

TÀI LIỆU THỰC TẬP LẬP TRÌNH MẠNG: LAB 06

DANH MỤC THUẬT NGỮ TIẾNG ANH

Từ	Nghĩa của từ
abstract	Trừu tượng
break	Dừng vòng lặp
catch	Từ khóa đầu của một khối bắt ngoại lệ
continue	Bỏ qua phần cuối vòng lặp, tiếp tục sang bước tiếp theo
default	Giá trị mặc định của phương thức switch()
extends	Kế thừa
final	Một hằng số, phương thức hay một lớp không được ghi đè
finally	Một phần của khối xử lý ngoại lệ try luôn được thực hiện
implements	Thực hiện giao diện
import	Khai báo một gói thư viện
instanceof	Kiểm tra một đối tượng là một thể hiện của lớp
interface	Giao diện
new	Tạo một đối tượng mới của lớp
null	Tham chiếu rỗng
package	Gói
private	Tiền tố chỉ được truy cập bởi phương thức của lớp
protected	Tiền tố được truy cập bởi phương thức của lớp, lớp con của và các lớp khác trong cùng một gói
public	Tiền tố có thể được truy cập bởi phương thức của tất cả các lớp
return	Trả về của một phương thức
super	Gọi phương thức của lớp cha
synchronized	Đồng bộ
this	Tham chiếu đến đối tượng hiện tại

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
URL	Uniform Resource Locator
CSDL	Cơ Sở Dữ Liệu
JDBC	Java Database Connectivity
CNTT	Công Nghệ Thông Tin
HĐH	Hệ Điều Hành
MVC	Model-View-Control
DNS	Domain Name System
API	Application Programming Interface
FTP	File Transfer Protocol
JDK	Java Development Kit
GB	GigaByte
UCLN	Ước Chung Lớn Nhất
BCNN	Bội Chung Nhỏ Nhất
RAM	Random Access Memory
RMI	Remote Method Invocation
JVM	Java Virtual Machine
NIC	Network Interface Card
ĐH KTKT CN	Đại học Kinh tế Kỹ thuật Công nghiệp

LỜI NÓI ĐẦU

Ngày nay do nhu cầu thực tế và do sự phát triển mạnh mẽ của nhiều công nghệ tích hợp, dẫn đến các chương trình ứng dụng hầu hết đều có khả năng thực hiện trên môi trường mạng. Ngôn ngữ JAVA là ngôn ngữ phù hợp để viết các ứng dụng mạng. So với lập trình thông thường, lập trình mạng đòi hỏi người lập trình hiểu biết và có kỹ năng tốt để viết các chương trình giao tiếp và trao đổi dữ liệu giữa các máy tính với nhau.

Để hỗ trợ sinh viên chuyên ngành CNTT trong nhà trường tiếp cận với kỹ thuật lập trình mới này, tiếp theo cuốn tài liệu học tập lý thuyết “**Công nghệ JAVA**”, chúng tôi xây dựng cuốn “**Bài tập lập trình mạng**”, nhằm cung cấp cho sinh viên những kiến thức và kỹ thuật cơ bản nhất để phát triển các chương trình ứng dụng mạng, thông qua các dạng bài tập từ cơ bản đến nâng cao qua các chủ đề: lập trình cơ bản, lập trình hướng đối tượng, lập trình CSDL JDBC, lập trình mạng dùng socket, lập trình phân tán với RMI. Sinh viên sẽ thực hiện các bài thực hành này trên phòng máy nhà trường.

Nội dung cuốn tài liệu bao gồm 12 bài lab chia thành các chủ đề khác nhau. Trong mỗi chủ đề chúng tôi đưa ra tóm tắt lý thuyết, bài tập mẫu, sau đó là bài tập tương tự, và bài tập tổng hợp. Kết quả qua những bài lab, sinh viên được rèn và thành thạo các kỹ năng lập trình hướng đối tượng, lập trình CSDL, lập trình với giao thức truyền thông có sẵn và khả năng tích hợp trong các ứng dụng khác nhau, nhất là các giao thức truyền thông thời gian thực, từ đó sinh viên có thể viết được các phần mềm quản lý theo mô hình MVC, xây dựng được các ứng dụng mạng, các ứng dụng tích hợp và triệu gọi lẫn nhau trên mạng Intranet (mạng cục bộ), mạng Internet (mạng toàn cầu), các hệ thống xử lý truy xuất dữ liệu phân tán hoàn chỉnh. Nội dung biên soạn phù hợp với chuẩn đầu ra của ngành CNTT và ngành mạng máy tính và truyền thông dữ liệu về kỹ năng và kiến thức. Sau khi học xong học phần này sinh viên có thể viết phần mềm quản lý, truyền thông.

Chúng tôi xin chân thành cảm ơn Thầy Nguyễn Hoàng Chiến, phó chủ nhiệm khoa, phụ trách khoa CNTT trường ĐH KTKT CN cùng với các đồng nghiệp đã đóng góp ý kiến cho cuốn tài liệu này. Vì tài liệu được biên soạn lần đầu, chúng tôi đã cố gắng hoàn chỉnh, song không tránh khỏi thiếu sót. Rất mong nhận được sự góp ý của bạn đọc để tài liệu học tập được hoàn thiện hơn.

Xin trân trọng cảm ơn!

Nhóm tác giả

LAB 6. LẬP TRÌNH CƠ SỞ DỮ LIỆU JDBC [3,5,8]

A. MỤC TIÊU

- Trang bị cho sinh viên kỹ năng lập trình CSDL.
- Ôn tập lại các thao tác xử lý thêm, chèn, sửa xóa
- Kết hợp kiến thức tạo giao diện (JAVA SWING) lập trình ứng dụng Quản lý theo mô hình 3 lớp (MVC).

B. NỘI DUNG

- Thực hiện được việc tải và cài đặt JDBC driver cho project
- Thực hiện kết nối CSDL với các hệ Quản trị CSDL SQL Server, MySQL,...
- Thực hiện được việc kết nối và truy xuất (thêm, chèn, sửa, xóa) CSDL.
- Thực hiện được việc xử lý kết xuất kết quả truy xuất CSDL.

C. YÊU CẦU PHẦN CỨNG, PHẦN MỀM

- Máy tính cài HĐH windows, RAM tối thiểu 1GB
- Phần mềm SQL Server 2015, NETBEAN 8.0, JDK 1.8

D. KẾT QUẢ SAU KHI HOÀN THÀNH

- Hiểu được phương pháp kết nối CSDL trong Java bằng JDBC.
- Xây dựng ứng dụng kết nối CSDL.

E. HƯỚNG DẪN CHI TIẾT

Các bước chính trong kết nối CSDL MySQL với JDBC:

1. Nạp điều khiển driver:

Class.forName(driverString);

Trong đó **driverString** là chuỗi chỉ định tên của trình điều khiển cần nạp.

Ví dụ: Nạp điều khiển driver của CSDL MySQL

```
Class.forName("com.mysql.jdbc.Driver");
```

2. Kết nối CSDL: gọi phương thức DriverManager.getConnection(url,user,pass) để nhận về đối tượng Connection kết nối với CSDL, url có dạng:

jdbc:subprotocol:subname

Trong đó:

subprotocol: giao thức tương ứng với loại CSDL

subname: tên cầu nối ODBC thông qua đó ta có thể kết nối tới CSDL

Ví dụ :

```
url = "jdbc:mysql://localhost:3306/";
```

```
user = "root";
```

```
pass = "";
```

```
myConnection = DriverManager.getConnection(url, user, pass);
```

3. Tạo đối tượng Statement: gọi phương thức `createStatement()` của đối tượng `Connection`. Đối tượng `Statement` dùng để thực hiện các câu truy vấn.

Ví dụ:

```
Statement stmt;
```

```
stmt = myConnection.createStatement();
```

4. Tạo truy vấn dữ liệu: Có 3 loại truy vấn:

`executeQuery(strSQL)` : dùng cho câu lệnh `SELECT`, kết quả trả về kiểu `ResultSet`

Ví dụ : `ResultSet rs;`

```
rs = stmt.executeQuery("Select * From TheLoai");
```

`executeUpdate(strSQL)` : dùng cập nhật dữ liệu như `INSERT`, `UPDATE`, `DELETE`

`execute(strSQL)` : dùng trong trường hợp không rõ kiểu truy vấn.

5. Đóng kết nối: gọi phương thức `close()` tương ứng để giải phóng vùng nhớ.

