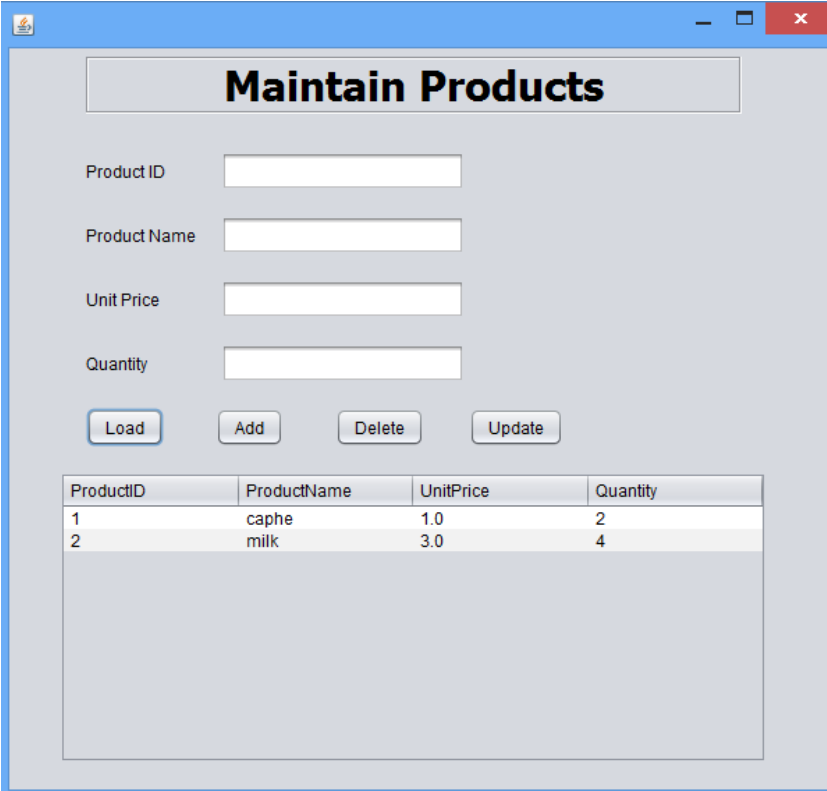


**Sau khi thực hành xong bài này, học viên có khả năng :**

- Xây dựng được ứng dụng Swing sử dụng JDBC kết nối cơ sở dữ liệu SQL Server.
- Trình bày được các bước phát triển ứng dụng theo kiến trúc 2 lớp trong Java

Xây dựng ứng dụng Swing có giao diện sau đây và thực hiện các chức năng : lấy danh sách các mặt hàng, thêm mới , xoá , sửa thông tin các mặt hàng .Ứng dụng được phát triển theo kiến trúc 2 lớp.



ProductID	ProductName	UnitPrice	Quantity
1	caphe	1.0	2
2	milk	3.0	4

**Bước 1** . Mở ứng dụng NetBean , New Project | Java | Java Application , nhấn Next , trên hộp thoại “New Java Application ” điền các thông tin trong hộp thoại như sau :  
Project Name : **ASM\_Ngay\_1\_JDBC** ; project Location : “nhập đường dẫn lưu ứng dụng” và **Bỏ chọn** “Create Main Class” , nhấn Finish để thúc .

1.1 Nhấp phải chuột trên **<default package>** chọn New | Java Class , đặt tên lớp : **Product** , nhấn Finish để kế thúc .

- 1.2 Nhấp phải chuột trên **<default package>** chọn New | Java Class , đặt tên lớp : **ProductData**, nhấp Finish để kết thúc
- 1.3 Nhấp phải chuột trên **<default package>** chọn New | JFrame Form , đặt tên lớp : **MaintainProducts**, nhấp Finish để kết thúc
- 1.4 Mở ứng dụng SQL Server 2008/2012 , tạo cơ sở dữ liệu tên SaleManager gồm một bảng tên **Products** có cấu trúc như sau :

T430.SaleManager - dbo.Products X			
	Column Name	Data Type	Allow Nulls
PK	ProductID	int	<input type="checkbox"/>
	ProductName	nvarchar(50)	<input checked="" type="checkbox"/>
	UnitPrice	float	<input checked="" type="checkbox"/>
	Quantity	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- 1.5 Từ project , nhấp phải chuột trên **Libraries** | Add JAR / Folder... , chọn đường dẫn đến tập tin **sqljdbc4.jar** đã có trên đĩa và nhấp Open để thêm vào project .

