

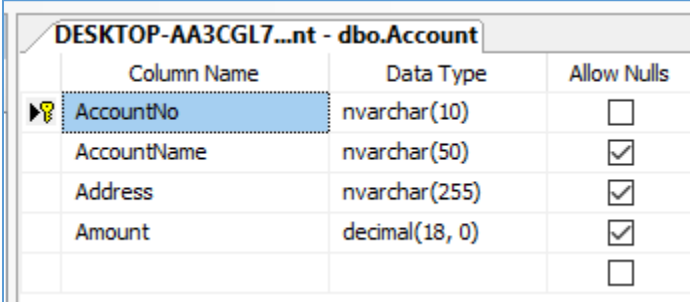
Hướng Dẫn Lab5

- Mục tiêu: Hướng dẫn sử dụng Java Swing với CSDL SQL Server theo mô hình 3 lớp (lớp giao diện, lớp model và lớp db).

Bài Tập 1: Quản lý tài khoản

1.1 Sử dụng MS SQL Server tạo CSDL: *QLTaiKhoan* với 1 table như sau

Account (AccountNo NVARCHAR(10), AccountName NVARCHAR(100), Address NVARCHAR(255) , Amount decimal).



DESKTOP-AA3CGL7...nt - dbo.Account			
	Column Name	Data Type	Allow Nulls
▶	AccountNo	nvarchar(10)	<input type="checkbox"/>
	AccountName	nvarchar(50)	<input checked="" type="checkbox"/>
	Address	nvarchar(255)	<input checked="" type="checkbox"/>
	Amount	decimal(18, 0)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Nhập sẵn 1 số dòng vào CSDL

	AccountNo	AccountName	Address	Amount
	210121121	Nguyen Van A	TP HCM	3000
	213232574	Nguyen Van B	Binh Duong	4000
▶	214454644	Nguyen Van C	Dong Nai	5000
*	NULL	NULL	NULL	NULL

Kết nối CSDL trên với project Java Swing để thực hiện các yêu cầu sau:

1.2 Khi mới load JFrame lấy các dữ liệu tương ứng từ bảng Account để đưa vào jTable

Quản lý tài khoản

Quản lý thông tin tài khoản

Số tài khoản:

Tên khách hàng:

Địa chỉ:

Số tiền:

STT	Mã tài khoản	Tên khách hàng	Địa chỉ	Số tiền
1	210121121	Nguyen Van A	TP HCM	3,000
2	213232574	Nguyen Van B	Binh Duong	4,000
3	214454644	Nguyen Van C	Dong Nai	5,000

Tổng Tiền:

1.3 Khi chọn 1 dòng ở jTable, hiện thị lên các jtextbox ở phía trên tương ứng.

1.4 Khi nhấn vào nút thêm mới.

- Kiểm tra thông tin tất cả thông tin tài khoản được nhập, nếu không xuất thông báo *“Thông tin tài khoản không được để trống!”*
- Thêm tài khoản hiện vừa nhập liệu vào CSDL và load lại form từ CSDL. Xuất thông báo *“Thêm mới thông tin tài khoản thành công!”*

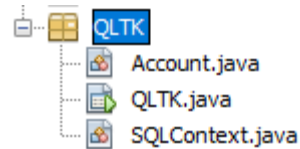
1.5 Khi nhấn vào nút “Sửa”:

- Kiểm tra thông tin tất cả thông tin tài khoản được nhập, nếu không xuất thông báo *“Thông tin tài khoản không được để trống!”*
- Cập nhật tài khoản vừa thay đổi vào CSDL và load lại form từ CSDL. Xuất thông báo *“Cập nhật thông tin tài khoản thành công!”*

1.6 Khi nhấn vào nút Xóa. thì xuất hiện cảnh báo YES/NO “Bạn có muốn xóa không?”

Nhấn YES sẽ thực hiện xóa dòng dữ liệu tài khoản trong JTable và CSDL và thông báo “*Xóa tài khoản thành công!*”. Sau khi xóa thông tin thành công, Xóa các giá trị textbox về empty.

