



FPT POLYTECHNIC



Conceive Design Implement Operate

QUẢN LÝ NHÂN VIÊN

GIẢNG VIÊN: NGUYỄN NGHIỆM

www.poly.edu.vn

- ⊙ Kết thúc bài học này bạn có khả năng
 - ⊙ Viết mã điều khiển cho cửa sổ NhanVienJDialog



EduSys - Quản lý nhân viên

QUẢN LÝ NHÂN VIÊN QUẢN TRỊ

CẬP NHẬT

DANH SÁCH

MÃ NV	MẬT KHẨU	HỌ VÀ TÊN	VAI TRÒ
NoPT	123456	Phạm Thị Nở	Nhân viên
PheoNC	123456	Nguyễn Chí Phèo	Nhân viên
TeoNV	songlong	Nguyễn Văn Tèo	Trưởng phòng

EduSys - Quản lý nhân viên

QUẢN LÝ NHÂN VIÊN QUẢN TRỊ

CẬP NHẬT

DANH SÁCH

Mã nhân viên

Mật khẩu

Xác nhận mật khẩu

Họ và tên

Vai trò

☐ Trưởng phòng
 ☒ Nhân viên

Thêm

Sửa

Xóa

Mới

|<

<<

>>

>|

❑ Form

- ❖ Đọc dữ liệu từ form và tạo NhanVien
- ❖ Hiển thị dữ liệu NhanVien lên form

❑ Bảng

- ❖ Đổ dữ liệu vào bảng
- ❖ Đọc lấy NhanVien của một hàng

❑ Quản lý

- ❖ Cập nhật dữ liệu (btnSua)
- ❖ Xóa dữ liệu (btnXoa)
- ❖ Xóa trắng form (btnMoi)
- ❖ Thêm mới vào CSDL (btnThem)