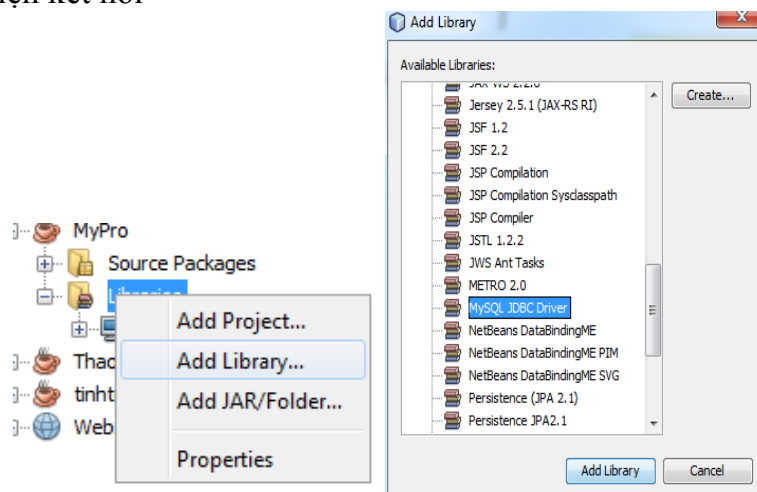
```
stmt.close();
```

```
rs.close();
```

Bài 1. Thực hiện kết nối CSDL MYSQL

Mở `MYSQL`, tạo bảng “taikhoan” gồm hai trường “ID” và “tên” trong “csdl1”. Nhập 5 bản ghi cho csdl. Thực hiện kết nối CSDL, in thông tin tài khoản ra màn hình

Bước 1: add thư viện kết nối



Hình 4. Thêm thư viện MYSQL JDBC Driver

Bước 2:

Tạo tạo bảng “taikhoan” có hai trường ID kiểu **int** và tên kiểu **char** trong csdl1 bằng `MYSQL`. Tạo file java `KetNoiCSDL` thực hiện các bước kết nối

```
public class KetNoiCSDL {  
    Connection cn= null;  
    public KetNoiCSDL() throws SQLException {
```

```
String url="jdbc:mysql://localhost:3306/csd11";
this.cn=DriverManager.getConnection(url,"root","");
}
```

Bước 3: Tạo phương thức LayDL trả về kiểu ResultSet

```
public ResultSet LayDL (String tenbang) throws SQLException {
    ResultSet kq=null;
    Statement st=this.cn.createStatement();
    String sql= "select * from taikhoan";
    kq=st.executeQuery(sql);
    return kq;
}
```

Bước 4. Tạo phương thức main và sử dụng


```
public static void main(String[] args) throws SQLException
{
    KetNoiCSDL a= new KetNoiCSDL();
    ResultSet rs=a.LayDL("taikhoan");
    while(rs.next())
    {
        System.out.println(rs.getString(1));
    }
}
```

Bài 2. Thực hành thêm thư viện kết nối Sqlserver

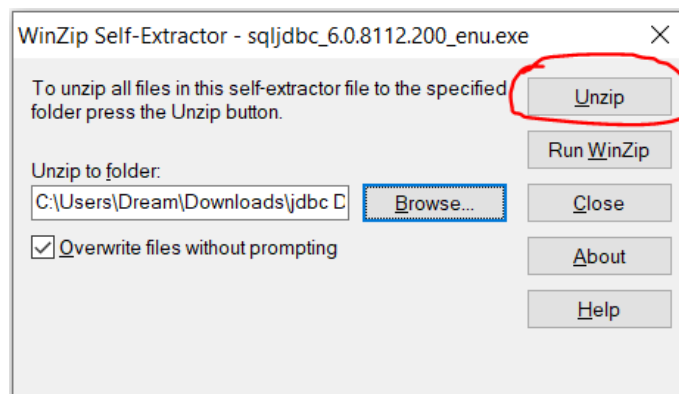
Bước 1: Tải driver từ microsoft:

<https://www.microsoft.com/en-us/download/details.aspx?id=54671>

Bước 2: Unzip file vừa tải vào folder bất kỳ

Name	Date modified	Type	Size
 sqljdbc_6.0.8112.200_enu.exe	17/02/2018 5:06 CH	Application	2.308 KB

Giao diện unzip



Hình 5. Download JDBC Driver cho SQL

Folder thu được sau khi unzip

Name	Date modified	Type	Size
sqljdbc_6.0	23/02/2019 10:49 ...	File folder	

Bước 3: Mở folder vừa unzip được theo đường dẫn\sqljdbc_6.0\enu\auth\x64

Copy file file sqljdbc_auth.dll vào thư mục jdk của java theo đường dẫn:

Win 64bit: C:\Program Files\Java\jdk1.8.0_202\bin

Win 32bit: C:\Program Files (x86)\Java\jdk1.8.0_202\bin

(Phần gạch chân có thể khác tùy phiên bản jdk)

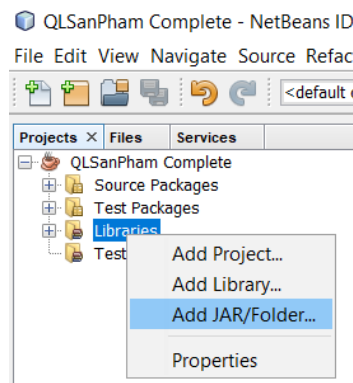
Bước 4: Mở folder unzip được ở trên theo đường dẫn\sqljdbc_6.0\enu\jre8

(Lưu ý: jre8 hoặc jre7 tùy phiên bản jre được cài đặt trên máy)

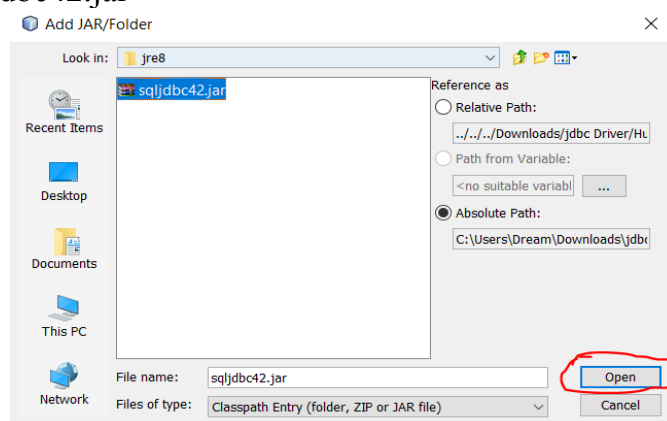
Trong đó có file: sqljdbc42.jar

Add file sqljdbc42.jar vào Library của project Netbean :

- Click chuột phải vào Library -> Add JAR/Folder

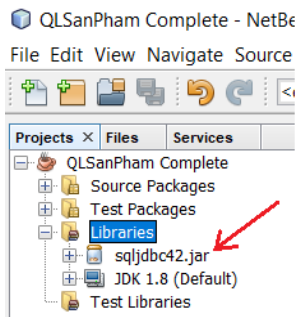


- Tìm đến file sqljdbc42.jar



Hình 6. Thêm sqljdbc42.jar

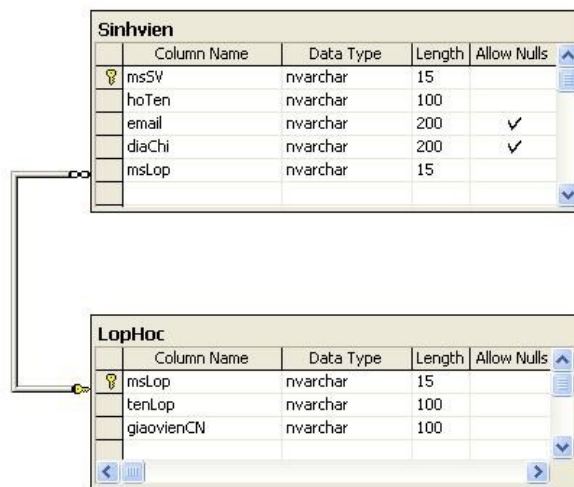
- Thêm thành công:



Để kết nối với SQL Server dùng chuỗi kết nối

```
String connString =
    "jdbc:sqlserver://localhost:1433;integratedSecurity=true;
    databaseName=" + dbName;
```

Bài 3. Cho cơ sở dữ liệu QLSV với cấu trúc bảng như sau:



- Kết nối với CSDL và hiển thị thông tin lên JFrame như hình sau:



- Khi nhấn nút Thêm hiện lên màn hình có nút Hủy và Lưu:



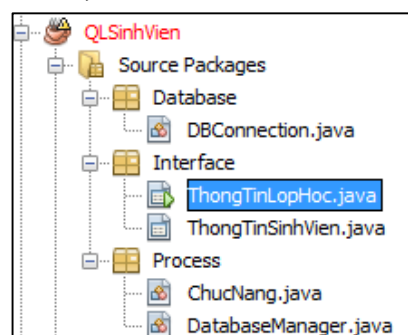
- ✓ Nhấn Hủy hoặc lưu, sau khi thực hiện sẽ khôi phục giao diện trở lại như ban đầu.
- ✓ Khi nhấn nút Sửa