Hướng Dẫn



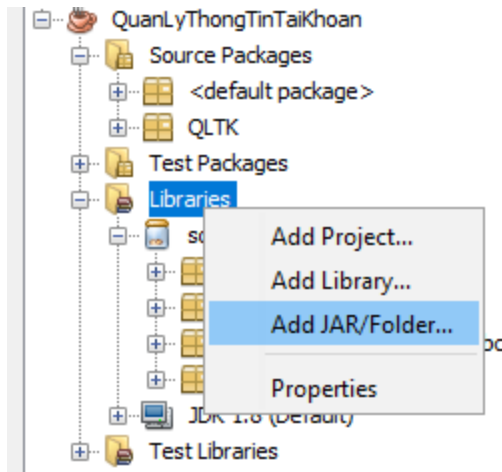
Ý tưởng: Xây dựng 3 lớp và hướng đối tượng: 

Lớp giao diện

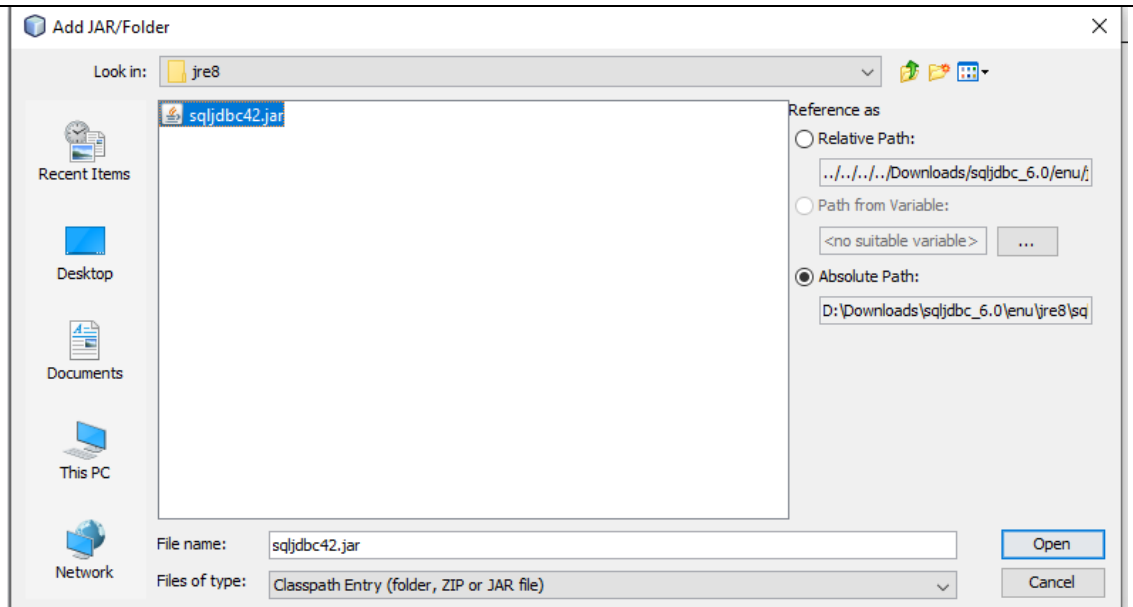
Lớp SQLContext: Sử dụng để kết nối với SQL server, trong đó hỗ trợ 2 hàm executeQuery (trả về danh sách record) và ExcuteUpdate (thực hiện query)

Lớp class Account: Đại diện cho lớp model, trong đó chúng ta có thể viết các hàm GetAll(), Insert, Update, Delete..

- Thực hiện: Cài đặt thư viện từ Driver cho SQLServer:



Chọn thư viện connect với SQL server từ java là sqljdbc42.jar



- Viết 1 Java class để kết nối, Ví dụ SQLContext.java (các hàm có throws exception có nghĩa là khi gặp lỗi sẽ đưa ra ngoài xử lý, việc này giao diện sẽ làm...)

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
public class SQLContext {
    public Connection GetConnection() throws Exception
    {
        String userName ="sa";
        String password = "123456";
        String dbURL =
"jdbc:sqlserver://localhost:1433;databaseName=QLAccount;";
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection conn = java.sql.DriverManager.getConnection(dbURL,
userName,password);
        return conn;
    }
}
```

```
public ResultSet ExcuteQuery(String query) throws Exception
{
    Statement statement = GetConnection().createStatement();
    return statement.executeQuery(query);
}
```

```
public void ExcuteUpdate(String sql) throws Exception
{
    Statement statement = GetConnection().createStatement();
    statement.executeUpdate(sql);
}
}
```

- Viết Java class Account: Có đầy đủ các hàm lấy tất cả, insert, update, delete 😊 vì yêu cầu bài toán có những chức năng này

