

SAMPLE PROJECT

BÀI 5: THỰC HIỆN DỰ ÁN 3

- ⊙ Kết thúc bài học này bạn có khả năng
 - ⊙ Hiển thị dữ liệu lên bảng
 - ⊙ Hiển thị dữ liệu lên form
 - ⊙ Thêm mới dữ liệu vào CSDL
 - ⊙ Xóa dữ liệu khỏi CSDL
 - ⊙ Cập nhật dữ liệu vào CSDL
 - ⊙ Điều hướng dữ liệu
 - ⊙ Điều khiển trạng thái form
 - ⊙ Tổ chức mã cho trang quản lý



- 📖 Giới thiệu trang quản lý nhân viên
- 📖 Tổ chức mã cho trang quản lý nhân viên
- 📖 Mã truy vấn và thao tác dữ liệu
- 📖 Mã điều hướng
- 📖 Mã điều khiển trạng thái form



Hướng dẫn viết mã cho trang quản lý

QUẢN LÝ NHÂN VIÊN

QUẢN LÝ NHÂN VIÊN QUẢN TRỊ

CẬP NHẬT DANH SÁCH

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

Hiển thị dữ liệu sau khi
của sổ mở ra

Nhấp đúp lên hàng muốn
xem hoặc sửa đổi

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

< << >> >|

Nhấp chuột vào các nút

Thao tác nào?

- ❑ Cài đặt mã cho các chức năng trên giao diện là viết mã điều khiển các sự kiện xảy ra trên giao diện nhằm thực hiện đúng mong muốn nghiệp vụ ứng dụng.
- ❑ Thông thường cần bổ sung mã vào các thành phần sau
 - ❖ Constructor
 - Lúc khởi tạo
 - ❖ windowOpen
 - Sau khi cửa sổ/hộp thoại được mở ra
 - ❖ actionPerformed
 - Khi click vào button hoặc chọn combobox
 - ❖ mouseClicked
 - Khi click chuột lên bảng

Hướng dẫn Tổ chức mã nguồn

- ❑ Không nên viết mã trực tiếp vào các điều khiển sự kiện, mã nhiều gây rối, khó quản lý
- ❑ Nên tách thành các phương thức giải quyết các chức năng sau đó gọi các phương thức này trong các điều khiển sự kiện
- ❑ Phần mã viết bổ sung nên đặt ở cuối cửa sổ hoặc hộp thoại

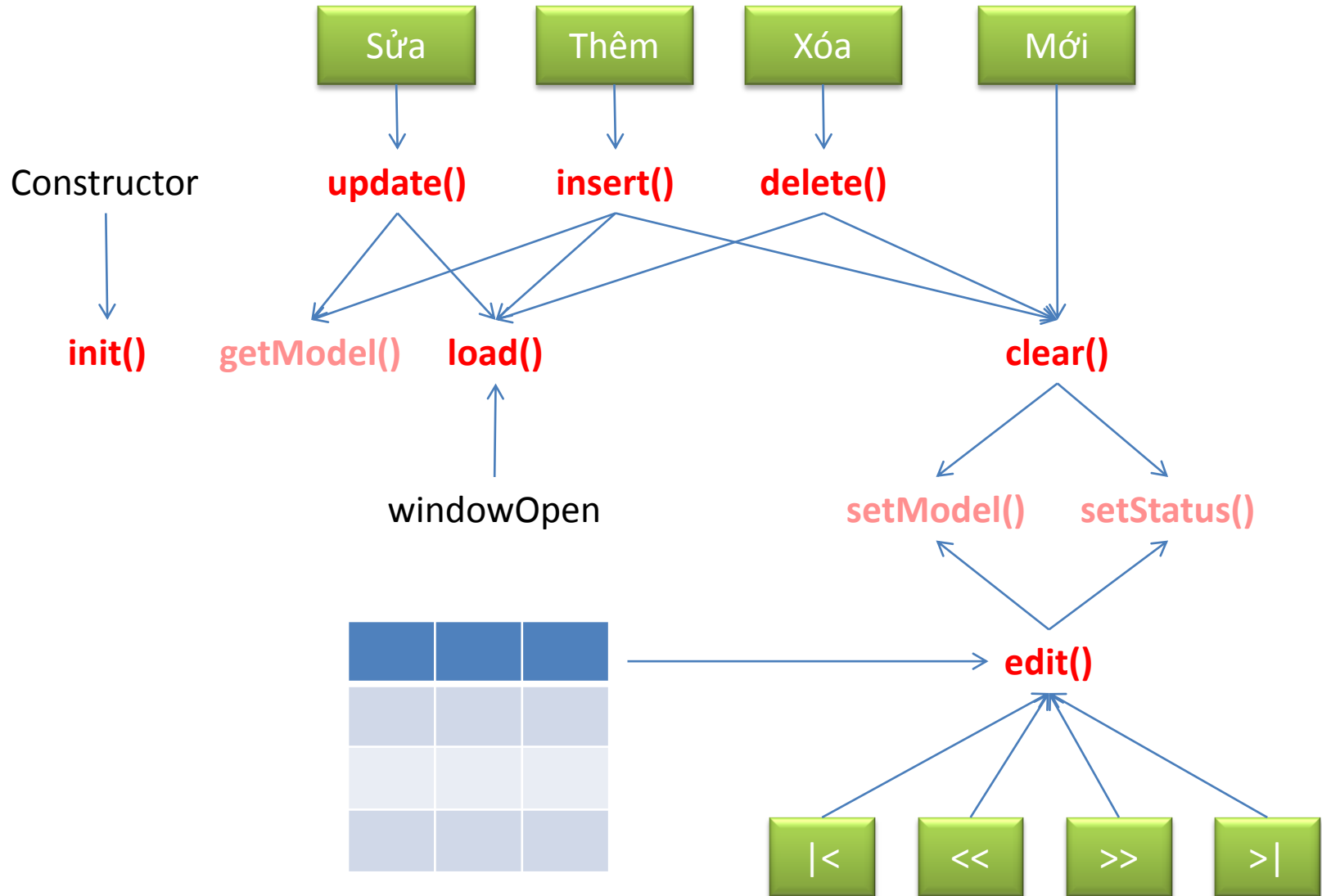
ĐƯỢC GỌI TRONG CÁC ĐIỀU KIỆN SỰ KIỆN

```
int index = 0; // vị trí của nhân viên đang hiển thị trên form
NhanVienDAO dao = new NhanVienDAO();

void init() { ...4 lines } ← Constructor
void load() { ...19 lines } ← Cửa sổ mở
void insert() { ...19 lines } ← Nhấp nút [Thêm]
void update() { ...18 lines } ← Nhấp nút [Sửa]
void delete() { ...14 lines } ← Nhấp nút [Xóa]
void edit() { ...13 lines } ← Nhấp chuột vào bảng hoặc nút điều hướng
void clear() { ...4 lines } ← Nhấp nút [Mới]

void setModel(NhanVien model) { ...8 lines }
NhanVien getModel() { ...8 lines }
void setStatus(boolean insertable) { ...13 lines }
```