- ✓ Khi nhấn nút “Xem danh sách sinh viên lớp hiện tại, 1 cửa sổ như sau xuất hiện hiển thị các sinh viên thuộc lớp hiện tại đang được chọn

Hình 6. Một số màn hình quản lý thông tin Sinh viên

Hướng dẫn:

Bước 1: Tạo project QLSinhVien, thiết kế theo mô hình MVC



Bước 2:

Tạo class DBConnection

```
public class DBConnection {
    String dbName = "QLSinhVien"; //tên cơ sở dữ liệu cần kết nối
    String connString =
    "jdbc:sqlserver://localhost:1433;integratedSecurity=true;databaseName="
```

```

+ dbName;

    public Connection GetConnection()
    {
        Connection conn;
        try {
            conn = DriverManager.getConnection(connString);
            return conn;
        } catch (SQLException ex) {
            Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE,null,ex);
            return null;
        }
    }

    public ResultSet GetData(String query)
    {
        Connection conn = GetConnection();
        if(conn==null)    //nếu không thể mở kết nối
        {
            CloseConnection(conn);    //Đóng kết nối
            return null;
        }

        Statement stm;
        try
        {
            stm = conn.createStatement();
            ResultSet rs = stm.executeQuery(query);
            return rs;
        } catch (SQLException ex) {
            Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE,null,ex);
            CloseConnection(conn);    //đóng kết nối
            return null;
        }
    }

    public boolean UpdateData(String query)
    {

```

```

        Connection conn = GetConnection();
        if(conn==null)
            return false;
        Statement stm;
        try
        {
            stm = conn.createStatement();
            stm.executeUpdate(query);
            return true;
        } catch (SQLException ex) {
Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE,null,ex);
            return false;
        }
    }

    public void CloseConnection(Connection conn)
    {
        try
        {
            conn.close();
        } catch (SQLException ex) {
Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE,null,ex);
        }
    }
}

```

Trong package process tạo hai **class** ChucNang và DatabaseManager

```

public class ChucNang {
    public static final int NONE = -1; //chưa chọn
    public static final int ADD = 0;
    public static final int UPDATE = 1;
}

public class DatabaseManager { // thực hiện thêm, sửa, xóa trên database
    public static boolean ThemLopHoc(String msLop, String tenLop, String
gvcn)
    {

```

```

        DBConnection dbConn = new DBConnection();

        String qr = "Insert Into LopHoc Values('"+msLop+"',
N'"+tenLop+"', N'"+gvcn+"')";

        return dbConn.UpdateData(qr);
    }

    public static boolean SuaLopHoc(String msLop, String tenLop, String
gvcn)
    {
        DBConnection dbConn = new DBConnection();

        String qr = "Update LopHoc Set tenlop = N'"+tenLop+"', giaovienCN
= N'"+gvcn+" Where mslop = '"+msLop+"'";

        return dbConn.UpdateData(qr);
    }

    public static boolean XoaLopHoc(String msLop)
    {
        String qr = "Delete From LopHoc Where mslop = '"+msLop+"'";
        DBConnection dbConn = new DBConnection();
        return dbConn.UpdateData(qr);
    }

    public static boolean LopHocToTable(JTable jTable)
    {
        try
        {
            DefaultTableModel dfTableModel = (DefaultTableModel)jTable.getModel();
            dfTableModel.setRowCount(0);

            DBConnection db = new DBConnection();
            ResultSet rs = db.GetData("Select * From LopHoc");
            String[] row = new String[3];
            while(rs.next())
            {
                row[0] = rs.getString(1);
                row[1] = rs.getString(2);
                row[2] = rs.getString(3);
                dfTableModel.addRow(row);
            }
        }
    }

```

```

        }
        return true;
    }catch (Exception ex)
    {
        Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE,null,ex);
        return false;
    }
}

public static boolean ThemSinhVien(String mssv, String hoTen, String
email, String diaChi, String msLop) {
    String qr = "Insert Into SinhVien Values('"+mssv+"',
N'"+hoTen+"', N'"+email+"', N'"+diaChi+"', '"+msLop+"')";
    DBConnection dbConn = new DBConnection();
    return dbConn.UpdateData(qr)==true;
}

public static boolean SuaSinhVien(String mssv, String hoTen, String
email, String diaChi, String msLop) {
    String qr = "Update SinhVien Set hoten = N'"+hoTen+"', email =
N'"+email+"', diachi = N'"+diaChi+"', mslop = '"+msLop+"' Where masv =
'"+mssv+"'";
    DBConnection dbConn = new DBConnection();
    return dbConn.UpdateData(qr);
}

public static boolean XoaSinhVien(String masv) {
    String qr = "Delete From SinhVien Where masv = '"+masv+"'";
    DBConnection dbConn = new DBConnection();
    return dbConn.UpdateData(qr);
}
}

```

//Lấy danh sách sinh viên theo lớp học

```

public static boolean SinhVienToTable_ByLopHoc(JTable jTable, String
msLop) {
    try {
        DefaultTableModel=(DefaultTableModel)jTable.getModel();
        dfTableModel.setRowCount(0);
    }
}

```

```

        DBConnection db = new DBConnection();
        ResultSet rs = db.GetData("Select * From SinhVien Where mslop =
        '"+msLop+"'");

        String[] row = new String[4];
        while(rs.next()){
            row[0] = rs.getString(1);
            row[1] = rs.getString(2);
            row[2] = rs.getString(3);
            row[3] = rs.getString(4);
            dfTableModel.addRow(row);
        }
        return true;
    }catch (Exception ex){
        Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE, null,
        ex);

        return false;
    }
}

public static int Count(String tableName, String columnName, String id)
{
    String qr = "";
    if(id.length()==0 || columnName.length()==0)
        qr = "Select COUNT(*) from "+tableName;
    else
        qr = "Select COUNT(*) from "+tableName+" Where "+columnName+" =
        '"+id+"'";

    DBConnection dbConn = new DBConnection();
    ResultSet rs = dbConn.GetData(qr);
    try
    {
        if(rs.next())
        {
            int count = Integer.parseInt(rs.getString(1));
            return count;
        }
    }
}

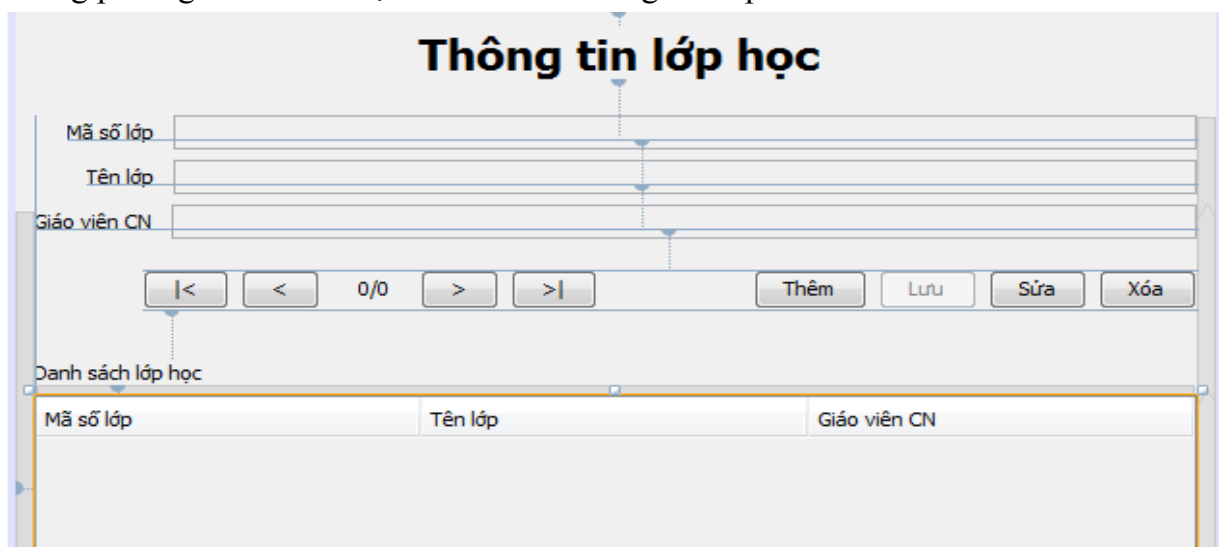
```

```

    }
    }catch (Exception ex) {
        Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE, null,
ex);
        return -1;
    }
    return -1;
}
}
}