## **Bước 2.** Phát triển các lớp cho tầng nghiệp vụ

- 2.1 Phát triển lớp **Product** , viết mã cho tập tin **Product.java** như sau :

```
public class Product {
    private int ProductID;
    private String ProductName;
    private float UnitPrice;
    protected int Quantity;

    public Product(int ProductID) {
        this.ProductID = ProductID;
    }

    public Product(int ProductID, String ProductName, float UnitPrice, int Quantity) {
        this.ProductID = ProductID;
        this.ProductName = ProductName;
        this.UnitPrice = UnitPrice;
        this.Quantity = Quantity;
    }
}
```

```

    public int getQuantity() {
        return Quantity;
    }
    public void setQuantity(int Quantity) {
        this.Quantity = Quantity;
    }
    public float getUnitPrice() {
        return UnitPrice;
    }
    public void setUnitPrice(float UnitPrice) {
        this.UnitPrice = UnitPrice;
    }
    public String getProductName() {
        return ProductName;
    }
    public void setProductName(String ProductName) {
        this.ProductName = ProductName;
    }
    public int getProductID() {
        return ProductID;
    }
    public void setProductID(int ProductID) {
        this.ProductID = ProductID;
    }
}

```

2.2 Phát triển lớp **ProductData** để truy cập dữ liệu , viết mã cho tập tin

**ProductData.java** như sau :

```

import java.sql.*;
import java.util.Vector;
/**
 *
 * @author MichaelLake
 */
public class ProductData {
    public static Connection getConnection() throws Exception{
        String url = "jdbc:sqlserver://localhost:1434;database=SaleManager";
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection cnn = DriverManager.getConnection(url,"sa","123");
        return cnn;
    }
}

```

```

public static Vector GetProductList() throws Exception{
    Connection cnn = null;
    try{
        cnn = GetConnection();
        String strSQL = "select * from Products";
        Vector<Vector<String>> productList = new Vector<Vector<String>>();
        PreparedStatement pre = cnn.prepareStatement(strSQL);
        ResultSet rs = pre.executeQuery();
        while(rs.next())
        {
            Vector<String> productDetail = new Vector<String>();
            productDetail.add(rs.getString(1));
            productDetail.add(rs.getString(2));
            productDetail.add(rs.getString(3));
            productDetail.add(rs.getString(4));
            productList.add(productDetail);
        }
        return productList;
    }
    catch(Exception ex){
        throw new Exception(ex.getMessage());
    }
    finally{
        cnn.close();
    }
}

public static boolean InsertProduct(Product NewProduct) throws Exception{
    Connection cnn = null;
    boolean result = false;
    try{
        cnn = GetConnection();
        String strSQL = "insert into Products values(?,?,?,?)";
        PreparedStatement ps = cnn.prepareStatement(strSQL);
        ps.setInt(1,NewProduct.getProductID());
        ps.setString(2,NewProduct.getProduct_name());
        ps.setFloat(3,NewProduct.getUnitPrice());
        ps.setInt(4,NewProduct.getQuantity());
        result = (ps.executeUpdate() > 0);
        ps.close();
    }
}