SỬ DỤNG TRONG CÁC PHƯƠNG THỨC Ở TRÊN



Hướng dẫn viết mã cho các phương thức chức năng cơ sở

□ Khai báo

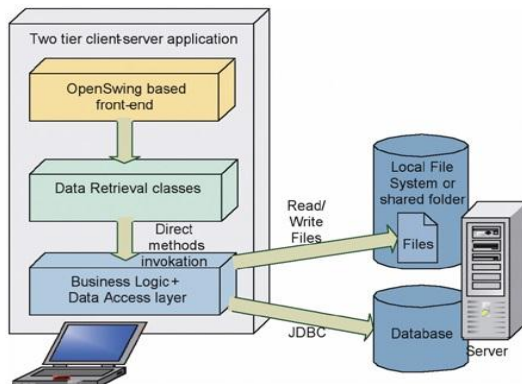
- ❖ `NhanVienDAO dao = new NhanVienDAO();`
 - Đối tượng làm việc với dữ liệu bảng NhanVien
- ❖ `int index = 0;`
 - Giữ vị trí nhân viên đang xem

□ `init()`

- ❖ `setIconImage(ShareHelper.APP_ICON);`
 - Đặt biểu tượng cho cửa sổ
- ❖ `setLocationRelativeTo(null);`
 - Đặt cửa sổ ở giữa màn hình

❑ Truy vấn dữ liệu và hiển thị lên bảng

```
void load() {  
    DefaultTableModel model = (DefaultTableModel) tblGridView.getModel();  
    model.setRowCount(0);  
    try {  
        List<NhanVien> list = dao.select();  
        for (NhanVien nv : list) {  
            Object[] row = {  
                nv.getMaNV(),  
                nv.getMatKhau(),  
                nv.getHoTen(),  
                nv.getVaiTro() ? "Trưởng phòng" : "Nhân viên"  
            };  
            model.addRow(row);  
        }  
    }  
    catch (Exception e) {  
        DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  
    }  
}
```



SAMPLE PROJECT

BÀI 5: THỰC HIỆN DỰ ÁN 3

- ❑ getModel() chuyển form thành NhanVien
- ❑ setModel() hiển thị NhanVien lên form

```
void setModel(NhanVien model) {  
    txtMaNV.setText(model.getMaNV());  
    txtHoTen.setText(model.getHoTen());  
    txtMatKhau.setText(model.getMatKhau());  
    txtXacNhanMK.setText(model.getMatKhau());  
    rdoTruongPhong.setSelected(model.getVaiTro());  
    rdoNhanVien.setSelected(!model.getVaiTro());  
}
```

