây dựng lớp NhanVien, 2 lớp NhanVienChinhThuc và NhanVienThoiVu kế thừa từ lớp NhanVien và gọi phương thức tinhLuong()

**NhanVien.java**

```

public class NhanVien {
    protected int ma;
    protected String ten;

    public NhanVien()
    {

    }

    public NhanVien(int ma,String ten)
    {
        this.ma=ma;
        this.ten=ten;
    }

    public void setMa(int ma)
    {
        this.ma=ma;
    }

    public int getMa()
    {
        return this.ma;
    }

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

    public String getTen()
    {
        return this.ten;
    }

    public void tinhLuong()
    {
        System.out.println("Tính lương cho nhân viên "
+this.getTen());
    }
}

```

**NhanVienChinhThuc.java**

```

public class NhanVienChinhThuc extends NhanVien {

    public NhanVienChinhThuc()
    {
        super();
    }
    public NhanVienChinhThuc(int ma,String ten)
    {
        super(ma,ten);
    }

    @Override
    public void tinhLuong() {
        System.out.println("Tính lương nhân viên chính thức");
    }
}

```

NhanVienThoiVu.java

```

public class NhanVienThoiVu extends NhanVien
{
    public NhanVienThoiVu()
    {
        super();
    }
    public NhanVienThoiVu(int ma,String ten)
    {
        super(ma,ten);
    }
    @Override
    public void tinhLuong()
    {
        System.out.println("Tính lương nhân viên thời vụ");
    }
}

```

KeThua.java

```

public class KeThua {

    public static void main(String[] args) {
        NhanVienChinhThuc nv1=new NhanVienChinhThuc(1, "Nguyễn Văn
A");
        NhanVienThoiVu nv2=new NhanVienThoiVu(2, "Nguyễn Văn B");

        nv1.tinhLuong();
        nv2.tinhLuong();
    }
}

```

**Bài 8:**

**Yêu cầu:** Ghi xuống file 1 đối tượng Sach



DocGhiDoiTuong.java

```
class Sach implements Serializable {
    public String maSach;
    public String tenSach;
}

public class DocGhiDoiTuong {
    public static void main(String[] args) {
        Sach sach=new Sach();
        sach.maSach="1";
        sach.tenSach="Java cơ bản";

        try {
            FileOutputStream f = new FileOutputStream("sach.dat");
            ObjectOutputStream oos = new ObjectOutputStream(f);
            oos.writeObject(sach);
            oos.close();
            f.close();
        } catch (IOException e) {
            System.out.println("Có lỗi xảy ra!");
        }
    }
}
```

Bài 9:

Yêu cầu: Nhập danh sách đối tượng sinh viên từ bàn phím, tiến hành lưu thông tin danh sách, và sắp xếp danh sách theo điểm trung bình, đọc danh sách sau khi sắp xếp.

DocGhiDanhSachSV.java

```
class SinhVien implements Serializable{
    String maSV;
    String hoTen;
    double dTB;
}

public class DocGhiDanhSachSV {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        ArrayList<SinhVien> listSV=new ArrayList<>();
        System.out.println("Nhập số lượng sv: ");
        int n=sc.nextInt();
        for(int i=0;i<n;i++) {
            System.out.println("Thông tin sv thứ " +(i+1));
            SinhVien sv=new SinhVien();
            sc.nextLine();
            System.out.println("Mã sv: ");
            sv.maSV=sc.nextLine();

            System.out.println("Tên sv: ");
            sv.hoTen=sc.nextLine();
        }
    }
}
```

```

        System.out.println("Điểm TB: ");
        sv.dTB=sc.nextDouble();

        listSV.add(sv);
    }

    ghiFile(listSV, "dsSV.dat");
    //đọc danh sách sau khi đã sắp xếp
    ArrayList<SinhVien> newList=new ArrayList<>();

    try {

        FileInputStream fis=new
FileInputStream("dsSV.dat");
        ObjectInputStream ois=new
ObjectInputStream(fis);
        newList=(ArrayList<SinhVien>) ois.readObject();
        ois.close();
        fis.close();
    } catch (Exception e) {
        System.out.println("Lỗi");
    }
    sapXepDTB(newList);
    System.out.println("Danh sách sv sau khi đã sắp xếp");
    System.out.println("Mã sv\t Tên sv\t DTB");
    sapXepDTB(newList);
    for(int i=0;i<newList.size();i++) {

        System.out.println(newList.get(i).maSV + "\t" +
newList.get(i).hoTen + "\t" + newList.get(i).dTB);
    }
}

private static void sapXepDTB(ArrayList<SinhVien> listSV) {
    Collections.sort(listSV, new Comparator<SinhVien>() {
        @Override
        public int compare(SinhVien o1, SinhVien o2) {
            if(o1.dTB<o2.dTB)
                return 1;
            else if(o1.dTB>o2.dTB)
                return -1;
            return 0;
        }
    });
}

private static void ghiFile(ArrayList<SinhVien> listSV, String
name) {
    try {
        FileOutputStream fos=new FileOutputStream(name);

```

```

        ObjectOutputStream oos=new ObjectOutputStream(fos);
        oos.writeObject(listSV);
        oos.close();
        fos.close();
        System.out.println("Ghi thành công");
    } catch (Exception e) {
        System.out.println("Ghi lỗi");
    }
}
}
}

```

## Bài 10:

**Yêu cầu:** Xây dựng class Phong, 2 class PhongMayTinh và PhongThiNghiem kế thừa từ class Phong. Xây dựng phương thức tìm kiếm theo mã phòng.

Phong.java

```

public class Phong {
    private String maPhong;
    private String dayNha;
    private double dienTich;
    private int soBongDen;

    public String getMaPhong() {
        return maPhong;
    }
    public void setMaPhong(String maPhong) {
        this.maPhong = maPhong;
    }
    public String getDayNha() {
        return dayNha;
    }
    public void setDayNha(String dayNha) {
        this.dayNha = dayNha;
    }
    public double getDienTich() {
        return dienTich;
    }
    public void setDienTich(double dienTich) {
        this.dienTich = dienTich;
    }
    public int getSoBongDen() {
        return soBongDen;
    }
    public void setSoBongDen(int soBongDen) {
        this.soBongDen = soBongDen;
    }
    public void nhap() {
        System.out.println("Nhập mã phòng: ");
    }
}

```

```

        this.maPhong=new Scanner(System.in).nextLine();
        System.out.println("Nhập dãy nhà: ");
        this.dayNha=new Scanner(System.in).nextLine();
        System.out.println("Nhập diện tích: ");
        this.dienTich=new Scanner(System.in).nextDouble();
        System.out.println("Nhập số bóng đèn: ");
        this.soBongDen=new Scanner(System.in).nextInt();
    }
    public void xuat() {
        System.out.println("Mã phòng: "+this.getMaPhong());
        System.out.println("Dãy nhà: "+this.getDayNha());
        System.out.println("Diện tích: "+this.getDienTich());
        System.out.println("Số bóng đèn: "+this.getSoBongDen());
    }
}

```

PhongMayTinh.java

```

public class PhongMayTinh extends Phong {
    int soMayTinh;

    public int getSoMayTinh() {
        return soMayTinh;
    }

    public void setSoMayTinh(int soMayTinh) {
        this.soMayTinh = soMayTinh;
    }
    @Override
    public void nhap() {
        super.nhap();
        System.out.println("Nhập số máy tính: ");
        this.soMayTinh=new Scanner(System.in).nextInt();
    }
    @Override
    public void xuat() {
        super.xuat();
        System.out.println("Số máy tính: "+this.getSoMayTinh());
    }
}