❑ Điều hướng

- ❖ btnFirst
- ❖ btnPrev
- ❖ btnNext
- ❖ btnLast

❑ Điều khiển trạng thái nút quản lý

MÔ HÌNH ỨNG DỤNG QUẢN LÝ NHÂN VIÊN

EduSys - Quản lý nhân viên

QUẢN LÝ NHÂN VIÊN QUẢN TRỊ

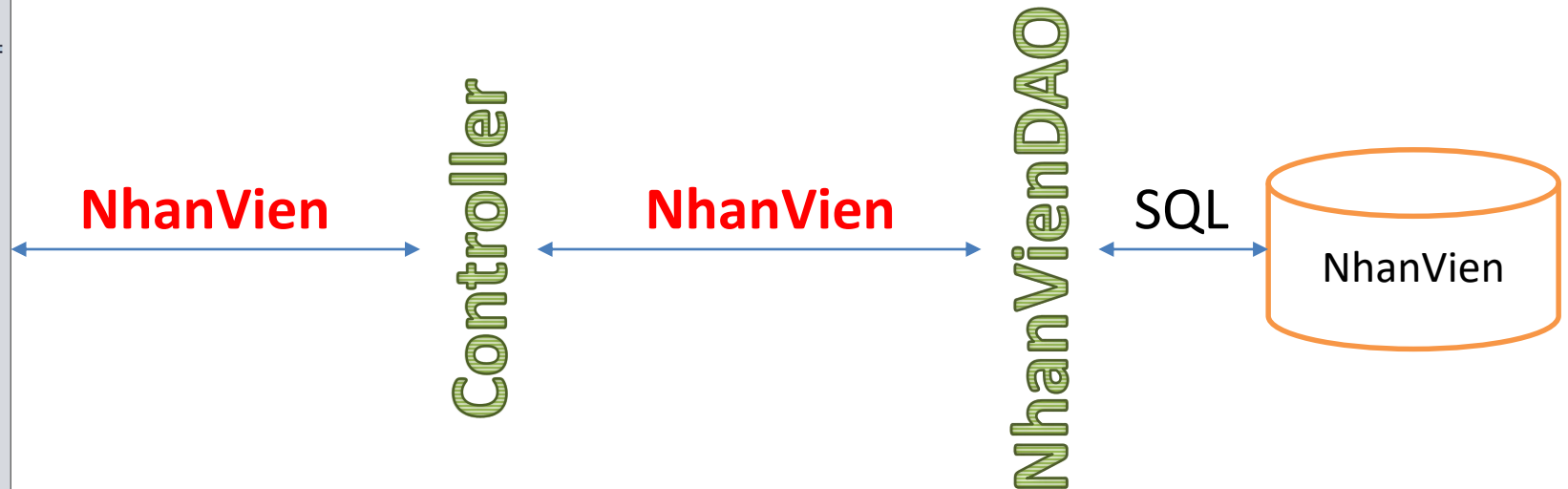
Mã nhân viên

Mật khẩu

Xác nhận mật khẩu

Họ và tên

Vai trò
☐ Trưởng phòng ☒ Nhân viên



Controller Code

```
NhanVienDAO dao = new NhanVienDAO(); // Làm việc với bảng NhanVien
int row = -1; // Hàng được chọn hiện tại trên bảng
void init(){...} // [NhanVienJDialog Open]
void insert(){...} // [btnThem]
void update(){...} // [btnSua]
void delete(){...} // [btnXoa]
void clearForm(){...} // [btnMoi]
void edit() {...} // [tblHocVien double click]
void first(){...} // [btnFirst]
void prev(){...} // [btnPrev]
void next(){...} // [btnNext]
void last(){...} // [btnLast]
void fillTable() {...} // đổ dữ liệu vào tblHocVien
void setForm(NhanVien nv){...} // hiển thị NhanVien lên form
NhanVien getForm(){...} // Tạo NhanVien từ form
void updateStatus(){...} // Cập nhật trạng thái các nút
```

ĐỔ DỮ LIỆU VÀO JTABLE - FILLTABLE()

```
void fillTable() {  
    DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();  
    model.setRowCount(0); // xóa tất cả các hàng trên JTable  
    try {  
        List<NhanVien> list = dao.selectAll(); // đọc dữ liệu từ CSDL  
        for (NhanVien nv : list) {  
            Object[] row = {nv.getMaNV(), nv.getMatKhau(), nv.getHoTen(),  
                            nv.getVaiTro() ? "Trưởng phòng" : "Nhân viên"  
            };  
            model.addRow(row); // thêm một hàng vào JTable  
        }  
    }  
    catch (Exception e) {  
        MsgBox.alert(this, "Lỗi truy vấn dữ liệu!");  
    }  
}
```

```
void init(){  
    setLocationRelativeTo(null);  
  
    ★ this.fillTable();  
    this.row = -1;  
    this.updateStatus();  
}
```

```
void setForm(NhanVien nv){
    txtMaNV.setText(nv.getMaNV());
    txtHoTen.setText(nv.getHoTen());
    txtMatKhau.setText(nv.getMatKhau());
    txtMatKhau2.setText(nv.getMatKhau());
    rdoTruongPhong.setSelected(nv.getVaiTro());
    rdoNhanVien.setSelected(!nv.getVaiTro());
}
```

```
NhanVien getForm(){
    NhanVien nv = new NhanVien();
    nv.setMaNV(txtMaNV.getText());
    nv.setHoTen(txtHoTen.getText());
    nv.setMatKhau(new String(txtMatKhau.getPassword()));
    nv.setVaiTro(rdoTruongPhong.isSelected());
    return nv;
}
```

```
void clearForm(){
    NhanVien nv = new NhanVien();
    ★ this.setForm(nv);
    this.row = -1;
    this.updateStatus();
}
```

[btnMoi]

```
void edit() {
    String manv = (String) tblNhanVien.getValueAt(this.row, 0);
    NhanVien nv = dao.selectById(manv);
    ★ this.setForm(nv);
    tabs.setSelectedIndex(0);
    this.updateStatus();
}
```

[tblNhanVien]

THÊM MỚI DỮ LIỆU - INSERT()

```
void insert(){
    NhanVien nv = getForm();
    String mk2 = new String(txtMatKhai2.getPassword());
    if(!mk2.equals(nv.getMatKhai())){
        MsgBox.alert(this, "Xác nhận mật khẩu không đúng!");
    }
    else{
        try {
            dao.insert(nv); this.fillTable(); this.clearForm();
            MsgBox.alert(this, "Thêm mới thành công!");
        }
        catch (Exception e) {
            MsgBox.alert(this, "Thêm mới thất bại!");
        }
    }
}
```

