

# HƯỚNG DẪN ASSIGNMENT

## Java2 - MOB1023

Java

Giảng viên hướng dẫn: Cô Hồ Thị Hồng Nga

## 1. THIẾT KẾ GIAO DIỆN

Thiết kế giao diện như hình (Sử dụng Jframe trong NetBean)

Bao gồm các controler:

- Các Lable: QUẢN LÝ NHÂN VIÊN, Mã nhân viên, Họ tên, Tuổi, Email, Lương, Status, Timer.
- Các Text Field chứa thông tin: Mã nhân viên, Họ tên, Tuổi, Email, Lương
- Các Button: New, Save, Delete, Find, Open, Exit. Các nút điều hướng: First, Previous , Next, Last ( |< << >> >| )
- Table: chứa danh sách nhân viên.

Yêu cầu:

- Các điều kiện đặt tên (Variable Name) theo đúng qui ước để thuận tiện cho việc code các chức năng xử lý.
- Các chức năng xử lý nghiệp vụ tổ chức mã ở cuối file code của JFrame cho dễ quản lý.

## 2. TỔ CHỨC CLASS MODEL

Xây dựng lớp Employee có implements Serializable để có thể lưu vào file

```

+ /**...4 lines */
public class Employee implements Serializable {

    private String maNV;
    private String hoTenNV;
    private Integer tuoi;
    private String email;
    private Double luong;

    public Employee() {
    }

    public Employee(String maNV, String hoTenNV, Integer tuoi,
                    String email, Double luong) {
        this.maNV = maNV;
        this.hoTenNV = hoTenNV;
        this.tuoi = tuoi;
        this.email = email;
        this.luong = luong;
    }

```

Các phương thức **getter**

```

- public String getMaNV() {
    return maNV;
}

- public String getHoTenNV() {
    return hoTenNV;
}

- public Integer getTuoi() {
    return tuoi;
}

- public String getEmail() {
    return email;
}

- public Double getLuong() {
    return luong;
}

```

## Các phương thức **setter**

```

public void setMaNV(String maNV) {
    this.maNV = maNV;
}

public void setHoTenNV(String hoTenNV) {
    this.hoTenNV = hoTenNV;
}

public void setTuoi(Integer tuoi) {
    this.tuoi = tuoi;
}

public void setEmail(String email) {
    this.email = email;
}

public void setLuong(Double luong) {
    this.luong = luong;
}

```

### 3. CÁC YÊU CẦU VỀ XỬ LÝ CHỨC NĂNG

#### a. Class XFile xử lý đọc/ghi file

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

/**...4 lines */
public class XFile {
    public static Object readObject(String path) {
        Object object = null;
        try {
            FileInputStream fis = new FileInputStream(path);
            if (fis.available() > 0) {
                ObjectInputStream ois = new ObjectInputStream(fis);
                object = ois.readObject();
                ois.close();
                fis.close();
            }
        } catch (Exception e) {
            throw new RuntimeException(e);
        } finally {
            return object;
        }
    }

    public static void writeObject(String path, Object object) {
        try {
            FileOutputStream fos = new FileOutputStream(path);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(object);
            oos.close();
            fos.close();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

#### b. Các phương thức cơ bản xử lý nghiệp vụ

```
// Tạo các biến dùng chung
List<Employee> lstNV = new ArrayList<>();
DefaultTableModel model;
int index = -1;
String fileName = "femployee.dat";
```

```

+
-
/** Đổ dữ liệu từ danh sách lstNV lên table ...3 lines */
public void fillToTable() {
    model = (DefaultTableModel) tblListNV.getModel();
    model.setRowCount(0);
    List<Employee> lst = lstNV;
    for (Employee em : lst) {
        Vector row = new Vector();
        row.add(em.getMaNV());
        row.add(em.getHoTenNV());
        row.add(em.getTuoi());
        row.add(em.getEmail());
        row.add(em.getLuong());
        model.addRow(row);
    }
}