```

PhongThiNghiem.java

```

public class PhongThiNghiem extends Phong {
    private String chuyenNganh;

    public String getChuyenNganh() {
        return chuyenNganh;
    }

    public void setChuyenNganh(String chuyenNganh) {
        this.chuyenNganh = chuyenNganh;
    }
}

```

```

@Override
public void nhap() {
    super.nhap();
    System.out.println("Nhập chuyên ngành: ");
    this.chuyenNganh=new Scanner(System.in).nextLine();
}
@Override
public void xuat() {
    super.xuat();
    System.out.println("Chuyên ngành:
"+this.getChuyenNganh());
}
}

TestPhong.java

public class TestPhong {

    public static void main(String[] args) {
        ArrayList<Phong> listPhong=new ArrayList<Phong>();
        Scanner sc=new Scanner(System.in);
        System.out.println("Nhập số lượng phòng: ");
        int n=sc.nextInt();
        for(int i=0;i<n;i++) {
            System.out.println("Chọn loại phòng: (1-Máy tính; 2-
Thí nghiệm) ");
            int choose=sc.nextInt();
            System.out.println("Thông tin phòng thứ "+(i+1));
            switch(choose) {
                case 1:{
                    Phong p=new PhongMayTinh();
                    p.nhap();
                    listPhong.add(p);
                    break;
                }
                case 2:{
                    Phong p=new PhongThiNghiem();
                    p.nhap();
                    listPhong.add(p);
                    break;
                }
                default:
                    System.out.println("Lựa chọn không
đúng");
                    break;
            }
        }

        String maPhong="";
        System.out.println("Nhập mã phòng cần tìm: ");
        maPhong=new Scanner(System.in).nextLine();
    }
}

```

```

        timKiem(listPhong, maPhong);
    }

    private static void timKiem(ArrayList<Phong> listPhong, String
maPhong) {
        int dem=0;
        for(int i=0;i<listPhong.size();i++) {
            if(listPhong.get(i).getMaPhong().equals(maPhong)) {
                System.out.println("Thông tin phòng có mã "+
maPhong);

                listPhong.get(i).xuat();
                dem++;
                break;
            }
        }
        if(dem==0)
            System.out.println("Không tìm thấy phòng");
    }
}

```

### PHẦN 3: 5 BÀI TẬP GIAO DIỆN

#### Bài 11:

The screenshot shows a Java Swing window titled "Quản lý xe máy". It features a form with six text input fields arranged in two columns. The left column contains "Mã xe", "Hãng sản xuất", and "Giá thành". The right column contains "Tên xe", "Năm sản xuất", and "Số lượng". Below these fields are two buttons: "Thêm" (Add) and "Tiếp tục" (Continue). At the bottom of the window, there is a table with six columns: "Mã Xe", "Tên xe", "Hãng sản xuất", "Năm sản xuất", "Giá thành", and "Số lượng". Below the table, there are two labels: "Tổng số lượng 0" and "Tổng giá thành 0".

**Yêu cầu:** Quản lý cửa hàng xe máy

**Chức năng:** nút thêm để thêm 1 xe máy vào table, nút tiếp tục để xóa trắng dữ liệu trong các textfield.

**Tính tổng số lượng và tổng giá thành trong table**

XeMay.java

```
Package baitapgiaodien.model;
```

```
public class XeMay {
    private String maXe;
    private String tenXe;
    private String hangSX;
    private int namSX;
    private double giaThanh;
    private int soLuong;
    public XeMay(String maXe, String tenXe, String hangSX, int
namSX, double giaThanh, int soLuong) {
        super();
        this.maXe = maXe;
        this.tenXe = tenXe;
        this.hangSX = hangSX;
        this.namSX = namSX;
        this.giaThanh = giaThanh;
        this.soLuong=soLuong;
    }

    public String getMaXe() {
        return maXe;
    }

    public void setMaXe(String maXe) {
        this.maXe = maXe;
    }

    public String getTenXe() {
        return tenXe;
    }

    public void setTenXe(String tenXe) {
        this.tenXe = tenXe;
    }

    public String getHangSX() {
        return hangSX;
    }

    public void setNhaSX(String hangSX) {
        this.hangSX = hangSX;
    }

    public int getNamSX() {
        return namSX;
    }

    public void setNamSX(int namSX) {
        this.namSX = namSX;
    }
}
```

```

    public double getGiaThanh() {
        return giaThanh;
    }

    public void setGiaThanh(double giaThanh) {
        this.giaThanh = giaThanh;
    }

    public int getSoLuong() {
        return soLuong;
    }

    public void setSoLuong(int soLuong) {
        this.soLuong = soLuong;
    }

    public double tinhGia(int soLuong, double giaThanh) {
        return soLuong*giaThanh;
    }

    @Override
    public String toString() {
        return this.getMaXe() + "-" + this.getTenXe() + "-" +
this.getHangSX() + "-" + this.getNamSX() + "-" + this.getGiaThanh()
+ "-" + this.getSoLuong();
    }

}

```

Giao diện QLXeMay.java

```

public class QLXeMay extends JFrame {

    private JPanel contentPane;
    private JTextField txtMaXe;
    private JTextField txtHangSanXuat;
    private JTextField txtTenXe;
    private JTextField txtNamSanXuat;
    private JTextField txtGiaThanh;
    private JLabel lblGiThnh;
    private JTable tblXe;
    private JTextField txtSoLuong;
    DefaultTableModel modelXe;
    int tongSL=0;
    double tongGiaThanh=0.0;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {

```



```

máy");

        QLXeMay frame = new QLXeMay("Quản lý xe

        frame.setVisible(true);
        frame.setLocationRelativeTo(null);
    } catch (Exception e) {
        e.printStackTrace();
    }
    });
}

/**
 * Create the frame.
 */
public QLXeMay(String name) {
    super(name);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 572, 365);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("Mã xe");
    lblNewLabel.setBounds(57, 17, 68, 14);
    contentPane.add(lblNewLabel);

    txtMaXe = new JTextField();
    txtMaXe.setBounds(161, 11, 86, 20);
    contentPane.add(txtMaXe);
    txtMaXe.setColumns(10);

    txtHangSanXuat = new JTextField();
    txtHangSanXuat.setColumns(10);
    txtHangSanXuat.setBounds(161, 50, 86, 20);
    contentPane.add(txtHangSanXuat);

    JLabel lblHngSnXut = new JLabel("Hãng sản xuất");
    lblHngSnXut.setBounds(57, 56, 94, 14);
    contentPane.add(lblHngSnXut);

    txtTenXe = new JTextField();
    txtTenXe.setColumns(10);
    txtTenXe.setBounds(375, 11, 86, 20);
    contentPane.add(txtTenXe);

    JLabel lblTnXe = new JLabel("Tên xe");
    lblTnXe.setBounds(286, 11, 65, 14);
    contentPane.add(lblTnXe);

    txtNamSanXuat = new JTextField();