```
import java.sql.ResultSet;
import java.util.ArrayList;
public class Account {
    public String AccountNo;
    public String AccountName;
    public String Address;
    public double Amount;

    public Account()
    {
    }

    public Account(String no, String name, String add, double amt)
    {
        AccountNo = no;
```

```

        AccountName = name;
        Address = add;
        Amount = amt;
    }

    public static ArrayList<Account> GetListAccount() throws Exception
    {
        ArrayList<Account> list = new ArrayList();
        String query = "select * from Account";
        ResultSet rs = new SQLContext().ExcuteQuery(query);
        while(rs.next())
        {
            Account temp = new Account();
            temp.AccountNo = rs.getString("AccountNo");
            temp.AccountName = rs.getString("AccountName");
            temp.Address= rs.getString("Address");
            temp.Amount = rs.getDouble("Amount");
            list.add(temp);
        }
        return list;
    }

    public void Insert() throws Exception
    {
        String sql = String.format("INSERT INTO Account VALUES('%s', '%s', '%s', %f)", this.AccountNo, this.AccountName, this.Address, this.Amount);
        new SQLContext().ExcuteUpdate(sql);
    }

    public void Update() throws Exception

```

```

    {
        String sql = String.format("UPDATE Account SET AccountName = '%s',
Address = '%s', Amount = %f WHERE AccountNo = '%s'", this.AccountName,
this.Address, this.Amount, this.AccountNo);
        new SQLContext().ExcuteUpdate(sql);
    }
    public static void Delete(String accountNo) throws Exception
    {
        String sql = String.format("DELETE Account WHERE AccountNo =
'%s'", accountNo);
        new SQLContext().ExcuteUpdate(sql);
    }

```

- Bây giờ cứ đọc đề để thực hiện ở lớp giao diện ☺

1.2 Khi mới load JFrame lấy các dữ liệu tương ứng từ bảng Account để đưa vào jTable -> Viết sự kiện event ở ***form WindowOpened***
try

```

    {
        System.out.println("Window Opened Event");
        BindAccount(); //Gọi hàm BindAccount viết ở dưới
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi",
JOptionPane.ERROR_MESSAGE);
    }

```

- Gọi hàm BindAccount: Để load tất cả account hiện có từ CSDL (tách ra 1 hàm mục đích tái sử dụng khi insert,update,delete ☺)

```

private void BindAccount() throws Exception
{

```

```

DefaultTableModel model =
(DefaultTableModel)jtableAccount.getModel();

model.setRowCount(0); //remove all rows

ArrayList<Account> listAccount = Account.GetListAccount();
double totalAmount = 0;
for (int i = 0; i < listAccount.size(); i++)
{
    Account temp = listAccount.get(i);
    model.addRow(new Object[]{ (i+1) , temp.AccountNo ,
temp.AccountName, temp.Address , temp.Amount });
    totalAmount += temp.Amount;
}
txtTotal.setText( totalAmount + "");
}

```

1.3 Khi chọn 1 dòng ở jTable, hiện thị lên các jtextbox ở phía trên tương ứng.

➔ Viết sự kiện ở jTable *MouseClicked*

```

try
{
    DefaultTableModel model = (DefaultTableModel)
jtableAccount.getModel();

    int row = jTableAccount.getSelectedRow();
    txtAccount.setText(model.getValueAt(row, 1).toString()); //lay
AccountNo tu jTable dong row

    txtAccountName.setText(model.getValueAt(row, 2).toString());
    txtAddress.setText(model.getValueAt(row, 3).toString());
    txtAmount.setText(model.getValueAt(row, 4).toString());
}
catch(Exception ex)

```



```

    {
        JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi",
JOptionPane.ERROR_MESSAGE);
    }

```

1.4 Khi nhấn vào nút thêm mới. -> viết sự kiện cho nút thêm mới

```

try
{
    Account temp = GetAccount(); //hàm lấy account từ nhập liệu và kiểm tra
temp.Insert(); //goi ham insert o lop Account
BindAccount();
    JOptionPane.showMessageDialog(this, "Thêm Mới Thành Công", "Thông
Báo", JOptionPane.INFORMATION_MESSAGE);
}
catch(Exception ex)
{
    JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi",
JOptionPane.ERROR_MESSAGE);
}

```

- Implement GetAccount() // mục đích tách ra hàm là tái sử dụng cho 1.5 thôi ☺