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

- ❑ Form được chia làm 2 trạng thái là Insertable (nhập mới) và Editable (đang xem)
- ❑ Nếu ở trạng thái insertable
 - ❖ [Thêm], [Mới]: Enable
- ❑ Nếu ở trạng thái editable
 - ❖ [Thêm]: Disable
 - ❖ [> >] và [> |]: Disable *nếu index ở cuối*
 - ❖ [< <] và [| <]: Disable *nếu index ở đầu*
- ❑ Trạng thái form sẽ được cập nhật sau khi cửa sổ mở hoặc sau mỗi thao tác của người dùng


```
void setStatus(boolean insertable){  
    txtMaNV.setEditable(insertable);  
    btnInsert.setEnabled(insertable);  
    btnUpdate.setEnabled(!insertable);  
    btnDelete.setEnabled(!insertable);  
  
    boolean first = this.index > 0;  
    boolean last = this.index < tblGridView.getRowCount() - 1;  
    btnFirst.setEnabled(!insertable && first);  
    btnPrev.setEnabled(!insertable && first);  
    btnNext.setEnabled(!insertable && last);  
    btnLast.setEnabled(!insertable && last);  
}
```

- ❑ Xóa trắng form bằng cách đặt một nhân viên mới lên form sau đó thay đổi trạng thái của form

```
void clear() {  
    this.setModel(new NhanVien());  
    this.setStatus(true);  
}
```

- ❑ Truy vấn nhân viên tương ứng với index hiện tại và hiển thị lên form, thay đổi trạng thái form

```
void edit() {  
    try {  
        String manv = (String) tblGridView.getValueAt(this.index, 0);  
        NhanVien model = dao.findById(manv);  
        if(model != null){  
            this.setModel(model);  
            this.setStatus(false);  
        }  
    }  
    catch (Exception e) {  
        DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  
    }  
}
```

- ❑ Tạo model từ thông tin trên form, insert vào CSDL, tải lại bảng và xóa trắng form

```
void insert() {  
    NhanVien model = getModel();  
  
    String confirm = new String(txtXacNhanMK.getPassword());  
    if(confirm.equals(model.getMatKhou())) {  
        try {  
            dao.insert(model);  
            this.load();  
            this.clear();  
            DialogHelper.alert(this, "Thêm mới thành công!");  
        }  
        catch (Exception e) {  
            DialogHelper.alert(this, "Thêm mới thất bại!");  
        }  
    }  
    else{  
        DialogHelper.alert(this, "Xác nhận mật khẩu không đúng!");  
    }  
}
```

❑ Tạo model từ form, cập nhật CSDL và tải lại bảng

```
void update() {  
    NhanVien model = getModel();  
  
    String confirm = new String(txtXacNhanMK.getPassword());  
    if(!confirm.equals(model.getMatKhau())) {  
        DialogHelper.alert(this, "Xác nhận mật khẩu không đúng!");  
    }  
    else{  
        try {  
            dao.update(model);  
            this.load();  
            DialogHelper.alert(this, "Cập nhật thành công!");  
        }  
        catch (Exception e) {  
            DialogHelper.alert(this, "Cập nhật thất bại!");  
        }  
    }  
}
```

- ❑ Xóa nhân viên có mã trên form, tải lại bảng, xóa trắng form

```
void delete() {  
    if (DialogHelper.confirm(this, "Bạn thực sự muốn xóa nhân viên này?")) {  
        String manv = txtMaNV.getText();  
        try {  
            dao.delete(manv);  
            this.load();  
            this.clear();  
            DialogHelper.alert(this, "Xóa thành công!");  
        }  
        catch (Exception e) {  
            DialogHelper.alert(this, "Xóa thất bại!");  
        }  
    }  
}
```

Hướng dẫn viết mã Điều khiển sự kiện

- ❑ Sau khi đã hoàn thành các phương thức chức năng thì việc điều khiển sự kiện để thực hiện theo yêu cầu của người dùng chỉ là việc gọi các phương thức đã định nghĩa một cách hợp lý.
- ❑ Thực hiện điều khiển các sự kiện
 - ❖ Constructor
 - `init()`
 - ❖ `windowOpen`
 - `this.load()`
 - `this.setStatus(true)`

❑ Nhấp đúp vào bảng

```
if(evt.getClickCount() == 2){  
    this.index = tblGridView.rowAtPoint(evt.getPoint());  
    if (this.index >= 0) {  
        this.edit();  
        tabs.setSelectedIndex(0); // chuyển sang tab form  
    }  
}
```

❑ Nhấp các nút chức năng

- ❖ [Thêm]: `this.insert();`
- ❖ [Sửa]: `this.update();`
- ❖ [Xóa]: `this.delete();`
- ❖ [Mới]: `this.clear();`

❑ Nhấp các nút điều hướng

- ❖ [`|<`]: `this.index = 0; this.edit();`
- ❖ [`<<`]: `this.index--; this.edit();`
- ❖ [`>>`]: `this.index++; this.edit();`
- ❖ [`>|`]: `this.index = tblGridView.getRowCount()-1; edit();`

❑ NhanVienDAO

- ❖ Insert()
- ❖ Update()
- ❖ Delete()
- ❖ Select()
- ❖ findById()

❑ Đọc ghi form

- ❖ getModel()
- ❖ setModel()

❑ Trạng thái

- ❖ setStatus()

❑ Tổ chức và điều khiển sự kiện





Cảm ơn