```

Trong package **Interface** tạo hai JFrame ThôngTinLopHoc



```

public class ThôngTinLopHoc extends javax.swing.JFrame {
    DefaultTableModel    dfTableModel;
    int chucNangDaChon = ChucNang.NONE;
    public ThôngTinLopHoc() {
        initComponents();
        dfTableModel = (DefaultTableModel)tblDSLop.getModel();
    }

    //Phương thức xử lý sự kiện khi lựa chọn ở bảng thay đổi
    void tblDSLop_SelectionChanged() {
        int row = tblDSLop.getSelectedRow();
        if(row>=0) {
            String msLop = (String)dfTableModel.getValueAt(row, 0);
            String tenLop = (String)dfTableModel.getValueAt(row, 1);
            String gvcn = (String)dfTableModel.getValueAt(row, 2);

```

```

        txtMSLop.setText(msLop.trim());
        txtTenLop.setText(tenLop.trim());
        txtGVCN.setText(gvcn.trim());
    }else {
        txtMSLop.setText("");
        txtTenLop.setText("");
        txtGVCN.setText("");
    }
    ReloadLblIndexTblDSLopHoc();
}

//Lấy dữ liệu cho bảng lớp học
void ReloadTaleLopHoc() {
    if(DatabaseManager.LopHocToTable(tblDSLop)==false)
        JOptionPane.showMessageDialog(null, "Lấy dữ liệu lớp học
        có lỗi", "Có lỗi xảy ra", JOptionPane.ERROR_MESSAGE);
    ReloadLblIndexTblDSLopHoc();
}

//Lấy lại dữ liệu label hiện chỉ số hàng hiện tại
void ReloadLblIndexTblDSLopHoc() {
    int rowSelected = tblDSLop.getSelectedRow();
    int totalRow = tblDSLop.getRowCount();
    lblIndexTblLopHoc.setText((rowSelected+1)+"/" + totalRow);
}

void SwitchMode(int chucNang) {
    chucNangDaChon = chucNang;
    switch (chucNang) {
        case ChucNang.ADD:{
            boolean trangThai = true;
            txtMSLop.setEnabled(trangThai);
            txtTenLop.setEnabled(trangThai);
            txtGVCN.setEnabled(trangThai);
            txtMSLop.requestFocus();
            btnSave.setEnabled(trangThai);
            btnUpdate.setEnabled(!trangThai);
        }
    }
}

```



```

        btnDelete.setEnabled(!trangThai);
        txtMSLop.setText("");
        txtTenLop.setText("");
        txtGVCN.setText("");
        btnAdd.setText("Hủy");
        break;
    }
    case ChucNang.UPDATE: {
        boolean trangThai = true;
        txtTenLop.setEnabled(trangThai);
        txtGVCN.setEnabled(trangThai);
        txtTenLop.requestFocus();
        btnSave.setEnabled(trangThai);
        btnAdd.setEnabled(!trangThai);
        btnDelete.setEnabled(!trangThai);
        btnUpdate.setText("Hủy");
        break;
    }
    case ChucNang.NONE: {
        boolean trangThai = false;
        txtMSLop.setEnabled(trangThai);
        txtTenLop.setEnabled(trangThai);
        txtGVCN.setEnabled(trangThai);
        btnSave.setEnabled(trangThai);
        btnAdd.setEnabled(true);
        btnUpdate.setEnabled(true);
        btnDelete.setEnabled(true);
        btnAdd.setText("Thêm");
        btnUpdate.setText("Sửa");
    }
}

}

boolean CheckInput() {
    String msLop = txtMSLop.getText().trim();

```

```

        String tenLop = txtTenLop.getText().trim();
        String gvcn = txtGVCN.getText().trim();
        if(msLop.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập mã số lớp", "Chưa
nhập mã số lớp", JOptionPane.WARNING_MESSAGE);
            txtMSLop.requestFocus();
            return false;
        }
        if(tenLop.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập tên lớp",
"Chưa nhập tên lớp", JOptionPane.WARNING_MESSAGE);
            txtTenLop.requestFocus();
            return false;
        }
        if(gvcn.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập giáo
viên chủ nhiệm", "Chưa nhập giáo viên chủ nhiệm",
JOptionPane.WARNING_MESSAGE);
            txtGVCN.requestFocus();
            return false;
        }
        return true;
    }

    void Exit() {
        int result = JOptionPane.showConfirmDialog(null, "Bạn có muốn
thoát không?", "Thoát?", JOptionPane.OK_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE);
        if(result == JOptionPane.CANCEL_OPTION)
            return;
        this.dispose();
    }

    private void formWindowOpened(java.awt.event.WindowEvent evt) {
        tblDSLop.getSelectionModel().addListSelectionListener(new
ListSelectionListener()
        {

```

```

        @Override
        public void valueChanged(ListSelectionEvent e) {
            TblDSLop_SelectionChanged();
        }
    });
    ReloadTaleLopHoc();
}

private void btnXemLopActionPerformed(java.awt.event.ActionEvent evt)
{
    String msLop = txtMSLop.getText().trim();
    String tenLop = txtTenLop.getText().trim();
    if(msLop.length()==0) {
        JOptionPane.showMessageDialog(null, "Chưa chọn lớp để xem",
        "Chưa chọn lớp", JOptionPane.WARNING_MESSAGE);
        return;
    }
    ThôngTinSinhVien ttstv = new ThôngTinSinhVien(msLop, tenLop);
    ttstv.setVisible(true);
}

private void formWindowClosing(java.awt.event.WindowEvent evt) {
    exit();
}

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    if(chucNangDaChon==ChucNang.NONE)
        SwitchMode(ChucNang.ADD);
    else
        SwitchMode(ChucNang.NONE);
}

private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    if(tblDSLop.getSelectedRow()==-1) {
        JOptionPane.showMessageDialog(null, "Chưa chọn lớp để sửa",
        "Chưa chọn lớp", JOptionPane.WARNING_MESSAGE);
        return;
    }
}

```

```

        if(chucNangDaChon==ChucNang.NONE)
            SwitchMode(ChucNang.UPDATE);
        else
            SwitchMode(ChucNang.NONE);
    }
    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt)
    {
        int selectedRow = tblDSLop.getSelectedRow();
        if(selectedRow<0) {
            JOptionPane.showMessageDialog(null, "Bạn chưa chọn lớp
            nào để xóa", "Chưa chọn lớp", JOptionPane.INFORMATION_MESSAGE);
            return;
        }
        int result = JOptionPane.showConfirmDialog(null, "Bạn có muốn xóa
        lớp này không", "Xóa lớp?",
        JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);
        if(result == JOptionPane.CANCEL_OPTION)
            return;
        String msLop = (String)tblDSLop.getValueAt(selectedRow, 0);
        if(DatabaseManager.Count("SinhVien", "mslop", msLop)>0) {
            JOptionPane.showMessageDialog(null, "Đã có sinh viên
            trong lớp này!", "Không thể xóa", JOptionPane.WARNING_MESSAGE);
            return;
        }
        if(DatabaseManager.XoaLopHoc(msLop))
        {
            btnAdd.requestFocus();
            SwitchMode(ChucNang.NONE);
            ReloadTaleLopHoc();
            JOptionPane.showMessageDialog(null, "Xóa thành công",
            "Thành công", JOptionPane.INFORMATION_MESSAGE);
            return;
        }
        else
        {

```

```

        JOptionPane.showMessageDialog(null, "Xóa thất bại", "Có lỗi xảy ra", JOptionPane.ERROR_MESSAGE);
        return;
    }
}

private void btnFirstActionPerformed(java.awt.event.ActionEvent evt) {
    if(tblDSLop.getRowCount()>0)
        tblDSLop.getSelectionModel().setSelectionInterval(0, 0);
}

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
    int rowSelected = tblDSLop.getSelectedRow();
    if(rowSelected>0) {
        rowSelected--;
        tblDSLop.getSelectionModel().setSelectionInterval(rowSelected, rowSelected);
    }
}

private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
    int rowSelected = tblDSLop.getSelectedRow();
    if(rowSelected<tblDSLop.getRowCount()-1)
    {
        rowSelected++;
        tblDSLop.getSelectionModel().setSelectionInterval(rowSelected, rowSelected);
    }
}

private void btnLastActionPerformed(java.awt.event.ActionEvent evt)
{
    if(tblDSLop.getRowCount()>0) {
        int lastRowIndex = tblDSLop.getRowCount()-1;
        tblDSLop.getSelectionModel().setSelectionInterval(lastRowIndex, lastRowIndex);
    }
}
}