```

private Account GetAccount() throws Exception
{
    if(""".equals(txtAccount.getText()) || "".equals(txtAccountName.getText()) ||
"".equals(txtAddress.getText()) )
    {
        throw new Exception("Thông tin tài khoản không được để trống!");
    }
    return new Account(txtAccount.getText(),txtAccountName.getText(),
txtAddress.getText(),Double.parseDouble(txtAmount.getText()) );
}

```

1.5 Khi nhấn vào nút sửa -> Viết sự kiện cho nút sửa

```

try

```

```

        {
            Account temp = GetAccount();
            temp.Update();
            BindAccount();
            JOptionPane.showMessageDialog(this, "Cập nhật tài khoản thành công",
"Thông Báo", JOptionPane.INFORMATION_MESSAGE);
        }
        catch(Exception ex)
        {
            JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi",
JOptionPane.ERROR_MESSAGE);
        }

```

1.6 Nút xóa _> Viết sự kiện cho nút xóa

```

try
{
    int input = JOptionPane.showConfirmDialog(null, "Bạn có muốn xóa không?",
"Cảnh báo", JOptionPane.YES_NO_OPTION);

    // 0=yes, 1=no, 2=cancel
    if(input == 0) //th yes
    {
        Account.Delete(txtAccount.getText());
        BindAccount();

        JOptionPane.showMessageDialog(this, "Xóa tài khoản thành công!", "Thông
Báo", JOptionPane.INFORMATION_MESSAGE);

        //Clear giá trị các bạn từ làm
    }
}
catch(Exception ex)
{
    JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi",
JOptionPane.ERROR_MESSAGE);
}

```

Bài tập 2: Thực hiện bài tập 1 – Lab05. Kết nối với CSDL

Thiết Kế & Cài đặt CSDL: QLSanpham, có 02 Table:

- ✓ LoaiSP(**Maloi char(2)**, Tenloai nvarchar(20))
- ✓ SanPham(**MaSP char(4)**, TenSP nvarchar(20), Dongia BigInt, Maloi char(2))
- ✓ Nhập liệu như sau:

LoaiSP			Sanpham				
	Maloi	Tenloai		MaSP	TenSP	Dongia	Maloi
1	BK	Bánh kẹo	1	SP01	Bánh mì	10000	BK
2	GK	Giải khát	2	SP02	Bánh bao	15000	BK
3	MP	Mỹ phẩm	3	SP03	Coca c...	12000	GK
			4	SP04	Pepsi	11000	GK
			5	SP05	Kem ch...	85000	MP

Viết Chương trình Java Swing thực hiện chương trình quản lý danh mục sản phẩm

2.1 Thiết kế giao diện quản lý tương tự như sau cho danh mục sản phẩm

Mã SP	Tên Sản Phẩm	Giá bán	Tên Loại	Mã Loại
SP01	Bánh mì	10000	Bánh kẹo	BK
SP02	Bánh Bao	15000	Bánh kẹo	BK
SP03	Coca cola	12000	Giải khát	GK
SP04	Pepsi	13000	Giải khát	GK
SP05	Kem	85000	Mỹ phẩm	MP

2.2 Khi Load form Thực hiện

- Lấy tất cả dữ liệu Loại Sản phẩm từ CSDL bảng “LoaiSP” vào JComboBox (hiện thị đúng format yêu cầu “Mã Loại – Tên Loại”)

- Lấy tất cả dữ liệu từ bảng “SanPham” hiện có để đưa vào jTable như trên

2.3 Khi chọn vào 1 dòng dữ liệu ở jTable. Hiện thị ngược lại các thông tin vừa chọn

2.4 Khi Click vào nút thêm mới / Update

- Kiểm tra tất cả thông tin bắt buộc phải nhập cho sản phẩm. Nếu không xuất hiện thông báo “*Vui lòng nhập đầy đủ thông tin sản phẩm!*”
- Kiểm tra giá tiền nhập trong 1-99999999. Nếu không xuất thông báo “*Giá tiền cho sản phẩm từ 1-99999999*”.
- Thêm mới / Update dữ liệu nhập vào CSDL. Cập nhật lại jTable

2.5 Khi Click vào nút xóa

- Hiện thị cảnh báo YES/NO “Bạn có muốn xóa không?”
- Nhấn YES: Xóa dữ liệu sản phẩm và cập nhật lại jTable

2.6 Khi hiện thị lên jTable không hiện thị mã loại. Chỉnh sửa lại chương trình để phù hợp với yêu cầu này ??? (Tự làm)

HƯỚNG DẪN

Ý Tưởng thực hiện theo mô hình 3 lớp và hướng đối tượng

- Lớp giao diện: kéo vào jtextbox, jcombobox, jTable...
- Lớp SQLContext: Sử dụng để kết nối với SQL server, trong đó hỗ trợ 2 hàm executeQuery (trả về danh sách record) và ExcuteUpdate (thực hiện query).
SV nhớ đổi ***database name*** là xong thôi , copy ☺
- Lớp class SanPham, LoaiSanPham: tương ứng các trường như ở database
- + Trong class LoaiSanPham: chúng ta cần viết hàm *lấy tất cả loại sản phẩm để trả về tất cả các loại sản phẩm mục đích đổ vào jcombobox*
- + Trong class SanPham: có lấy tất cả, insert, update, delete. Vì ở jTable cũng có yêu cầu hiển thị cả tên loại (nhưng tên loại lại ở LoaiSP) nên ta đổi lại câu query khi lấy tất cả bằng cách join với LoaiSP ☺

Lưu ý: Nhớ add thư viện sqjdbc42.jar vào library vì có sử dụng SQL.

HƯỚNG DẪN CÓ THỂ CHI TIẾT HƠN Ở DƯỚI ☺



- **Lớp Sản phẩm**