```

```

txtNamSanXuat.setColumns(10);
txtNamSanXuat.setBounds(375, 47, 86, 20);
contentPane.add(txtNamSanXuat);

JLabel lblNmSnXut = new JLabel("Năm sản xuất");
lblNmSnXut.setBounds(286, 47, 79, 14);
contentPane.add(lblNmSnXut);

txtGiaThanh = new JTextField();
txtGiaThanh.setColumns(10);
txtGiaThanh.setBounds(161, 89, 86, 20);
contentPane.add(txtGiaThanh);

lblGiThnh = new JLabel("Giá thành");
lblGiThnh.setBounds(57, 95, 68, 14);
contentPane.add(lblGiThnh);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 159, 536, 114);
contentPane.add(scrollPane);

tblXe = new JTable();
tblXe.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
        "M\u00E3 Xe", "T\u00EAn xe", "H\u00E3ng", "N\u0103m s\u00E2n xu\u00E1t", "Gi\u00E1 th\u00E0nh", "S\u01ED l\u01B0\u01EBng"
    }
));
scrollPane.setViewportViewView(tblXe);
modelXe=(DefaultTableModel) tblXe.getModel();
JLabel lblNewLabel_1 = new JLabel("Tổng số lượng");
lblNewLabel_1.setBounds(84, 283, 86, 14);
contentPane.add(lblNewLabel_1);

JLabel lblTngGiThnh = new JLabel("Tổng giá thành");
lblTngGiThnh.setBounds(289, 283, 86, 14);
contentPane.add(lblTngGiThnh);

txtSoLuong = new JTextField();
txtSoLuong.setColumns(10);
txtSoLuong.setBounds(376, 89, 86, 20);
contentPane.add(txtSoLuong);

JLabel lblSLng = new JLabel("Số lượng");
lblSLng.setBounds(286, 89, 68, 14);
contentPane.add(lblSLng);

JButton btnThem = new JButton("Thêm");

```

```

btnThem.setBounds(169, 120, 89, 23);
contentPane.add(btnThem);

JButton btnTiepTuc = new JButton("Tiếp tục");
btnTiepTuc.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        txtMaXe.setText("");
        txtTenXe.setText("");
        txtHangSanXuat.setText("");
        txtNamSanXuat.setText("");
        txtGiaThanh.setText("");
        txtSoLuong.setText("");
    }
});
btnTiepTuc.setBounds(268, 120, 89, 23);
contentPane.add(btnTiepTuc);

JLabel lblTongGiaThanh = new JLabel("0");
lblTongGiaThanh.setForeground(Color.RED);
lblTongGiaThanh.setBounds(383, 283, 46, 14);
contentPane.add(lblTongGiaThanh);

JLabel lblTongSoLuong = new JLabel("0");
lblTongSoLuong.setForeground(Color.RED);
lblTongSoLuong.setBounds(175, 283, 46, 14);
contentPane.add(lblTongSoLuong);
btnThem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        String maXe=txtMaXe.getText();
        String tenXe=txtTenXe.getText();
        String hangSX=txtHangSanXuat.getText();
        try {
            int
namSX=Integer.parseInt(txtNamSanXuat.getText());
            double
giaThanh=Double.parseDouble(txtGiaThanh.getText());
            int
soLuong=Integer.parseInt(txtSoLuong.getText());
            if(maXe.isEmpty()||
                tenXe.isEmpty()||
                hangSX.isEmpty()||

String.valueOf(namSX).isEmpty()||

String.valueOf(giaThanh).isEmpty()||

String.valueOf(soLuong).isEmpty()) {

JOptionPane.showMessageDialog(null,"Vui lòng không bỏ trống
trường nào!", "Cảnh báo", JOptionPane.INFORMATION_MESSAGE);

```

```

        return;
    }
    modelXe.addRow(new Object[] {maXe, tenXe,
hangSX, namSX, giaThanh, soLuong});
    XeMay xe=new XeMay(maXe, tenXe, hangSX,
namSX, giaThanh, soLuong);

    tongSL+=xe.getSoLuong();
    lblTongSoLuong.setText(tongSL+"");
    tongGiaThanh+=xe.tinhGia(soLuong,
giaThanh);

    lblTongGiaThanh.setText(tongGiaThanh+"");

    }
    catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null,"Xin
nhập đúng dạng hoặc không bỏ trống", "Cảnh báo",
JOptionPane.INFORMATION_MESSAGE);
    }
    });
}
}

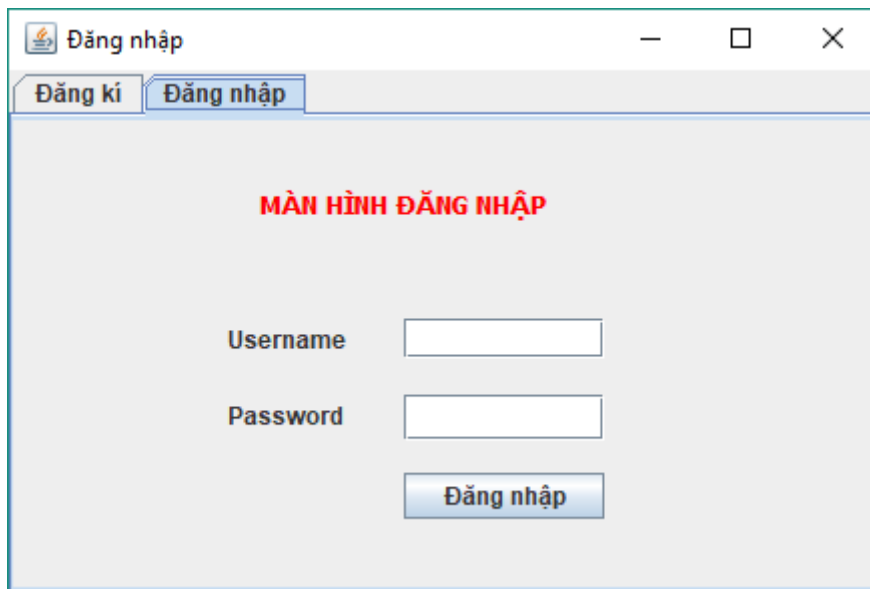
```

## Bài 12:

The screenshot shows a Java Swing window titled "Đăng nhập" (Login). It has two tabs: "Đăng kí" (Register) and "Đăng nhập" (Login). The "Đăng nhập" tab is currently selected. Inside the window, there are two text input fields labeled "Username" and "Password". Below these fields is a button labeled "Add account". At the bottom of the window, there is a table with two columns: "STT" (Serial Number) and "Username". The table is currently empty.

**Yêu cầu:** Xây dựng giao diện đăng ký, đăng nhập tài khoản

**Trong mục đăng ký:** nút Add account dùng để thêm tài khoản vào table



Trong mục Đăng nhập, khi nhấn nút Đăng nhập kiểm tra thông tin đăng nhập có trùng khớp.