```

```

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    String msLop = txtMSLop.getText().trim();
    String tenLop = txtTenLop.getText().trim();
    String gvcn = txtGVCN.getText().trim();
    if(chucNangDaChon==ChucNang.ADD) {
        if(CheckInput()==false)
            return;
        if(DatabaseManager.Count("LopHoc", "mslop", msLop)>0) {
            txtMSLop.requestFocus();
            JOptionPane.showMessageDialog(null, "Mã lớp bạn nhập
đã tồn tại trong csdl", "Trùng mã", JOptionPane.WARNING_MESSAGE);
            return;
        }
        if(DatabaseManager.ThemLopHoc(msLop, tenLop, gvcn)) {
            btnAdd.requestFocus();
            SwitchMode(ChucNang.NONE);
            ReloadTaleLopHoc();
            JOptionPane.showMessageDialog(null, "Thêm thành
công", "Thành công", JOptionPane.INFORMATION_MESSAGE);
            return;
        }else {
            JOptionPane.showMessageDialog(null, "Thêm thất
bại", "Có lỗi xảy ra", JOptionPane.ERROR_MESSAGE);
            return;
        }
    }
    if(chucNangDaChon==ChucNang.UPDATE) {
        if(CheckInput()==false)
            return;
        if(DatabaseManager.SuaLopHoc(msLop, tenLop, gvcn)) {
            btnUpdate.requestFocus();
            SwitchMode(ChucNang.NONE);
            ReloadTaleLopHoc();
            JOptionPane.showMessageDialog(null, "Sửa thành công",

```

```

        "Thành công", JOptionPane.INFORMATION_MESSAGE);
        return;
    }else
    {
        JOptionPane.showMessageDialog(null, "Sửa thất bại", "Có
lỗi ", JOptionPane.ERROR_MESSAGE);
        return;
    }
}

private void txtMSLopKeyTyped(java.awt.event.KeyEvent evt) {
    if ((txtMSLop.getText() + evt.getKeyChar()).length() > 15)
        evt.consume();
}

private void txtTenLopKeyTyped(java.awt.event.KeyEvent evt) {
    if ((txtTenLop.getText() + evt.getKeyChar()).length() > 50)
        evt.consume();
}

private void txtGVCNKeyTyped(java.awt.event.KeyEvent evt) {
    if ((txtGVCN.getText() + evt.getKeyChar()).length() > 50)
        evt.consume();
}
}

```

Bài 4. Xây dựng ứng dụng quản lý CSDL theo mô hình 3 lớp.

Thiết Kế & Cài đặt CSDL: QLSanpham, có 2 Table:

- LoaiSP(Maloai char(2), Tenloai nvarchar(20))
- SanPham(MaSP char(4), TenSP nvarchar(20), Dongia BigInt, Maloai char(2))

Nhập liệu như sau:

LoaiSP

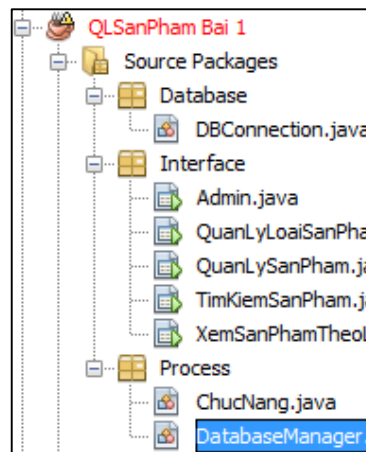
	Maloai	Tenloai
1	BK	Bánh kẹo
2	GK	Giải khát
3	MP	Mỹ phẩm

SanPham

	MaSP	TenSP	Dongia	Maloai
1	SP01	Bánh mì	10000	BK
2	SP02	Bánh bao	15000	BK
3	SP03	Coca c...	12000	GK
4	SP04	Pepsi	11000	GK
5	SP05	Kem ch...	85000	MP

Hướng dẫn:

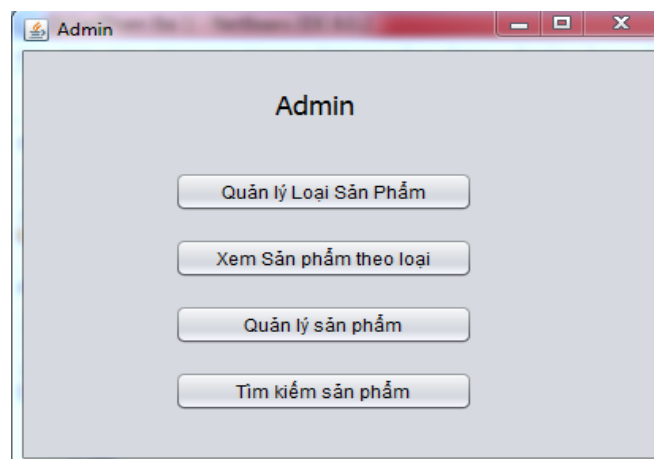
Tạo Project: QLSanpham. Tạo 3 gói (Database, Interface, Process) tương ứng



Trong packages Database tạo lớp DBConnection.java với nội dung tương tự file DBConnection.java trong bài tập trên; Thay tên cơ sở dữ liệu cần kết nối:

```
String dbName = "QLSanPham";
String connString =
"jdbc:sqlserver://localhost:1433;integratedSecurity=true;databaseName=" +
dbName;
```

Trong packages “**Interface**”, thiết kế màn hình ban đầu “Admin” gồm các chức năng:



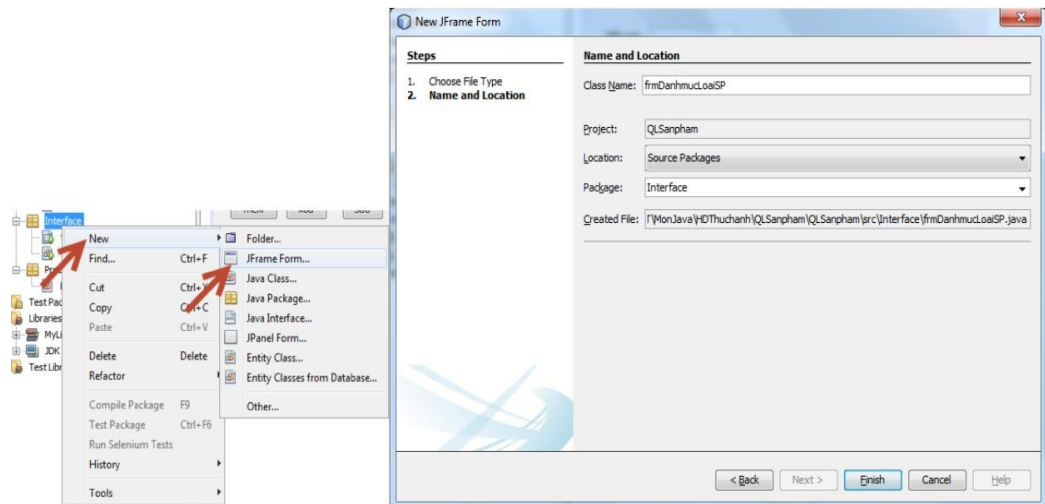
Xử lý sự kiện khi bấm btnDanhMucLoaiSP mở form QuanLySanPham

```
private void btnDanhMucLoaiSPActionPerformed(java.awt.event.ActionEvent
    evt) {
    QuanLyLoaiSanPham danhMucLoaiSP = new QuanLyLoaiSanPham();
    danhMucLoaiSP.admin = this;
    danhMucLoaiSP.setVisible(true);
    this.setVisible(false);
}
```


Tương tự viết sự kiện khi bấm btn XemSPTheoLoai hiện lên form XemSanPhamTheoLoai, bấm btnDanhMucSP hiện form QuanLySanPham, btnTimKiem hiện form TimKiemSanPham.