```
import java.sql.ResultSet;
import java.util.ArrayList;

public class SanPham {
    public String MaSP;
    public String TenSP;
    public long DonGia;
    public String MaLoai;

    public String TenLoai; //vi yeu cau hien thi TenLoai khi load Jtable
    public SanPham()
    {

    }

    public SanPham(String masp, String ten, long gia, String ml)
    {
        MaSP = masp;
        TenSP = ten;
        DonGia = gia;
        MaLoai = ml;
    }

    public static ArrayList<SanPham> GetListSanPham() throws Exception
    {
        ArrayList<SanPham> list = new ArrayList();
```

```

String query = "select * from SanPham left join LoaiSP on Sanpham.Maloai = LoaiSP.MaLoai";
ResultSet rs = new SQLContext().ExcuteQuery(query);
while(rs.next())
{
    SanPham temp = new SanPham();

    temp.MaSP = rs.getString("MaSP");
    temp.TenSP = rs.getString("TenSP");
    temp.DonGia= Long.parseLong(rs.getString("DonGia"));
    temp.MaLoai = rs.getString("MaLoai");
    temp.TenLoai = rs.getString("TenLoai");
    list.add(temp);
}
return list;
}

public void Insert() throws Exception
{
    String sql = String.format("INSERT INTO SanPham VALUES('%s', '%s', %s, '%s')", this.MaSP, this.TenSP,
this.DonGia, this.MaLoai);
    new SQLContext().ExcuteUpdate(sql);
}

public void Update() throws Exception
{
    String sql = String.format("UPDATE SanPham SET TenSP = '%s', DonGia = %s, MaLoai = '%s' WHERE MaSP = '%s'",
this.TenSP, this.DonGia, this.MaLoai, this.MaSP);
    new SQLContext().ExcuteUpdate(sql);
}

public static void Delete(String maSP) throws Exception
{
    String sql = String.format("DELETE SanPham WHERE MaSP = '%s'", maSP);
    new SQLContext().ExcuteUpdate(sql);
}
}

```

- ***Lớp Loại Sản Phẩm***

```

public class LoaiSanPham {

```

```

public String MaLoai;
public String TenLoai;
public LoaiSanPham()
{
}

public LoaiSanPham(String ml, String tl)
{
    MaLoai = ml;
    TenLoai = tl;
}

public static ArrayList<LoaiSanPham> GetListLoaiSanPham() throws Exception
{
    ArrayList<LoaiSanPham> list = new ArrayList();
    String query = "select * from LoaiSP";
    ResultSet rs = new SQLContext().ExcuteQuery(query);
    while(rs.next())
    {
        LoaiSanPham temp = new LoaiSanPham();
        temp.MaLoai = rs.getString("MaLoai");
        temp.TenLoai = rs.getString("TenLoai");
        list.add(temp);
    }
    return list;
}

@Override
public String toString() {
    return this.MaLoai + " - " + this.TenLoai;
}
}

```

- ***frmQLSanPham.java: Đoạn SAU này ko copy được đâu chịu khó gõ và nghĩ ☹***

2.2 Sự kiện Load-Form: formWindowOpened