TaiKhoan.java

```
package baitapgiaodien.model;

public class TaiKhoan implements Serializable{
    private String username;
    private String password;
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public TaiKhoan(String username, String password) {
        super();
        this.username = username;
        this.password = password;
    }
    @Override
    public String toString() {
        return this.getUsername() + "-" + this.getPassword();
    }
    public boolean checkAccount(TaiKhoan tk) {
        if(this.username==tk.getUsername() &&
this.password==tk.getPassword()) return true;
        return false;
    }
}
```

```
}
```

Giao diện: FormDangNhap.java

```
public class FormDangNhap extends JFrame {

    private JPanel contentPane;
    private JTextField txtUsername;
    private JPasswordField txtPassword;
    private JTable tblAccount;
    DefaultTableModel modelAccount;
    ArrayList<TaiKhoan> listTK=new ArrayList<TaiKhoan>();
    private int i=1;
    private JTextField txtTenDN;
    private JPasswordField txtMatKhau;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    FormDangNhap frame = new
FormDangNhap("Đăng nhập");
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public FormDangNhap(String title) {
        super(title);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JTabbedPane tabbedPane = new
JTabbedPane(JTabbedPane.TOP);
        tabbedPane.setBounds(0, 0, 434, 261);
        contentPane.add(tabbedPane);

        JPanel panel = new JPanel();
```

```

tabbedPane.addTab("Đăng kí", null, panel, null);
panel.setLayout(null);

JLabel lblNewLabel = new JLabel("Username");
lblNewLabel.setBounds(10, 11, 65, 14);
panel.add(lblNewLabel);

txtUsername = new JTextField();
txtUsername.setBounds(81, 8, 86, 20);
panel.add(txtUsername);
txtUsername.setColumns(10);

JLabel lblNewLabel_1 = new JLabel("Password");
lblNewLabel_1.setBounds(222, 11, 65, 14);
panel.add(lblNewLabel_1);

txtPassword = new JPasswordField();
txtPassword.setBounds(297, 8, 122, 20);
panel.add(txtPassword);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 66, 409, 156);
panel.add(scrollPane);

tblAccount = new JTable();
tblAccount.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
        "STT", "Username"
    }
));
scrollPane.setViewportViewView(tblAccount);

JButton btnAddAccount = new JButton("Add account");

btnAddAccount.setBounds(153, 36, 115, 23);
panel.add(btnAddAccount);

JPanel panel_1 = new JPanel();
tabbedPane.addTab("Đăng nhập", null, panel_1, null);
panel_1.setLayout(null);

txtTenDN = new JTextField();
txtTenDN.setBounds(195, 99, 100, 20);
panel_1.add(txtTenDN);
txtTenDN.setColumns(10);

JLabel lblNewLabel_2 = new JLabel("Username");
lblNewLabel_2.setBounds(107, 102, 100, 14);
panel_1.add(lblNewLabel_2);

```

```

JLabel lblPassword = new JLabel("Password");
lblPassword.setBounds(107, 140, 100, 14);
panel_1.add(lblPassword);

JButton btnDangNhap = new JButton("Đăng nhập");

btnDangNhap.setBounds(195, 176, 100, 23);
panel_1.add(btnDangNhap);

txtMatKhai = new JPasswordField();
txtMatKhai.setBounds(195, 137, 100, 23);
panel_1.add(txtMatKhai);

JLabel lblMnHnhng = new JLabel("MÀN HÌNH ĐĂNG NHẬP");
lblMnHnhng.setForeground(Color.RED);
lblMnHnhng.setFont(new Font("Tahoma", Font.BOLD, 13));
lblMnHnhng.setBounds(123, 11, 158, 62);
panel_1.add(lblMnHnhng);
btnAddAccount.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        modelAccount=(DefaultTableModel)
tblAccount.getModel();
        String username=txtUsername.getText();
        String password=txtPassword.getText();

        for(TaiKhoan data: listTK) {
            if(username.equals(data.getUsername())) {
                JOptionPane.showMessageDialog(null,
"Trùng username!");

                return;
            }
        }
        TaiKhoan taiKhoan=new TaiKhoan(username,
password);

        modelAccount.addRow(new Object[] {
            i, username
        });
        i++;
        listTK.add(taiKhoan);
    }
});
btnDangNhap.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        String tenDN=txtTenDN.getText();
        String matKhai=txtMatKhai.getText();
        TaiKhoan tk=new TaiKhoan(tenDN, matKhai);
        if(listTK.size()==0) {
            JOptionPane.showMessageDialog(null, "Danh
sách tài khoản chưa tồn tại!");

```



```

        return;
    }
    for(int i=0;i<listTK.size();i++) {

        if(tk.getUsername().equals(listTK.get(i).getUsername()) &&
tk.getPassword().equals(listTK.get(i).getPassword())) {
            JOptionPane.showMessageDialog(null,
"Chào mừng "+ tenDN +" đăng nhập thành công!", "Thông báo",
JOptionPane.INFORMATION_MESSAGE);
            return;
        }
    }
    JOptionPane.showMessageDialog(null, "Đăng
nhập thất bại", "Thông báo", JOptionPane.ERROR_MESSAGE);

}

});
}
}

```

Bài 13:

JlistSo.java

```

public class JListSo extends JFrame {

    private JPanel contentPane;
    private JTextField txtSo;
    DefaultListModel<Integer> modelSo=new DefaultListModel<>();
    ArrayList<Integer> list=new ArrayList<>();

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    JFrame frame = new JListSo("Thao tác số
trên JList");

                    frame.setVisible(true);
                    frame.setLocationRelativeTo(null);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**

```

```

    * Create the frame.
    */
    public JListSo(String title) {
        super(title);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 552, 344);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JPanel panel = new JPanel();
        panel.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Ch\u1ECDn
t\u00E1c v\u1EE5", TitledBorder.LEADING, TitledBorder.TOP, null, new
Color(0, 0, 0)));
        panel.setBounds(10, 11, 185, 195);
        contentPane.add(panel);
        panel.setLayout(null);

        JButton btnSoChan = new JButton("Tìm số chẵn");

        btnSoChan.setBounds(10, 22, 165, 23);
        panel.add(btnSoChan);

        JButton btnSoLe = new JButton("Tìm số lẻ");

        btnSoLe.setBounds(10, 56, 165, 23);
        panel.add(btnSoLe);

        JButton btnBanDau = new JButton("List ban đầu");

        btnBanDau.setBounds(10, 125, 165, 23);
        panel.add(btnBanDau);

        JButton btnChinhPhuong = new JButton("Tìm số chính
phương");

        btnChinhPhuong.setBounds(10, 91, 165, 23);
        panel.add(btnChinhPhuong);

        JButton btnTong = new JButton("Tổng giá trị");

        btnTong.setBounds(10, 159, 165, 23);
        panel.add(btnTong);

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new TitledBorder(null, "Nh\u1EADp
th\u00F4ng tin", TitledBorder.LEADING, TitledBorder.TOP, null,
null));
        panel_1.setBounds(206, 11, 320, 239);

```

```

contentPane.add(panel_1);
panel_1.setLayout(null);

txtSo = new JTextField();
txtSo.setBounds(10, 21, 86, 20);
panel_1.add(txtSo);
txtSo.setColumns(10);

JButton btnNhap = new JButton("Nhap");

btnNhap.setBounds(101, 20, 89, 20);
panel_1.add(btnNhap);

JCheckBox chkSoAm = new JCheckBox("Cho nhập số âm");
chkSoAm.setBounds(196, 20, 118, 23);
panel_1.add(chkSoAm);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 52, 300, 176);
panel_1.add(scrollPane);

JList listSo = new JList();
scrollPane.setViewportViewView(listSo);
btnNhap.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            int so=Integer.parseInt(txtSo.getText());
            if(!chkSoAm.isSelected()) {
                if(so<0) {

JOptionPane.showMessageDialog(null, "Không nhập số âm", "Cảnh
báo", JOptionPane.OK_OPTION);

                return;
            }
            modelSo.addElement(so);
            listSo.setModel(modelSo);
            list.add(so);
        }
        else {
            modelSo.addElement(so);
            listSo.setModel(modelSo);
            list.add(so);
        }
    }
    catch(NumberFormatException ex) {
        JOptionPane.showMessageDialog(null, "Chỉ
nhập số nguyên", "Cảnh báo", JOptionPane.OK_OPTION);
    }
}