Thiết kế giao diện cho Form “ QuanLySanPham”

- Tạo mới JFrame Form trong Package Interface:



Kéo thả lần lượt các Control JLabel vào Form:



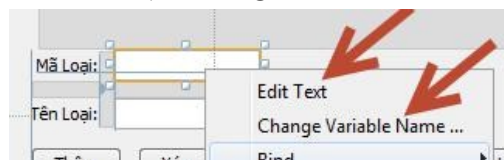
Double Click vào Control để cập nhật nhãn:



- Kéo thả lần lượt các Control JTextField vào Form:



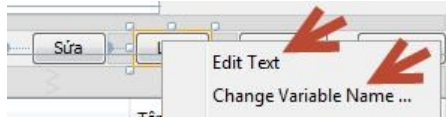
Click phải, chọn Edit Text (cập nhật nhãn), Change Variable Name (Đặt tên)



- Kéo thả lần lượt các Control JButton vào Form:



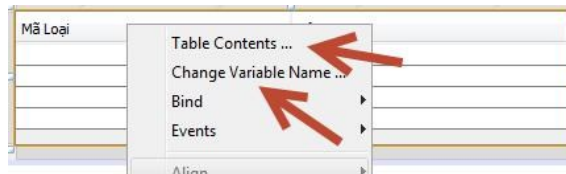
Đặt nhãn và tên cho các JButton (Tương tự JTextField và JLabel)



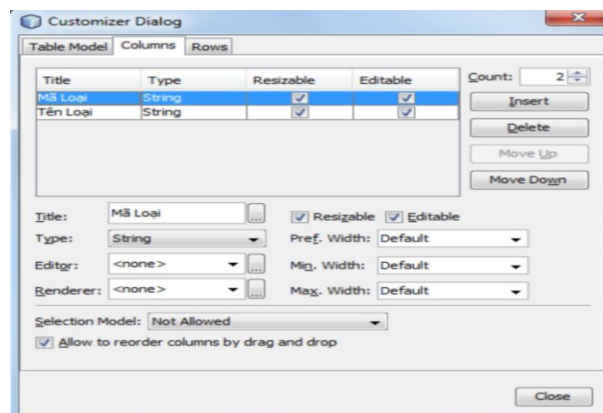
Kéo thả JTable vào Form:



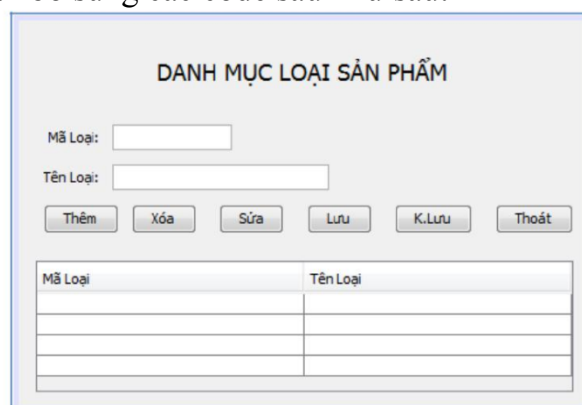
Click phải: Table Contents để điều chỉnh thiết kế, Change Variable Name (Đặt tên)



- Màn hình điều chỉnh thiết kế: Thực hiện thêm, xóa và cập nhật nhãn cho các tiêu đề cột



Mã nguồn file QuanLyLoaiSanPham.java, ngoài code tự sinh khi thiết kế giao diện dùng hình thức Design, cần bổ sung các code sau như sau:



```

public class QuanLyLoaiSanPham extends javax.swing.JFrame {
    DefaultTableModel dfTableModel;
    int chucNangDaChon = ChucNang.NONE;
    public Admin admin;
    public QuanLyLoaiSanPham() {
        initComponents();
        dfTableModel = (DefaultTableModel)tblLoaiSP.getModel();
    }
    //Phương thức load dữ liệu
    void ReloadLoaiSanPham() {
        if(DatabaseManager.LoaiSanPhamToTable(tblLoaiSP)==false) {
            JOptionPane.showMessageDialog(null, "Tải dữ liệu loại sản phẩm có lỗi, vui lòng thử lại sau", "Lỗi CSDL",
            JOptionPane.ERROR_MESSAGE);
            return;
        }
    }
    //Phương thức thay đổi lựa chọn dòng dữ liệu trên bảng
    void TblLoaiSP_SelectionChanged() {
        int row = tblLoaiSP.getSelectedRow();
        if(row>=0) {
            String maLoai = (String)dfTableModel.getValueAt(row, 0);
            String tenLoai = (String)dfTableModel.getValueAt(row, 1);
            txtMaLoai.setText(maLoai);
            txtTenLoai.setText(tenLoai);
        }else {
            txtMaLoai.setText("");
            txtTenLoai.setText("");
        }
    }
    void SetStateControl(boolean trangThai) {
        btnAdd.setEnabled(trangThai);
        btnUpdate.setEnabled(trangThai);
    }
}

```

```

        btnDelete.setEnabled(trangThai);
        btnSave.setEnabled(!trangThai);
        btnDontSave.setEnabled(!trangThai);
        btnExit.setEnabled(trangThai);
        tblLoaiSP.setEnabled(trangThai); }
    }

    //Phương thức chọn chức năng
    void SwitchMode(int chucNang) {
        chucNangDaChon = chucNang;
        switch (chucNang) {
            case ChucNang.ADD: {
                SetStateControl(false);
                txtMaLoai.setEnabled(true);
                txtTenLoai.setEnabled(true);
                txtMaLoai.setText("");
                txtTenLoai.setText("");
                txtMaLoai.requestFocus();
                break;
            }
            case ChucNang.UPDATE:{
                SetStateControl(false);
                txtMaLoai.setEnabled(false);
                txtTenLoai.setEnabled(true);
                txtMaLoai.requestFocus();
                break;
            }
            case ChucNang.NONE: {
                SetStateControl(true);
                txtMaLoai.setEnabled(false);
                txtTenLoai.setEnabled(false);
                TblLoaiSP_SelectionChanged();
                break;
            }
        }
    }
}

```

```

    }

    void Exit()
    {
        int result = JOptionPane.showConfirmDialog(null, "Bạn có muốn thoát không?", "Thoát?", JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);

        if(result == JOptionPane.CANCEL_OPTION)
            return;

        this.dispose();

        if(admin!=null)
            admin.setVisible(true);
    }

    boolean CheckInput() {
        String maLoai = txtMaLoai.getText().trim();
        String tenLoai = txtTenLoai.getText().trim();

        if(maLoai.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập mã loại", "Chưa nhập mã loại", JOptionPane.WARNING_MESSAGE);
            txtMaLoai.requestFocus();
            return false;
        }

        if(tenLoai.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập tên loại", "Chưa nhập tên loại", JOptionPane.WARNING_MESSAGE);
            txtTenLoai.requestFocus();
            return false;
        }

        return true;
    }

    private void formWindowOpened(java.awt.event.WindowEvent evt) {
        ReloadLoaiSanPham();

        tblLoaiSP.getSelectionModel().addListSelectionListener(new ListSelectionListener()
        {
            @Override

```

```

        public void valueChanged(ListSelectionEvent e) {
            TblLoaiSP_SelectionChanged();
        }
    });
}

private void txtMaLoaiKeyTyped(java.awt.event.KeyEvent evt) {
    if ((txtMaLoai.getText() + evt.getKeyChar()).length() > 2)
        evt.consume();
}

private void txtTenLoaiKeyTyped(java.awt.event.KeyEvent evt) {
    if ((txtTenLoai.getText() + evt.getKeyChar()).length() > 20)
        evt.consume();
}

//Code cho button Thêm
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    if(chucNangDaChon == ChucNang.NONE)
        SwitchMode(ChucNang.ADD);
    else
        SwitchMode(ChucNang.NONE);
}

private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    int rowSelected = tblLoaiSP.getSelectedRow();
    if(rowSelected<0) {
        JOptionPane.showMessageDialog(null, "Chưa chọn loại sản phẩm nào để sửa", "Chưa chọn loại sản phẩm",
JOptionPane.WARNING_MESSAGE);
        return;
    }
    if(chucNangDaChon == ChucNang.NONE)
        SwitchMode(ChucNang.UPDATE);
    else
        SwitchMode(ChucNang.NONE);
}

private void btnDontSaveActionPerformed(java.awt.event.ActionEvent evt){

```

```

        SwitchMode(ChucNang.NONE);
    }
//Code cho chức năng xóa dữ liệu
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt)
{
    int rowSelected = tblLoaiSP.getSelectedRow();
    if(rowSelected<0) {
        JOptionPane.showMessageDialog(null, "Bạn chưa chọn nhà xuất
bản", "Chưa chọn nhà xuất bản", JOptionPane.INFORMATION_MESSAGE);
        return;
    }
    int result = JOptionPane.showConfirmDialog(null, "Bạn muốn xóa
sản phẩm này không", "Xóa?",JOptionPane.OK_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE);
    if(result == JOptionPane.CANCEL_OPTION)
        return;
    String maLoai = (String)tblLoaiSP.getValueAt(rowSelected, 0);
    if(DatabaseManager.Count("SanPham", "MaLoai", maLoai)>0)
    {
        JOptionPane.showMessageDialog(null, "Đã có sản phẩm nằm trong
loại sản phẩm này, không thể xóa!", "Không thể xóa",
JOptionPane.WARNING_MESSAGE);
        return;
    }
    if(DatabaseManager.XoaLoaiSanPham(maLoai))
    {
        ReloadLoaiSanPham();
        JOptionPane.showMessageDialog(null, "Xóa thành công", "Thành
công", JOptionPane.INFORMATION_MESSAGE);
        return;
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Xóa thất bại", "Thất bại",
JOptionPane.ERROR_MESSAGE);
    }
}