```

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    try
    {
        System.out.println("Window Opened Event");
        BindLoaiSP();
        BindJTable();
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi", JOptionPane.ERROR_MESSAGE);
    }
}

private void BindJTable() throws Exception
{
    DefaultTableModel model = (DefaultTableModel) jTableSP.getModel();
    model.setRowCount(0); //remove all rows
    ArrayList<SanPham> listSP = SanPham.GetListSanPham();
    for (int i = 0; i < listSP.size(); i++)
    {
        SanPham temp = listSP.get(i);
        model.addRow(new Object[]{ temp.MaSP , temp.TenSP, temp.DonGia , temp.TenLoai,temp.MaLoai });
    }
}

private void BindLoaiSP() throws Exception
{
    DefaultComboBoxModel model = new DefaultComboBoxModel();
    ArrayList<LoaiSanPham> listLoaiSP = LoaiSanPham.GetListLoaiSanPham();
    for (int i = 0; i < listLoaiSP.size(); i++)
    {
        model.addElement(listLoaiSP.get(i));
    }
    cmbLoaiSanPham.setModel(model);
}

```

2.3 Sự kiện chọn 1 dòng

```

private void jTableSPMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    try
    {
        DefaultTableModel model = (DefaultTableModel) jTableSP.getModel();
        int row = jTableSP.getSelectedRow();
        txtMaSP.setText(model.getValueAt(row, 0).toString());
        txtTenSP.setText(model.getValueAt(row, 1).toString());
        txtGiaBan.setText(model.getValueAt(row, 2).toString());

        LoaiSanPham temp = new LoaiSanPham(model.getValueAt(row,4).toString(),model.getValueAt(row, 3).toString());
        // cmbLoaiSanPham.setSelectedItem(temp);
        ArrayList<LoaiSanPham> listLoaiSP = LoaiSanPham.GetListLoaiSanPham();
        for (int i = 0; i < listLoaiSP.size(); i++)
        {
            if(listLoaiSP.get(i).MaLoai.equals(temp.MaLoai))
            {
                cmbLoaiSanPham.setSelectedIndex(i);
                break;
            }
        }
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi", JOptionPane.ERROR_MESSAGE);
    }
}

```


2.4 Insert / Update

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try  
    {  
        SanPham temp = GetSP();  
        temp.Insert(); //gọi hàm insert ở SanPham  
        BindJTable();  
        JOptionPane.showMessageDialog(this, "Thêm Mới Sản phẩm Thành Công", "Thông Báo", JOptionPane.INFORMATION_MESSAGE);  
    }  
    catch (Exception ex)  
    {  
        JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi", JOptionPane.ERROR_MESSAGE);  
    }  
}  
  
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try  
    {  
        SanPham temp = GetSP();  
        temp.Update();  
        BindJTable();  
        JOptionPane.showMessageDialog(this, "Cập nhật Sản phẩm thành công", "Thông Báo", JOptionPane.INFORMATION_MESSAGE);  
    }  
    catch (Exception ex)  
    {  
        JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

- **Khi lấy ra đối tượng trước đó kiểm tra luôn để xuất thông báo**

```
private SanPham GetSP() throws Exception  
{  
    if("".equals(txtMaSP.getText()) || "".equals(txtTenSP.getText()) || "".equals(txtGiaBan.getText()))  
    {  
        throw new Exception("Vui lòng nhập đầy đủ thông tin sản phẩm!");  
    }  
    long gt = Long.parseLong(txtGiaBan.getText());  
    if(! (gt > 1 && gt < 99999999))  
        throw new Exception("Giá tiền cho sản phẩm từ 1-99999999");  
    return new SanPham(txtMaSP.getText(), txtTenSP.getText(), gt, ((LoaiSanPham)cmbLoaiSanPham.getSelectedItem()).MaLoai);  
}
```

2.5 Xóa

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try  
    {  
        int input = JOptionPane.showConfirmDialog(null, "Bạn có muốn xóa không?", "Cảnh báo", JOptionPane.YES_NO_OPTION);  
        // 0=yes, 1=no, 2=cancel  
        if(input == 0) //th yes  
        {  
            SanPham.Delete(txtMaSP.getText());  
            BindJTable();  
            JOptionPane.showMessageDialog(this, "Xóa Sản phẩm thành công!", "Thông Báo", JOptionPane.INFORMATION_MESSAGE);  
        }  
    }  
    catch (Exception ex)  
    {  
        JOptionPane.showMessageDialog(rootPane, ex.getMessage(), "Lỗi", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

Nhận xét: chắc các bạn đã có khả năng bịa ra được nhiều kiểu bài thế này ???