```

```

+
-
/** Hiển thị thông tin nhân viên của dòng được chọn trong ...4 lines */
public void showDetail() {
    List<Employee> lst = lstNV;
    Employee em = lst.get(index);
    txtManv.setText(em.getMaNV());
    txtHoTen.setText(em.getHoTenNV());
    txtTuoi.setText(em.getTuoi().toString());
    txtEmail.setText(em.getEmail());
    txtLuong.setText(em.getLuong().toString());
    fileHinh = em.getFileHinh();
    ImageIcon img = new ImageIcon(fileHinh);
    lblImage.setIcon(img);
    if (fileHinh != "") {
        lblImage.setText("");
    } else {
        lblImage.setText("Hình");
    }
}

```

```

+
-
/** Xóa nội dung các điều khiển textfield trên Form ...3 lines */
public void clearForm() {
    txtManv.setText("");
    txtHoTen.setText("");
    txtTuoi.setText("");
    txtEmail.setText("");
    txtLuong.setText("");
}

```

```

+
-
/** Lấy thông tin được nhập trên các điều khiển form để tạo nhân viên mới
public Employee takeInfo() {
    Employee em = new Employee();
    em.setMaNV(txtManv.getText());
    em.setHoTenNV(txtHoTen.getText());
    em.setTuoi(Integer.parseInt(txtTuoi.getText()));
    em.setEmail(txtEmail.getText());
    em.setLuong(Double.parseDouble(txtLuong.getText()));
    return em;
}

```

```

+
-
/** Di chuyển vị trí nhân viên trong danh sách ...3 lines */
private void goTo(int ind) {
    tblListNV.setRowSelectionInterval(ind, ind);
    showDetail();
    setStatus();
}

```

```

+
-
/** Lưu thông tin nhân viên Nếu nhân viên mới (index = -1) thì thêm mới
private void saveEmployee() {
    if (index == -1) {
        insert();
        index = lstNV.size() - 1;
    } else {
        update();
    }
    fillToTable();
    tblListNV.setRowSelectionInterval(index, index);
    setStatus();
}

```

```

+
-
/** Xóa nhân viên được chọn khỏi danh sách ...3 lines */
private void removeEmployee() {
    delete();
    fillToTable();
    clearForm();
    index = -1;
    tblListNV.clearSelection();
    setStatus();
    txtManv.requestFocus();
}

```

```

+
-
/** Tìm kiếm nhân viên theo mã NV do người dùng nhập vào ...3 lines */
private void findEmployee() {
    int i = find();
    if (i != -1) {
        index = i;
        showDetail();
        tblListNV.setRowSelectionInterval(index, index);
    } else {
        JOptionPane.showMessageDialog(this, "Không tìm thấy nhan vien");
    }
    setStatus();
}

```

```

+
-
/** Hiển thị trạng thái mẫu tin trên Form ...3 lines */
void setStatus() {
    List<Employee> lst = lstNV;
    int resc = 0;
    if (lst.size() > 0) {
        resc = index + 1;
    }
    lblStatus.setText("Record " + resc + " Of " + lst.size());
}

```

```

/** Hiển thị thời gian lên form ...3 lines */
public void runClock() {
    Thread t = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                try {
                    Date now = new Date();
                    SimpleDateFormat sdf = new SimpleDateFormat();
                    sdf.applyPattern("hh:mm aa");
                    String time = sdf.format(now);
                    lblTimer.setText(time);
                    Thread.sleep(60000);
                } catch (Exception e) {
                    break;
                }
            }
        }
    });
    t.start();
}

```

#### 4. CÁC CHỨC NĂNG GỢI Ý PHÁT TRIỂN THÊM.

- Thêm hình ảnh cho nhân viên.
- Thêm chức năng lọc dữ liệu và sắp xếp như giao diện gợi ý.

Quản Lý Nhân Viên

QUẢN LÝ NHÂN VIÊN

04:31 PM



MÃ NHÂN VIÊN

NV01

New

HỌ VÀ TÊN

Trần Thị Hạnh

Save

TUỔI

32

Delete

EMAIL

hanhtt@gmail.com

Find

LƯƠNG

2000000.0

Open

Add Image

<

<<

>>

>

Record 1 Of 5

Exit

Mã NV

Filter

Mã

Ascending

Sort

Mã	Họ Tên	Tuổi	Email	Lương
NV01	Trần Thị Hạnh	32	hanhtt@gmail.com	2000000.0
NV02	Trần Văn Tuấn	27	tuantv@gmail.com	5000000.0
NV03	Nguyễn Hồng Thi	22	thinh@gmail.com	1000000.0
NV04	Đoàn Công Hậu	38	haudct@gmail.com	8000000.0
NV05	Hoàng Thị Yến Oanh	25	oanhhty@gmail.com	3000000.0