```

```

        return;
    }
}
//Code cho chức năng Lưu
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    String maLoai = txtMaLoai.getText().trim();
    String tenLoai = txtTenLoai.getText().trim();
    if(chucNangDaChon == ChucNang.ADD) {
        if(CheckInput()==false)
            return;
        if(DatabaseManager.Count("LoaiSP", "MaLoai", maLoai)>0) {
            JOptionPane.showMessageDialog(null, "Mã loại này đã tồn
tại trong csdl", "Trùng mã", JOptionPane.WARNING_MESSAGE);
            return;
        }
        if(DatabaseManager.ThemLoaiSanPham(maLoai, tenLoai)) {
            SwitchMode(ChucNang.NONE);
            ReloadLoaiSanPham();
            btnAdd.requestFocus();
            JOptionPane.showMessageDialog(null, "Thêm thành công", "Thành
công", JOptionPane.INFORMATION_MESSAGE);
            return;
        }else{
            JOptionPane.showMessageDialog(null, "Thêm thất bại", "Thất bại",
JOptionPane.ERROR_MESSAGE);
            return;
        }
    }else if(chucNangDaChon==ChucNang.UPDATE) {
        if(CheckInput()==false)
            return;
        if(DatabaseManager.SuaLoaiSanPham(maLoai, tenLoai)) {
            SwitchMode(ChucNang.NONE);
            ReloadLoaiSanPham();
            btnUpdate.requestFocus();

```

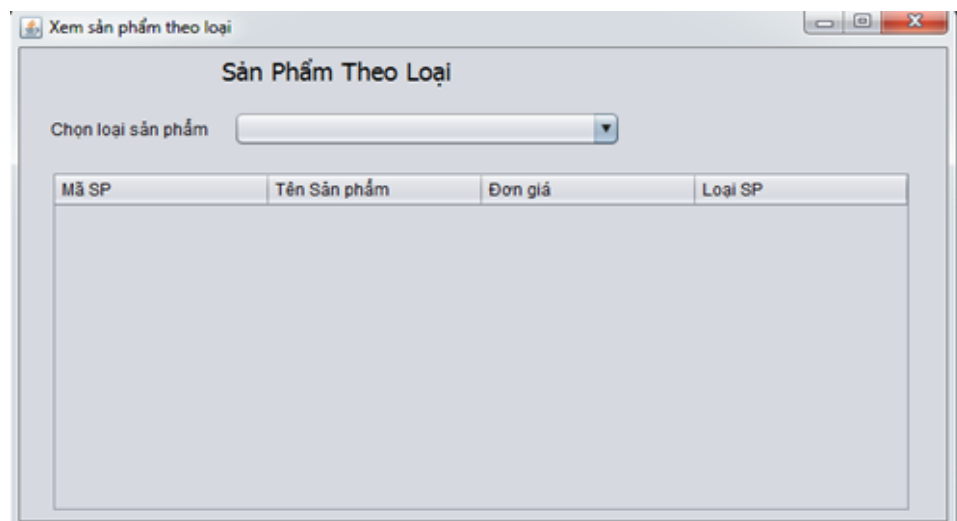


```

        JOptionPane.showMessageDialog(null, "Sửa thành công",
        "Thành công", JOptionPane.INFORMATION_MESSAGE);
        return;
    }
    else
    {
        btnAdd.requestFocus();
        JOptionPane.showMessageDialog(null, "Sửa thất bại", "Thất bại",
        JOptionPane.ERROR_MESSAGE);
        return;
    }
}
}

```

Tương tự thiết kế giao diện cho chức năng “Xem sản phẩm theo loại



Mã nguồn cho giao diện Sản Phẩm Theo Loại: Tạo Class XemSanPhamTheoLoai

```

public class XemSanPhamTheoLoai extends javax.swing.JFrame {
    public Admin admin;
    public XemSanPhamTheoLoai() {
        initComponents();
    }

    void ReloadSanPhamTheoLoaiDaChon() {
        String maLoai = (String)cbxLoaiSP.getSelectedItem();
        if(DatabaseManager.SanPhamToTable_ByLoaiSP(tblSanPham, maLoai)==false{
            JOptionPane.showMessageDialog(null, "Tải dữ liệu có lỗi, vui

```

```

lòng thử lại sau", "Lỗi CSDL", JOptionPane.ERROR_MESSAGE);

        return;
    }
}

void ReloadComboBoxLoaiSP() {
    if(DatabaseManager.LoaiSanPhamToComboBox(cbxLoaiSP)==false) {
        JOptionPane.showMessageDialog(null, "Tải dữ liệu loại sản phẩm có lỗi, vui lòng thử lại", "Lỗi CSDL", JOptionPane.ERROR_MESSAGE);
        return;
    }
}

void Exit() {
    int result = JOptionPane.showConfirmDialog(null, "Bạn muốn thoát không?", "Thoát?", JOptionPane.OK_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE);
    if(result == JOptionPane.CANCEL_OPTION) {
        return;
    }
    this.dispose();
    if(admin!=null)
        admin.setVisible(true);
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    ReloadComboBoxLoaiSP();
}

private void cbxLoaiSPItemStateChanged(java.awt.event.ItemEvent evt) {
    ReloadSanPhamTheoLoaiDaChon();
}

private void formWindowClosing(java.awt.event.WindowEvent evt) {
    Exit();
}
}

```

Thiết kế giao diện cho form quản lý sản phẩm

Tạo class QuanLySanPham

```
public class QuanLySanPham extends javax.swing.JFrame {
    DefaultTableModel tableModel;
    int chucNangDaChon = ChucNang.NONE;
    public Admin admin;
    public QuanLySanPham() {
        initComponents();
        tableModel = (DefaultTableModel)tblSanPham.getModel();
    }
    void ReloadSanPham() {
        if(DatabaseManager.SanPhamToTable(tblSanPham)==false) {
            JOptionPane.showMessageDialog(null, "Tải dữ liệu sản phẩm có lỗi, vui lòng thử lại sau", "Lỗi CSDL", JOptionPane.ERROR_MESSAGE);
            return;
        }
    }
    void ReloadComboBoxLoaiSP() {
        if(DatabaseManager.LoaiSanPhamToComboBox(cbxLoaiSP)==false) {
            JOptionPane.showMessageDialog(null, "Tải dữ liệu loại sản phẩm có lỗi, vui lòng thử lại sau", "Lỗi CSDL", JOptionPane.ERROR_MESSAGE);
            return;
        }
    }
    void TblDanhMucSP_SelectionChanged() {
        int row = tblSanPham.getSelectedRow();
        if(row>=0) {
```

```

        String maSP = (String)tableModel.getValueAt(row, 0);
        String tenSP = (String)tableModel.getValueAt(row, 1);
        String giaBan = (String)tableModel.getValueAt(row, 2);
        String loaiSP = (String)tableModel.getValueAt(row, 3);
        txtMaSP.setText(maSP);
        txtTenSP.setText(tenSP);
        txtPrice.setText(giaBan);
        cbxLoaiSP.setSelectedItem(loaiSP);
    }else {
        ClearTextField();
    }
}

void SetStateControl(boolean trangThai) {
    btnAdd.setEnabled(trangThai);
    btnUpdate.setEnabled(trangThai);
    btnDelete.setEnabled(trangThai);
    btnSave.setEnabled(!trangThai);
    btnNotSave.setEnabled(!trangThai);
    btnExit.setEnabled(trangThai);
    tblSanPham.setEnabled(trangThai);
}

void SetStateTextField(boolean trangThai) {
    txtMaSP.setEnabled(trangThai);
    txtTenSP.setEnabled(trangThai);
    txtPrice.setEnabled(trangThai);
    cbxLoaiSP.setEnabled(trangThai);
}

void ClearTextField() {
    txtMaSP.setText("");
    txtTenSP.setText("");
    txtPrice.setText("");
}

void SwitchMode(int chucNang) {
    chucNangDaChon = chucNang;
    switch (chucNang) {

```

```

        case ChucNang.ADD: {
            SetStateControl(false);
            SetStateTextField(true);
            ClearTextField();
            txtMaSP.requestFocus();
            break;
        }
        case ChucNang.UPDATE: {
            SetStateControl(false);
            SetStateTextField(true);
            txtMaSP.setEnabled(false);
            txtTenSP.requestFocus();
            break;
        }
        case ChucNang.NONE: {
            SetStateControl(true);
            SetStateTextField(false);
            TblDanhMucSP_SelectionChanged();
            break;
        }
    }

}

void Exit() {
    int result = JOptionPane.showConfirmDialog(null, "Bạn có muốn thoát?", "Thoát?", JOptionPane.OK_CANCEL_OPTION,
    JOptionPane.QUESTION_MESSAGE);
    if(result == JOptionPane.CANCEL_OPTION)
        return;
    this.dispose();
    if(admin!=null)
        admin.setVisible(true);
}

boolean CheckInput() {
    String maSP = txtMaSP.getText().trim();