```

```

    }
    });
    JButton btnExit = new JButton("Thoát chương trình");
    btnExit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            int a=JOptionPane.showConfirmDialog(null, "Bạn
có muốn thoát chương trình?", "Xác nhận",
JOptionPane.YES_NO_OPTION);

            if(a==JOptionPane.YES_OPTION)
                dispose();
            else
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        }

    });
    btnExit.setBounds(189, 261, 147, 23);
    contentPane.add(btnExit);
    btnTong.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            int tong=0;
            for(int data:list) {
                tong+=data;
            }
            JOptionPane.showMessageDialog(null, tong, "Kết
quả", JOptionPane.INFORMATION_MESSAGE);
        }

    });
    btnSoChan.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            modelSo.clear();
            for(int data:list) {
                if(data%2==0 && !modelSo.contains(data))
                {

                    modelSo.addElement(data);

                }
            }
            listSo.setForeground(Color.BLUE);
            listSo.setModel(modelSo);
        }
    });
    btnSoLe.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

```

```

        modelSo.clear();
        for(int data:list) {
            if(data%2!=0 && !modelSo.contains(data))

                modelSo.addElement(data);

        }
    }
    listSo.setForeground(Color.BLUE);
    listSo.setModel(modelSo);
}

});
btnBanDau.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        modelSo.clear();
        for(int data:list) {
            modelSo.addElement(data);
        }
        listSo.setModel(modelSo);
    }
});
btnChinhPhuong.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        modelSo.clear();
        for(int data:list) {
            int x=(int) Math.sqrt(data);
            if(x*x==data && !modelSo.contains(data))
            {
                modelSo.addElement(data);
            }
        }
        listSo.setModel(modelSo);
    }
});
}

}

```

#### Bài 14:

**Yêu cầu:** Quản lý thông tin nhân viên của từng phòng ban

**Chức năng:** chọn phòng ban từ combobox, thông tin nhân viên của từng phòng ban hiển thị vào list, các nút thêm: thêm 1 nhân viên; lưu: dùng cập nhật 1 nhân viên, xoá: xoá 1 nhân viên, thoát: thoát chương trình, ghi file: ghi danh sách nhân viên của phòng ban được chọn.

```
package baitapgiaodien.model;
```

```
NhanVien.java
```

```
public class NhanVien implements Serializable {
    private String maNhanVien;
    private String tenNhanVien;
    private Date namSinh;
    private Date ngayVaoLamViec;
    private PhongBan phong;

    public PhongBan getPhong() {
        return phong;
    }
    public void setPhong(PhongBan phong) {
        this.phong = phong;
    }
    public String getMaNhanVien() {
        return maNhanVien;
    }
    public void setMaNhanVien(String maNhanVien) {
        this.maNhanVien = maNhanVien;
    }
    public String getTenNhanVien() {
        return tenNhanVien;
    }
    public void setTenNhanVien(String tenNhanVien) {
        this.tenNhanVien = tenNhanVien;
    }
    public Date getNamSinh() {
```

```

        return namSinh;
    }
    public void setNamSinh(Date namSinh) {
        this.namSinh = namSinh;
    }
    public Date getNgayVaoLamViec() {
        return ngayVaoLamViec;
    }
    public void setNgayVaoLamViec(Date ngayVaoLamViec) {
        this.ngayVaoLamViec = ngayVaoLamViec;
    }
    public NhanVien(String maNhanVien, String tenNhanVien, Date
namSinh, Date ngayVaoLamViec) {
        super();
        this.maNhanVien = maNhanVien;
        this.tenNhanVien = tenNhanVien;
        this.namSinh = namSinh;
        this.ngayVaoLamViec = ngayVaoLamViec;
    }
    @Override
    public String toString() {
        return this.getTenNhanVien();
    }
}

```

PhongBan.java

```

public class PhongBan implements Serializable {
    private String maPhong;
    private String tenPhong;
    Vector<NhanVien> listNhanVien;
    public Vector<NhanVien> getListNhanVien() {
        return listNhanVien;
    }
    public void setListNhanVien(Vector<NhanVien> listNhanVien) {
        this.listNhanVien = listNhanVien;
    }
    public PhongBan() {
        super();
        this.listNhanVien=new Vector<>();
    }
    public PhongBan(String maPhong, String tenPhong) {
        super();
        this.maPhong = maPhong;
        this.tenPhong = tenPhong;
        this.listNhanVien=new Vector<>();
    }
    public String getMaPhong() {
        return maPhong;
    }
    public void setMaPhong(String maPhong) {
        this.maPhong = maPhong;
    }
}

```

```

    }
    public String getTenPhong() {
        return tenPhong;
    }
    public void setTenPhong(String tenPhong) {
        this.tenPhong = tenPhong;
    }
    public void themNhanVien(NhanVien nhanVien) {
        this.listNhanVien.add(nhanVien);
    }
    @Override
    public String toString() {
        // TODO Auto-generated method stub
        return this.getTenPhong();
    }
}

```

Giao diện: QLNhanVien.java

```

public class QLNhanVien extends JFrame {

    private JPanel contentPane;
    private JTextField txtMaNV;
    private JTextField txtTenNV;
    private JTextField txtNamSinh;
    private JTextField txtNgayVaoLam;

    PhongBan phongBanselected;
    NhanVien nhanVienselected;
    ArrayList<PhongBan> dsPB=new ArrayList<PhongBan>();
    private ArrayList<NhanVien> dsNV=new ArrayList<NhanVien>();
    SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    QLNhanVien frame = new QLNhanVien("Quản
lý nhân viên");

                    frame.setVisible(true);
                    frame.setLocationRelativeTo(null);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**

```



```

    * Create the frame.
    */
    public QLNhanVien(String name) {
        super(name);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 408);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Chọn phòng ban:");
        lblNewLabel.setBounds(74, 11, 110, 14);
        contentPane.add(lblNewLabel);

        JComboBox<PhongBan> cboPhongBan = new
JComboBox<PhongBan>();

        cboPhongBan.setBounds(194, 8, 128, 20);
        contentPane.add(cboPhongBan);

        JPanel panel = new JPanel();
        panel.setBounds(0, 46, 182, 244);
        contentPane.add(panel);
        panel.setForeground(Color.BLACK);
        panel.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Danh
sách", TitledBorder.LEADING, TitledBorder.TOP, null, Color.BLUE));
        panel.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 24, 162, 209);
        panel.add(scrollPane);

        JList listNV = new JList();
        listNV.addListSelectionListener(new
ListSelectionListener() {
            public void valueChanged(ListSelectionEvent arg0) {
                if(listNV.getSelectedIndex() == -1) return;
                nhanVienselected = (NhanVien)
listNV.getSelectedValue();

                txtMaNV.setText(nhanVienselected.getMaNhanVien());

                txtTenNV.setText(nhanVienselected.getTenNhanVien());

                txtNamSinh.setText(sdf.format(nhanVienselected.getNamSinh()));

                txtNgayVaoLam.setText(sdf.format(nhanVienselected.getNgayVaoLam
Viec()));
            }
        }
    }