Class Employee bổ sung thuộc tính fileHinh lưu đường dẫn tập tin hình ảnh

```
public class Employee implements Serializable {

    private String maNV;
    private String hoTenNV;
    private Integer tuoi;
    private String email;
    private Double luong;
    private String fileHinh;
```

```
+ /** Chọn hình ảnh, lưu đường dẫn và load ImageIcon ho lable hình ảnh ...4 l
- private void btnImageActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JFileChooser fc = new JFileChooser();
    if (fc.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
        try {
            fileHinh = fc.getCurrentDirectory().toString() + "\\\"
                        + fc.getSelectedFile().getName();
            BufferedImage img = ImageIO.read(fc.getSelectedFile());
            lblImage.setIcon(new ImageIcon(img));
        } catch (IOException ex) {
        }
    }
}
```

Thực hiện chức năng lọc dữ liệu

```
+ /** Thực hiện chức năng lọc dữ liệu ...4 lines */
- private void btnFilterActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (btnFilter.getText().equalsIgnoreCase("Filter")) {
        lstFilter = filter(cboFilter.getSelectedItem().toString(),
                           txtCriteria.getText());
        btnFilter.setText("ShowAll");
        filter = true;
    } else {
        btnFilter.setText("Filter");
        txtCriteria.setText("");
        filter = false;
    }
    fillToTable();
    index = 0;
    showDetail();
    setStatus();
    tblListNV.setRowSelectionInterval(index, index);
}
```

```

+
-
/** Lọc danh sách nhân viên theo tiêu chí chọn trên Form ...7 lines */
private List<Employee> filter(String selectedItem, String criteria) {

    List<Employee> lstFilt = new ArrayList<>();
    switch (selectedItem) {
        case "Mã NV":
            for (Employee nv : lstNV) {
                if (nv.getMaNV().equalsIgnoreCase(criteria)) {
                    lstFilt.add(nv);
                }
            }
            break;
        case "Họ tên":
            for (Employee nv : lstNV) {
                if (nv.getHoTenNV().contains(criteria)) {
                    lstFilt.add(nv);
                }
            }
            break;
        case "Tuổi từ":
            for (Employee nv : lstNV) {
                if (nv.getTuoi() >= (Integer.parseInt(criteria))) {
                    lstFilt.add(nv);
                }
            }
            break;
        case "Lương từ":
            for (Employee nv : lstNV) {
                if (nv.getLuong() >= (Integer.parseInt(criteria))) {
                    lstFilt.add(nv);
                }
            }
            break;
    }
    return lstFilt;
}

```

## Thực hiện chức năng sắp xếp

```

+ /** Thực hiện chức năng sắp xếp ...4 lines */
- private void btnSortActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sort(cboSort.getSelectedItem().toString(), chkAsc.isSelected());
    fillToTable();
    index = 0;
    tblListNV.setRowSelectionInterval(index, index);
    showDetail();
    setStatus();
}

```

```

+ /** Sắp xếp danh sách nhân viên theo tiêu chí chọn trên Form ...5 lines
- private void sort(String SelectCol, boolean selected) {
    Comparator<Employee> com = new Comparator<Employee>() {
        @Override
        public int compare(Employee o1, Employee o2) {
            switch (SelectCol) {
                case "Họ tên":
                    return o1.getHoTenNV().compareTo(o2.getHoTenNV());
                case "Tuổi":
                    return o1.getTuoi().compareTo(o2.getTuoi());
                case "Email":
                    return o1.getEmail().compareTo(o2.getEmail());
                case "Luong":
                    return o1.getLuong().compareTo(o2.getLuong());
            }
            return o1.getMaNV().compareTo(o2.getMaNV());
        }
    };
    List<Employee> lst = lstNV;
    if (filter) {
        lst = lstFilter;
    }
    Collections.sort(lst, com);
    if (!selected) {
        Collections.reverse(lst);
    }
}

```

Chúc các bạn làm bài tốt.