```

```

        String tenSP = txtTenSP.getText().trim();
        String price = txtPrice.getText().trim();
        String maLoai = (String)cbxLoaiSP.getSelectedItem();
        if(maSP.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập mã sản phẩm", "Chưa nhập mã sản phẩm", JOptionPane.WARNING_MESSAGE);
            txtMaSP.requestFocus();
            return false;
        }
        if(tenSP.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập tên sản phẩm", "Chưa nhập tên sản phẩm", JOptionPane.WARNING_MESSAGE);
            txtTenSP.requestFocus();
            return false;
        }
        if(price.length()==0) {
            JOptionPane.showMessageDialog(null, "Vui lòng nhập giá bán", "Chưa nhập giá bán", JOptionPane.WARNING_MESSAGE);
            txtPrice.requestFocus();
            return false; }
        if(maLoai == null || maLoai.length() == 0) {
            JOptionPane.showMessageDialog(null, "Chưa có mã loại. Vui lòng thêm ít nhất một mã loại vào CSDL", "Chưa có mã loại", JOptionPane.WARNING_MESSAGE);
            btnExit.requestFocus();
            return false;
        } return true;
    }
}

```

Thiết kế giao diện Tìm kiếm sản phẩm như sau

Tạo class TimKiemSanPham, code cho chức năng tìm kiếm sản phẩm.

```
private void btnTimKiemActionPerformed(java.awt.event.ActionEvent evt) {
    if(rdbTheoTen.isSelected()==true) {
        String tenSP = txtTenSP.getText().trim();
        if(tenSP.length()==0){
            JOptionPane.showMessageDialog(null, "Vui lòng nhập từ khóa
tên sản phẩm", "Chưa nhập thông tin cần
thiết",JOptionPane.WARNING_MESSAGE);
            txtTenSP.requestFocus();
            return;
        }
        if(DatabaseManager.SearchSanPhamWithName(tblDanhMucSP, tenSP)==false
        {
            JOptionPane.showMessageDialog(null, "Tìm kiếm thất bại", "Lỗi
CSDL", JOptionPane.ERROR_MESSAGE);
            return;
        }
    }
    if(rdbTheoGia.isSelected()==true)
    {
        String giaTu = txtGiaTu.getText().trim();
        String giaDen = txtGiaDen.getText().trim();
        if(giaTu.length()==0){
            JOptionPane.showMessageDialog(null, "Vui lòng nhập giá
từ", "Chưa nhập thông tin cần thiết", JOptionPane.WARNING_MESSAGE);
            txtGiaTu.requestFocus();
            return;
        }
    }
}
```

```

    }

    if(giaDen.length()==0){
        JOptionPane.showMessageDialog(null, "Vui lòng nhập giá đến", "Chưa
        nhập thông tin cần thiết", JOptionPane.WARNING_MESSAGE);

        txtGiaDen.requestFocus();

        return;
    }

    if(DatabaseManager.SearchSanPhamWithPrice(tblDanhMucSP, giaTu,
    giaDen)==false)
    {
        JOptionPane.showMessageDialog(null, "Tìm kiếm thất bại", "Lỗi
        CSDL", JOptionPane.ERROR_MESSAGE);
    return ;
    }
}

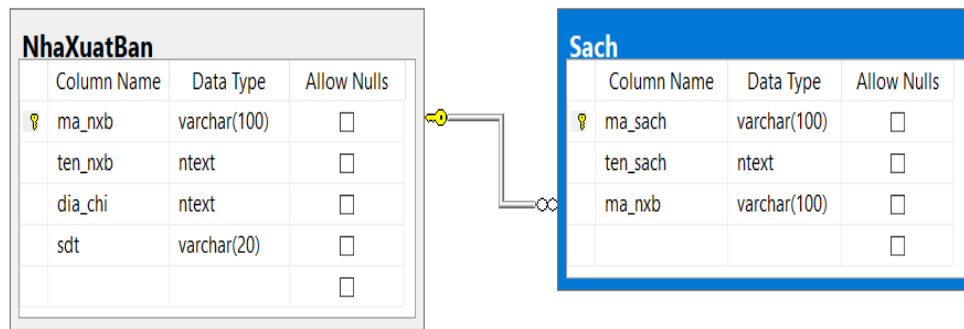
private void txtGiaTuKeyTyped(java.awt.event.KeyEvent evt) {
    try
    {
        String kyTuVuaNhap = evt.getKeyChar() + "";
        Integer.parseInt(kyTuVuaNhap);
    }catch(NumberFormatException ex) {
        evt.consume();
    }
}

private void txtGiaDenKeyTyped(java.awt.event.KeyEvent evt) {
    try {
        String kyTuVuaNhap = evt.getKeyChar() + "";
        Integer.parseInt(kyTuVuaNhap);
    }catch(NumberFormatException ex) {
        evt.consume();
    }
}
}

```


Bài 5. Tổng hợp

Xây dựng ứng dụng quản lý thư viện, thiết kế và cài đặt CSDL trong SQL



The screenshot shows a web application interface for managing publishers. The main section is titled "Thông tin nhà xuất bản" (Publisher Information). It contains four input fields: "Mã nhà xuất bản" (Publisher Code), "Tên nhà xuất bản" (Publisher Name), "Địa chỉ" (Address), and "Điện thoại" (Phone Number). Below these fields are navigation buttons: "Vẽ trước" (Previous), "1/1", and "Vẽ sau" (Next). On the right side, there is a "Thực hiện" (Perform) section with four buttons: "Thêm" (Add), "Lưu" (Save), "Sửa" (Edit), and "Xóa" (Delete). At the bottom, there is a "Danh sách nhà xuất bản" (Publisher List) table with columns: "Mã nhà xuất bản", "Tên nhà xuất bản", "Địa chỉ", and "Số điện thoại". A "Tìm kiếm" (Search) button is located at the bottom left.

Khi thực hiện chức năng tìm kiếm hiện màn hình giao diện sau

The screenshot shows the search results page. At the top, there is a "Tìm kiếm" (Search) section with a text input field labeled "Nhập từ khóa mã nhà xuất bản" (Enter keyword publisher code) and a "Tìm kiếm" (Search) button. Below this is a table with three columns: "Mã sách" (Book Code), "Tên sách" (Book Name), and "Nhà xuất bản" (Publisher). The table is currently empty.

Hình 7. Một số màn hình quản lý Thư viện

Gợi ý: Các chức năng trong màn hình chính tương tự các bài đã làm, đối với chức năng tìm kiếm dùng khóa **like** và **%** trong câu truy vấn tìm kiếm tương đối.

TÀI LIỆU THAM KHẢO

- [1]. Cay S. Horstmann, *Core Java Volum I - Fundamentals, Tenth Edition*, NewYork : Prentice Hall, 2016.
- [2]. Cay S. Horstmann. *Core Java Volum II - Advanced Features, Tenth Edition*, New York : Prentice Hall, 2017.
- [3].Eng.haneen Ei-masry, *Java database connection*, Islamic University of Gaza Faculty of Engineering Department of Computer Engineering ECOM 4113: DataBase Lab, 2014.
- [4]. Angelos Stavrou, *Advanced Network Programming Lab using Java*, Network Security, ISA 656, Angelos Stavrou.
- [5]. Marenglen Biba, Ph.D, *Manual for Lab practices, Remote Method Invocation Three Tier Application with a Database Server*, Department of Comsputer Science, University of New York.
- [6].Elliotte Rusty Harold, *Java Network Programming, Fourth Edition*, O'Reilly Media, 2013.
- [7]. Đoàn Văn Ban, *Lập trình hướng đối tượng với JAVA*, Nhà xuất bản Khoa học và Kỹ thuật, 2005.
- [8]. ThS. Dương Thành Phết, *Bài tập thực hành Chuyên đề 1 CNPM- Java*, Khoa CNTT- Trường ĐH Công nghệ TP.HCM.
- [9]. <https://www.oracle.com/technetwork/java/socket-140484.html#>
- [10]. https://personales.unican.es/corcuerp/java/Labs/LAB_22.htm
- [11]. <http://www.nrcmec.org/pdf/Manuals/CSE/student/2-2%20java16-17.pdf>
- [12]. <http://cse.mait.ac.in/pdf/LAB%20MANUAL/JAVA.pdf>
- [13]. https://www.academia.edu/35283541/Bài_tập_môn_lập_trình_hướng_đối_tượng