```

```

});
scrollPane.setViewportViewView(listNV);

JScrollPane scrollPane_1 = new JScrollPane();
scrollPane_1.setBounds(204, 46, 220, 244);
contentPane.add(scrollPane_1);

JPanel panel_1 = new JPanel();
panel_1.setBorder(new
TitledBorder(UIManager.getBorder("TitledBorder.border"), "Th\u00F4ng
tin chi ti\u1EBFt", TitledBorder.LEADING, TitledBorder.TOP, null,
Color.BLUE));
scrollPane_1.setViewportViewView(panel_1);
panel_1.setLayout(null);

JLabel lblNewLabel_1 = new JLabel("Mã nhân viên");
lblNewLabel_1.setBounds(67, 23, 86, 14);
panel_1.add(lblNewLabel_1);

txtMaNV = new JTextField();
txtMaNV.setBounds(32, 48, 148, 20);
panel_1.add(txtMaNV);
txtMaNV.setColumns(10);

JLabel lblTen = new JLabel("Tên nhân viên");
lblTen.setBounds(67, 79, 86, 14);
panel_1.add(lblTen);

txtTenNV = new JTextField();
txtTenNV.setColumns(10);
txtTenNV.setBounds(32, 101, 148, 20);
panel_1.add(txtTenNV);

JLabel lblNmSinh = new JLabel("Năm sinh");
lblNmSinh.setBounds(67, 132, 86, 14);
panel_1.add(lblNmSinh);

txtNamSinh = new JTextField();
txtNamSinh.setColumns(10);
txtNamSinh.setBounds(32, 157, 148, 20);
panel_1.add(txtNamSinh);

txtNgayVaoLam = new JTextField();
txtNgayVaoLam.setColumns(10);
txtNgayVaoLam.setBounds(32, 211, 148, 20);
panel_1.add(txtNgayVaoLam);

JLabel lblNgyVoLm = new JLabel("Ngày vào làm");
lblNgyVoLm.setBounds(67, 186, 86, 14);
panel_1.add(lblNgyVoLm);

```

```

        JButton btnThem = new JButton("Thêm");

        btnThem.setBounds(214, 301, 89, 23);
        contentPane.add(btnThem);

        JButton btnLuu = new JButton("Lưu");

        btnLuu.setBounds(325, 301, 89, 23);
        contentPane.add(btnLuu);

        JButton btnXoa = new JButton("Xóa");

        btnXoa.setBounds(215, 334, 89, 23);
        contentPane.add(btnXoa);

        JButton btnThoat = new JButton("Thoát");
        btnThoat.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                int a=JOptionPane.showConfirmDialog(null, "Bạn
có muốn thoát chương trình?", "Xác nhận",
JOptionPane.YES_NO_OPTION);

                if(a==JOptionPane.YES_OPTION)
                    dispose();
                else
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            }
        });
        btnThoat.setBounds(326, 334, 89, 23);
        contentPane.add(btnThoat);

        JButton btnGhiFile = new JButton("Ghi file");
        btnGhiFile.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    FileOutputStream fos=new
FileOutputStream("dsnv"+ phongBanselected.getMaPhong() +".dat");
                    ObjectOutputStream oos=new
ObjectOutputStream(fos);
                    if(phongBanselected==null) {
                        JOptionPane.showMessageDialog(null,
"Chọn phòng ban cần ghi file");
                        return;
                    }

                    oos.writeObject(phongBanselected.getListNhanVien());
                    oos.close();
                    fos.close();
                    JOptionPane.showMessageDialog(null, "Ghi
thành công");
                }
            }
        });
    }
}

```

```

        catch(Exception ex) {
            JOptionPane.showMessageDialog(null,
"Lỗi");
        }
    });
    btnGhiFile.setBounds(42, 318, 89, 23);
    contentPane.add(btnGhiFile);

    PhongBan ptc=new PhongBan("ptc","Phòng tổ chức");
    PhongBan pkt=new PhongBan("pkt","Phòng kỹ thuật");
    PhongBan phc=new PhongBan("phc","Phòng hành chính");
    dsPB.add(ptc);
    dsPB.add(pkt);
    dsPB.add(phc);

    ptc.themNhanVien(new NhanVien("NV01", "Nguyễn Văn A", new
Date(1998-1900, 01, 01), new Date(2018-1900, 01, 01)));
    ptc.themNhanVien(new NhanVien("NV02", "Nguyễn Văn B", new
Date(1997-1900, 01, 01), new Date(2018-1900, 02, 14)));
    ptc.themNhanVien(new NhanVien("NV03", "Nguyễn Thị B", new
Date(1997-1900, 01, 05), new Date(2018-1900, 02, 05)));

    pkt.themNhanVien(new NhanVien("NV04", "Nguyễn Văn C", new
Date(1995-1900, 10, 10), new Date(2018-1900, 02, 01)));
    pkt.themNhanVien(new NhanVien("NV05", "Nguyễn Văn D", new
Date(1995-1900, 11, 10), new Date(2019-1900, 02, 06)));

    phc.themNhanVien(new NhanVien("NV06", "Nguyễn Thị C", new
Date(1996-1900, 03, 10), new Date(2018-1900, 02, 21)));
    for(PhongBan phong:dsPB) {
        cboPhongBan.addItem(phong);
    }
    cboPhongBan.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            if(cboPhongBan.getSelectedIndex()==-1) return;
            phongBanselected =(PhongBan)
cboPhongBan.getSelectedItem();

            listNV.setListData(phongBanselected.getListNhanVien());
        }
    });

    btnThem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            try {
                NhanVien nhanVien=new
NhanVien(txtMaNV.getText(), txtTenNV.getText(),
sdf.parse(txtNamSinh.getText()),
sdf.parse(txtNgayVaoLam.getText()));

```

```

        if(phongBanselected==null) {
            JOptionPane.showMessageDialog(null,
"Phải chọn phòng ban");
            return;
        }
        for(NhanVien
nv:phongBanselected.getListNhanVien()) {

            if(nhanVien.getMaNhanVien().equalsIgnoreCase(nv.getMaNhanVien()
)) {

                JOptionPane.showMessageDialog(null, "Trùng mã nhân viên ở phòng
" + phongBanselected.getTenPhong());
                return;
            }
        }
        phongBanselected.themNhanVien(nhanVien);

        listNV.setListData(phongBanselected.getListNhanVien());
        } catch (ParseException e) {
            JOptionPane.showMessageDialog(null,
e.toString());
        }
    });
    btnLuu.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {

                if(phongBanselected==null) {
                    JOptionPane.showMessageDialog(null,
"Phải chọn phòng ban");
                }
                else if(nhanVienselected==null) {
                    JOptionPane.showMessageDialog(null,
"Phải chọn nhân viên cần chỉnh sửa");
                }
                else {

                    nhanVienselected.setMaNhanVien(txtMaNV.getText());

                    nhanVienselected.setTenNhanVien(txtTenNV.getText());

                    nhanVienselected.setNgayVaoLamViec(sdf.parse(txtNgayVaoLam.getT
ext()));

                    nhanVienselected.setNamSinh(sdf.parse(txtNamSinh.getText()));

                    listNV.setListData(phongBanselected.getListNhanVien());
                }
            } catch (ParseException e1) {

```

```

JOptionPane.showMessageDialog(null,
e1.toString());
    }
    });

    btnXoa.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(nhanVienselected!=null) {

phongBanselected.getListNhanVien().remove(nhanVienselected);

listNV.setListData(phongBanselected.getListNhanVien());
                txtMaNV.setText("");
                txtNamSinh.setText("");
                txtNgayVaolam.setText("");
                txtTenNV.setText("");
            }
        }
    });
}

```

**Bài 15:**

**Yêu cầu:**

Quản lý sinh viên: bên trái là cây thư mục để quản lý, bên phải là 1 table chứa thông tin sinh viên, dưới là các nút chức năng

Chức năng: khi chọn 1 lớp bất kì bên cây thư mục, dữ liệu sinh viên sẽ hiển thị lên table. Các nút thêm, xoá, sửa 1 nhân viên, nút lưu để ghi danh sách xuống file, nút thoát để kết thúc chương trình.