[btnThem]

❑ Insert thành công thì

- ❖ fillTable(): tải lại dữ liệu lên bảng
- ❖ clearForm(): làm sạch form

```
void update(){
    NhanVien nv = getForm();
    String mk2 = new String(txtMatKhau2.getPassword());
    if(!mk2.equals(nv.getMatKhau())){
        MsgBox.alert(this, "Xác nhận mật khẩu không đúng!");
    }
    else{
        try {
            dao.update(nv); this.fillTable();
            MsgBox.alert(this, "Cập nhật thành công!");
        }
        catch (Exception e) {
            MsgBox.alert(this, "Cập nhật thất bại!");
        }
    }
}
```

**[btnSua]**

- ❑ Update thành công thì
 - ❖ fillTable(): tải lại dữ liệu lên bảng

XÓA DỮ LIỆU - DELETE()

```
void delete(){
    if(!Auth.isManager()){
        MsgBox.alert(this, "Bạn không có quyền xóa nhân viên!");
    }
    else{
        String manv = txtMaNV.getText();
        if(manv.equals(Auth.user.getMaNV())){
            MsgBox.alert(this, "Bạn không được xóa chính bạn!");
        }
        else if(MsgBox.confirm(this, "Bạn thực sự muốn xóa nhân viên này?")){
            try {
                dao.delete(manv); this.fillTable(); this.clearForm();
                MsgBox.alert(this, "Xóa thành công!");
            }
            catch (Exception e) {
                MsgBox.alert(this, "Xóa thất bại!");
            }
        }
    }
}
```

[btnXoa]

- ☐ Trưởng phòng mới được xóa
- ☐ Không được xóa chính mình
- ☐ Delete thành công thì
 - ❖ fillTable(): tải lại dữ liệu lên bảng
 - ❖ clearForm(): làm sạch form

```
void first(){  
    this.row = 0;  
    this.edit();  
}
```

[btnFirst]

```
void prev(){  
    if(this.row > 0){  
        this.row--;  
        this.edit();  
    }  
}
```

[btnPrev]

```
void next(){  
    if(this.row < tblNhanVien.getRowCount() - 1){  
        this.row++;  
        this.edit();  
    }  
}
```

[btnNext]

```
void last(){  
    this.row = tblNhanVien.getRowCount() - 1;  
    this.edit();  
}
```

[btnLast]

```
void updateStatus(){
    boolean edit = (this.row >= 0);
    boolean first = (this.row == 0);
    boolean last = (this.row == tblNhanVien.getRowCount() - 1);
    // Trạng thái form
    txtMaNV.setEditable(!edit);
    btnThem.setEnabled(!edit);
    btnSua.setEnabled(edit);
    btnXoa.setEnabled(edit);
    // Trạng thái điều hướng
    btnFirst.setEnabled(edit && !first);
    btnPrev.setEnabled(edit && !first);
    btnNext.setEnabled(edit && !last);
    btnLast.setEnabled(edit && !last);
}
```

init()

clearForm()

edit()

- ❑ Trong các phương thức xử lý sự kiện chỉ việc gọi đến các phương thức tương ứng
- ❑ Cần chú ý đến sự kiện **double click** chuột vào Jtable được xử lý như sau:

```
private void tblNhanVienMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    if(evt.getClickCount() == 2){  
        this.row = tblNhanVien.getSelectedRow();  
        this.edit();  
    }  
}
```

```
NhanVienDAO dao = new NhanVienDAO(); // Làm việc với bảng NhanVien
int row = -1; // Hàng được chọn hiện tại trên bảng
void init(){...} // [NhanVienJDialog Open]
void insert(){...} // [btnThem]
void update(){...} // [btnSua]
void delete(){...} // [btnXoa]
void clearForm(){...} // [btnMoi]
void edit() {...} // [tblHocVien double click]
void first(){...} // [btnFirst]
void prev(){...} // [btnPrev]
void next(){...} // [btnNext]
void last(){...} // [btnLast]
void fillTable() {...} // đổ dữ liệu vào tblHocVien
void setForm(NhanVien nv){...} // hiển thị NhanVien lên form
NhanVien getForm(){...} // Tạo NhanVien từ form
void updateStatus(){...} // Cập nhật trạng thái các nút
```





Cảm ơn