```

```
        catch (Exception ex) {
            throw new Exception(ex.getMessage());
        }
        finally{
            cnn.close();
        }
        return result ;
    }

    public static boolean UpdateProduct(Product NewProduct) throws Exception{
        Connection cnn = null;
        boolean result = false;
        try{
            cnn = GetConnection();
            String strSQL = "update Products set ProductName= ?,UnitPrice = ?,Quantity = ? "
                + " where ProductID = ?";
            PreparedStatement ps = cnn.prepareStatement(strSQL);
            ps.setString(1,NewProduct.getProductName());
            ps.setFloat(2,NewProduct.getUnitPrice());
            ps.setInt(3,NewProduct.getQuantity());
            ps.setInt(4,NewProduct.getProductID());
            result = (ps.executeUpdate() > 0);
            ps.close();
        }
        catch (Exception ex){
            throw new Exception(ex.getMessage());
        }
        finally{
            cnn.close();
        }
        return result;
    }
}
```

```

public static boolean DeleteProduct(Product NewProduct) throws Exception{
    Connection cnn = null;
    boolean result = false;
    try{
        cnn = GetConnection();
        String strSQL = "Delete Products where ProductID = ?";
        PreparedStatement ps = cnn.prepareStatement(strSQL);
        ps.setInt(1,NewProduct.getProductID());
        result = (ps.executeUpdate() > 0);
        ps.close();
    }
    catch(Exception ex){
        throw new Exception(ex.getMessage());
    }
    finally{
        cnn.close();
    }
    return result;
}
}

```

**Bước 3** Phát triển ứng dụng MaintainProducts , thiết kế giao diện cho **MaintainProducts.java** :

STT	Loại Control	Tên thuộc tính/ Tên Sự Kiện	Giá trị thiết lập
<b>1</b>	JLabel	text	Maintain Products
		font	Tahoma , size 30 , Bold
		border	EtchedBoder
		horizontalAlignment	CENTER
<b>2</b>	JLabel	text	Product ID
<b>3</b>	JLabel	text	Product Name
<b>4</b>	JLabel	text	UnitPrice
<b>5</b>	JLabel	text	Quantity
<b>6</b>	JTextField	text	
		Variable Name	txtProductID

7	JTextField	text	
		Variable Name	txtProductName
8	JTextField	text	
		Variable Name	txtUnitPrice
9	JTextField	text	
		Variable Name	txtQuantity
10	JTable	Variable Name	productTable
		Event	mouseClicked
11	JButton	text	Load
		Variable Name	btnLoad
		Event	ActionPerform
12	JButton	text	Add
		Variable Name	btnAdd
		Event	ActionPerform
13	JButton	text	Delete
		Variable Name	btnDelete
		Event	ActionPerform
14	JButton	text	Update
		Variable Name	btnUpdate
		Event	ActionPerform

Viết mã cho **MaintainProducts.java** :

```
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author MichaelLake
 */
public class MaintainProducts extends javax.swing.JFrame {

    public MaintainProducts() {
        initComponents();
    }
}
```

.....