**Quản lý sinh viên**

Mã	Họ Tên	Ngày sinh	Giới tính
1624801040051	Huynh Thanh T...	Mon Sep 28 00:...	Nam
1624801040051	Huynh Thanh T...	Mon Sep 28 00:...	Nữ

**Thông tin sinh viên**

Mã sinh viên: 1624801040051      Tên sinh viên: nh Thanh Tam

Ngày sinh: 28/09/1998      Giới tính: ☐ Nam

Thêm   Xoá   Sửa   Lưu   Thoát

```
package baitapgiaodien.model;
```

```
SinhVien.java
```

```
public class SinhVien implements Serializable{
    private String maSinhVien;
    private String tenSinhVien;
    private Date ngaySinh;
    private boolean gioiTinh;
    private Lop lop;

    public Lop getLop() {
        return lop;
    }
    public void setLop(Lop lop) {
        this.lop = lop;
    }
    public String getMaSinhVien() {
        return maSinhVien;
    }
    public void setMaSinhVien(String maSinhVien) {
        this.maSinhVien = maSinhVien;
    }
    public String getTenSinhVien() {
        return tenSinhVien;
    }
    public void setTenSinhVien(String tenSinhVien) {
        this.tenSinhVien = tenSinhVien;
    }
    public Date getNgaySinh() {
        return ngaySinh;
    }
    public void setNgaySinh(Date ngaySinh) {
        this.ngaySinh = ngaySinh;
    }
    public boolean isGioiTinh() {
        return gioiTinh;
    }
    public void setGioiTinh(boolean gioiTinh) {
        this.gioiTinh = gioiTinh;
    }
    public SinhVien(String maSinhVien, String tenSinhVien, Date
ngaySinh, boolean gioiTinh) {
        super();
        this.maSinhVien = maSinhVien;
        this.tenSinhVien = tenSinhVien;
        this.ngaySinh = ngaySinh;
        this.gioiTinh = gioiTinh;
    }
    public SinhVien() {
        super();
    }
}
```

```

        @Override
        public String toString() {
            return this.tenSinhVien;
        }
    }
}

```

Lop.java

```

public class Lop implements Serializable {
    private String maLop;
    private String tenLop;
    private ArrayList<SinhVien> listSinhVien=new ArrayList<>();
    private Khoa khoa;
    public Khoa getKhoa() {
        return khoa;
    }
    public void setKhoa(Khoa khoa) {
        this.khoa = khoa;
    }
    public ArrayList<SinhVien> getListSinhVien() {
        return listSinhVien;
    }
    public void setListSinhVien(ArrayList<SinhVien> listSinhVien) {
        this.listSinhVien = listSinhVien;
    }
    public String getMaLop() {
        return maLop;
    }
    public void setMaLop(String maLop) {
        this.maLop = maLop;
    }
    public String getTenLop() {
        return tenLop;
    }
    public void setTenLop(String tenLop) {
        this.tenLop = tenLop;
    }
    public Lop(String maLop, String tenLop) {
        super();
        this.maLop = maLop;
        this.tenLop = tenLop;
    }
    public Lop() {
        super();
    }
    @Override
    public String toString() {
        return this.tenLop;
    }
    public void themSinhVien(SinhVien sinhVien) {
        this.listSinhVien.add(sinhVien);
        sinhVien.setLop(this);
    }
}

```



```

    }

}

Khoa.java

public class Khoa implements Serializable{
    private String maKhoa;
    private String tenKhoa;
    private ArrayList<Lop> listLop=new ArrayList<>();

    public ArrayList<Lop> getListLop() {
        return listLop;
    }
    public void setListLop(ArrayList<Lop> listLop) {
        this.listLop = listLop;
    }
    public String getMaKhoa() {
        return maKhoa;
    }
    public void setMaKhoa(String maKhoa) {
        this.maKhoa = maKhoa;
    }
    public String getTenKhoa() {
        return tenKhoa;
    }
    public void setTenKhoa(String tenKhoa) {
        this.tenKhoa = tenKhoa;
    }
    public Khoa(String maKhoa, String tenKhoa) {
        super();
        this.maKhoa = maKhoa;
        this.tenKhoa = tenKhoa;
    }
    public Khoa() {
        super();
    }
    public void themLop(Lop lop) {
        this.listLop.add(lop);
        lop.setKhoa(this);
    }
    @Override
    public String toString() {
        return this.tenKhoa;
    }
}

```

DocGhiFile.java

```

public class DocGhiFile {
    public static boolean ghiFile(ArrayList<Khoa> listKhoa, String
name) {

```

```

        try {
            FileOutputStream fileOutputStream=new
FileOutputStream(name);
            ObjectOutputStream objectOutputStream=new
ObjectOutputStream(fileOutputStream);
            objectOutputStream.writeObject(listKhoa);
            objectOutputStream.close();
            fileOutputStream.close();
            return true;
        } catch (Exception e) {
        }
        return false;
    }
    public static ArrayList<Khoa> docFile(String name){
        ArrayList<Khoa> listKhoa=new ArrayList<>();
        try {
            FileInputStream fileInputStream=new
FileInputStream(name);
            ObjectInputStream inputStream=new
ObjectInputStream(fileInputStream);
            listKhoa=(ArrayList<Khoa>) inputStream.readObject();
            inputStream.close();
            fileInputStream.close();
        } catch (Exception e) {
        }
        return listKhoa;
    }
}

```

Giao diện: QLSV.java

```

public class QLSV extends JFrame {

    private JPanel contentPane;
    private JTable tblLop;
    private JTextField txtMaSV;
    private JTextField txtNgaySinh;
    private JTextField txtTenSV;
    private static final String NAME = "qlsv.txt";
    int posselected;
    Button btnThem;

    JTree tree;
    ArrayList<Khoa> listKhoa;
    Lop lopSelected;
    DefaultTableModel modelLop;
    SinhVien svSelected;
    SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
    DefaultMutableTreeNode root=new DefaultMutableTreeNode("Đại học
Thủ Dầu Một");
    /**
     * Launch the application.

```

```

        */
        public static void main(String[] args) {
            EventQueue.invokeLater(new Runnable() {
                public void run() {
                    try {
                        QLSV frame = new QLSV("Quản lý sinh
viên");

                        frame.setVisible(true);
                        frame.setLocationRelativeTo(null);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            });
        }

        /**
         * Create the frame.
         */
        public QLSV(String title) {
            super(title);
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setBounds(100, 100, 604, 366);
            contentPane = new JPanel();
            contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
            setContentPane(contentPane);
            contentPane.setLayout(null);

            JScrollPane scrollPane = new JScrollPane();

            scrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);
            scrollPane.setBounds(10, 11, 151, 305);
            contentPane.add(scrollPane);

            tree = new JTree();

            tree.setModel(new DefaultTreeModel(root));
            scrollPane.setViewportView(tree);

            JScrollPane scrollPane_1 = new JScrollPane();
            scrollPane_1.setBounds(171, 11, 407, 184);
            contentPane.add(scrollPane_1);

            tblLop = new JTable();

            tblLop.setModel(new DefaultTableModel(
                new Object[][] {
                },
                new String[] {

```

```

        "M\u00E3", "H\u1ECD T\u00EAn", " Ng\u00E0y
sinh", "Gi\u1EDBi t\u00EDnh"
    });
    scrollPane_1.setViewportViewView(tblLop);

    JPanel panel = new JPanel();
    panel.setBorder(new TitledBorder(null, "Th\u00F4ng tin
sinh vi\u00EAn", TitledBorder.LEADING, TitledBorder.TOP, null,
null));
    panel.setBounds(171, 206, 407, 70);
    contentPane.add(panel);
    panel.setLayout(null);

    JLabel lblMSinhVin = new JLabel("Mã sinh viên");
    lblMSinhVin.setBounds(10, 19, 70, 14);
    panel.add(lblMSinhVin);

    txtMaSV = new JTextField();
    txtMaSV.setBounds(90, 16, 86, 20);
    panel.add(txtMaSV);
    txtMaSV.setColumns(10);

    JLabel lblNgaySinh = new JLabel("Ngày sinh");
    lblNgaySinh.setBounds(10, 42, 70, 14);
    panel.add(lblNgaySinh);

    txtNgaySinh = new JTextField();
    txtNgaySinh.setColumns(10);
    txtNgaySinh.setBounds(90, 39, 86, 20);
    panel.add(txtNgaySinh);

    JLabel lblTnSinhVin = new JLabel("Tên sinh viên");
    lblTnSinhVin.setBounds(211, 16, 80, 14);
    panel.add(lblTnSinhVin);

    JLabel lblGiiTnh = new JLabel("Giới tính");
    lblGiiTnh.setBounds(211, 42, 56, 14);
    panel.add(lblGiiTnh);
    ButtonGroup groupGioiTinh=new ButtonGroup();

    txtTenSV = new JTextField();
    txtTenSV.setBounds(294, 16, 86, 20);
    panel.add(txtTenSV);
    txtTenSV.setColumns(10);

    JCheckBox chkNam = new JCheckBox("Nam");
    chkNam.setBounds(283, 38, 97, 23);
    panel.add(chkNam);

    JButton btnThem = new JButton("Thêm");

```

```

        btnThem.setBounds(209, 287, 68, 23);
        contentPane.add(btnThem);

        JButton btnXoa = new JButton("Xóa");

        btnXoa.setBounds(287, 287, 59, 23);
        contentPane.add(btnXoa);

        JButton btnSua = new JButton("Sửa");

        btnSua.setBounds(356, 287, 59, 23);
        contentPane.add(btnSua);

        JButton btnLuu = new JButton("Lưu");

        btnLuu.setBounds(425, 287, 59, 23);
        contentPane.add(btnLuu);

        JButton btnThoat = new JButton("Thoát");
        btnThoat.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                dispose();
            }
        });
        btnThoat.setBounds(494, 287, 84, 23);
        contentPane.add(btnThoat);

        loadData();
        tree.addTreeSelectionListener(new TreeSelectionListener() {

            @Override
            public void valueChanged(TreeSelectionEvent e) {
                DefaultMutableTreeNode
node=(DefaultMutableTreeNode) tree.getLastSelectedPathComponent();

                try {
                    lopSelected=(Lop) node.getUserObject();
                    showLop();
                }
                catch (Exception ex) {
                }
            }

        });
        btnThem.addActionListener(new ActionListener() {

```

```

@Override
public void actionPerformed(ActionEvent arg0) {
    if(lopSelected!=null) {
        try {

            SinhVien sinhVien=new
SinhVien(txtMaSV.getText(), txtTenSV.getText(),
sdf.parse(txtNgaySinh.getText()),chkNam.isSelected());
            lopSelected.themSinhVien(sinhVien);

            showLop();

        }catch (Exception e) {
            JOptionPane.showMessageDialog(null,
"Thêm thất bại");
        }

    }else {
        JOptionPane.showMessageDialog(null, "Chưa
chọn lớp cần thêm");
    }
}

});
btnXoa.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        int posselected=tblLop.getSelectedRow();

        if(posselected>=0) {

            lopSelected.getListSinhVien().remove(posselected);

        }
        txtMaSV.setText("");
        txtTenSV.setText("");
        txtNgaySinh.setText("");
        chkNam.setSelected(false);
        showLop();
    }
});
btnSua.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        posselected=tblLop.getSelectedRow();
        SinhVien
svselected=lopSelected.getListSinhVien().get(posselected);
        svselected.setMaSinhVien(txtMaSV.getText());
        svselected.setTenSinhVien(txtTenSV.getText());
        try {

            svselected.setNgaySinh(sdf.parse(txtNgaySinh.getText()));
        } catch (ParseException e1) {
            // TODO Auto-generated catch block

```

```

        e1.printStackTrace();
    }
    if(chkNam.isSelected())
        svselected.setGioiTinh(true);
    else svselected.setGioiTinh(false);
    showLop();
}
});
tblLop.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent arg0) {
        posselected=tblLop.getSelectedRow();
        SinhVien
svselected=lopSelected.getListSinhVien().get(posselected);
        txtMaSV.setText(svselected.getMaSinhVien());
        txtTenSV.setText(svselected.getTenSinhVien());

        txtNgaySinh.setText(sdf.format(svselected.getNgaySinh()));
        if(svselected.isGioiTinh())
chkNam.setSelected(true);
        else chkNam.setSelected(false);
    }
});
btnLuu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        boolean kq=DocGhiFile.ghiFile(listKhoa, NAME);
        if(kq==true) {
            JOptionPane.showMessageDialog(null, "Lưu
ok");
        }
        else JOptionPane.showMessageDialog(null, "Lưu
thất bại");
    }
});
}

```

```

private void loadData() {
    listKhoa=DocGhiFile.docFile(NAME);
    if(listKhoa.size()==0) {
        listKhoa =new ArrayList<>();
        Khoa khoaCNTT=new Khoa("CNTT", "Khoa Công nghệ thông
tin");
        khoaCNTT.themLop(new Lop("D4HT01", "Lớp hệ thống
D14HT01"));
        khoaCNTT.themLop(new Lop("D4PM01", "Lớp phần mềm
D14HT01"));
        Khoa khoaKT=new Khoa("KT", "Khoa Kinh tế");
        khoaKT.themLop(new Lop("D14QT01", "Lớp quản trị
D14QT01"));
    }
}

```

```

        khoaKT.themLop(new Lop("D14KT01", "Lớp kế toán
D14QT01"));

        listKhoa.add(khoaCNTT);
        listKhoa.add(khoaKT);

    }
    for(Khoa khoa:listKhoa) {
        DefaultMutableTreeNode nodeKhoa=new
DefaultMutableTreeNode(khoa);
        root.add(nodeKhoa);
        for(Lop lop:khoa.getListLop()) {
            DefaultMutableTreeNode nodeLop=new
DefaultMutableTreeNode(lop);
            nodeKhoa.add(nodeLop);
        }
    }

}

private void showLop() {
    modelLop=(DefaultTableModel) tblLop.getModel();
    modelLop.setRowCount(0);
    for(SinhVien sinhVien:lopSelected.getListSinhVien()) {

        modelLop.addRow(new Object[] {
            sinhVien.getMaSinhVien(),
sinhVien.getTenSinhVien(), sinhVien.getNgaySinh(),
sinhVien.isGioiTinh()?"Nam":"Nữ"
        });
    }

}

}

```