//Phuong thuc hien thi danh sach cac mat hang trong jTable : productTable

```
public void ViewProduct() {
    //Khai bao cot
    Vector columnHeader = new Vector();
    columnHeader.add("ProductID");
    columnHeader.add("ProductName");
    columnHeader.add("UnitPrice");
    columnHeader.add("Quantity");
    Vector<Vector<String>> productList;
    try {
        //goi phuong thuc lay vecto du lieu
        productList = ProductData.GetProductList();
        DefaultTableModel tableModel = new DefaultTableModel();
        tableModel.setDataVector(productList, columnHeader);
        productTable.setModel(tableModel);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}

private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int ProductID = Integer.parseInt(txtProductID.getText());
        String ProductName= txtProductName.getText();
        float UnitPrice = Float.parseFloat(txtUnitPrice.getText());
        int Quantity = Integer.parseInt(txtQuantity.getText());
        Product p = new Product(ProductID, ProductName, UnitPrice, Quantity);
        boolean result = ProductData.UpdateProduct(p);
        if (result) {
            ViewProduct();
            JOptionPane.showMessageDialog(this, "Update successful.");
        }
    }
    catch (Exception ex){
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}
```



```

private void productTableMouseClicked(java.awt.event.MouseEvent evt) {
    int rowIndex = productTable.getSelectedRow();
    txtProductID.setText(productTable.getValueAt(rowIndex, 0).toString());
    txtProductName.setText(productTable.getValueAt(rowIndex, 1).toString());
    txtUnitPrice.setText(productTable.getValueAt(rowIndex, 2).toString());
    txtQuantity.setText(productTable.getValueAt(rowIndex, 3).toString());
}

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    int ProID = Integer.parseInt(txtProductID.getText());
    try{
        Product p = new Product(ProID);
        if(ProductData.DeleteProduct(p)){
            ViewProduct();
            JOptionPane.showMessageDialog(this, "Delete successful.");
        }
    }
    catch (Exception ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        int ProductID = Integer.parseInt(txtProductID.getText());
        String ProductName= txtProductName.getText();
        float UnitPrice = Float.parseFloat(txtUnitPrice.getText());
        int Quantity = Integer.parseInt(txtQuantity.getText());
        Product p = new Product(ProductID, ProductName, UnitPrice, Quantity);
        boolean result = ProductData.InsertProduct(p);
        if (result) {
            ViewProduct();
            JOptionPane.showMessageDialog(this, "Insert successful.");
        }
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}

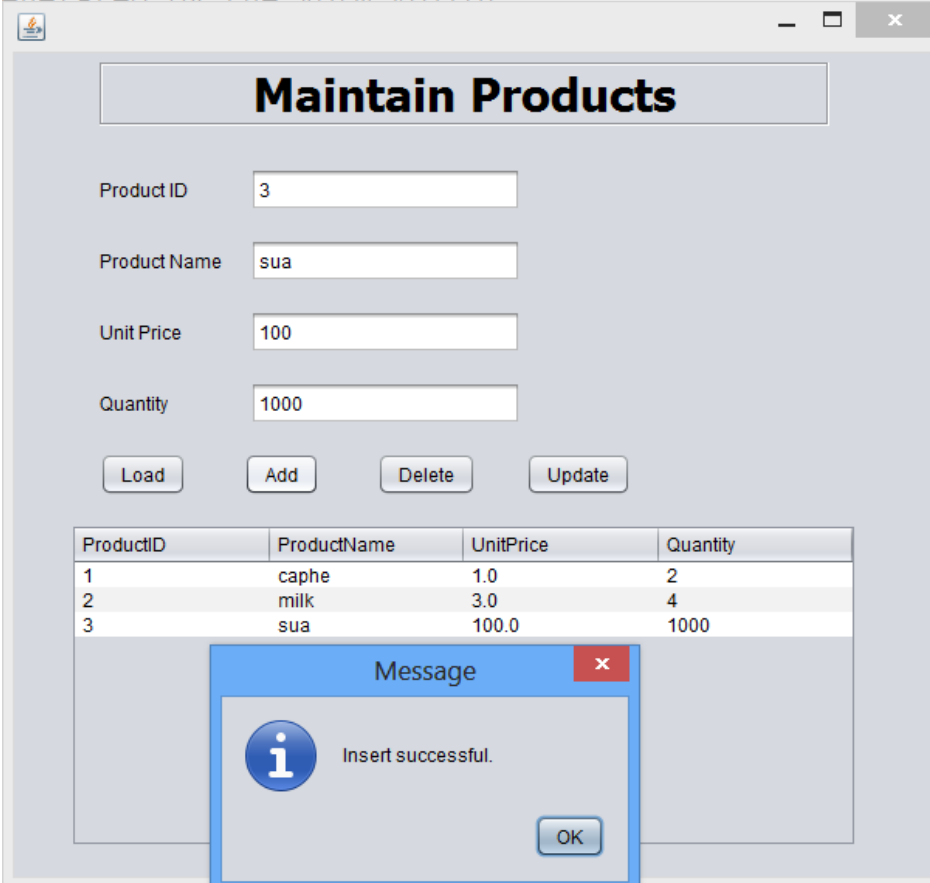
private void btnLoadActionPerformed(java.awt.event.ActionEvent evt) {
    ViewProduct();
}

```

.....

} // end class

**Bước 4:** Chạy ứng dụng và kiểm tra các chức năng .



**Maintain Products**

Product ID:


Product Name:

Unit Price:

Quantity:

ProductID	ProductName	UnitPrice	Quantity
1	caphe	1.0	2
2	milk	3.0	4
3	sua	100.0	1000

**Message**

 Insert successful.