

BÀI THỰC HÀNH SỐ 5

I. Mục tiêu.

Hiểu và sử dụng được thư viện Java JDBC API để kết nối và truy xuất dữ liệu từ trình quản trị CSDL MySQL kết hợp với Swing.

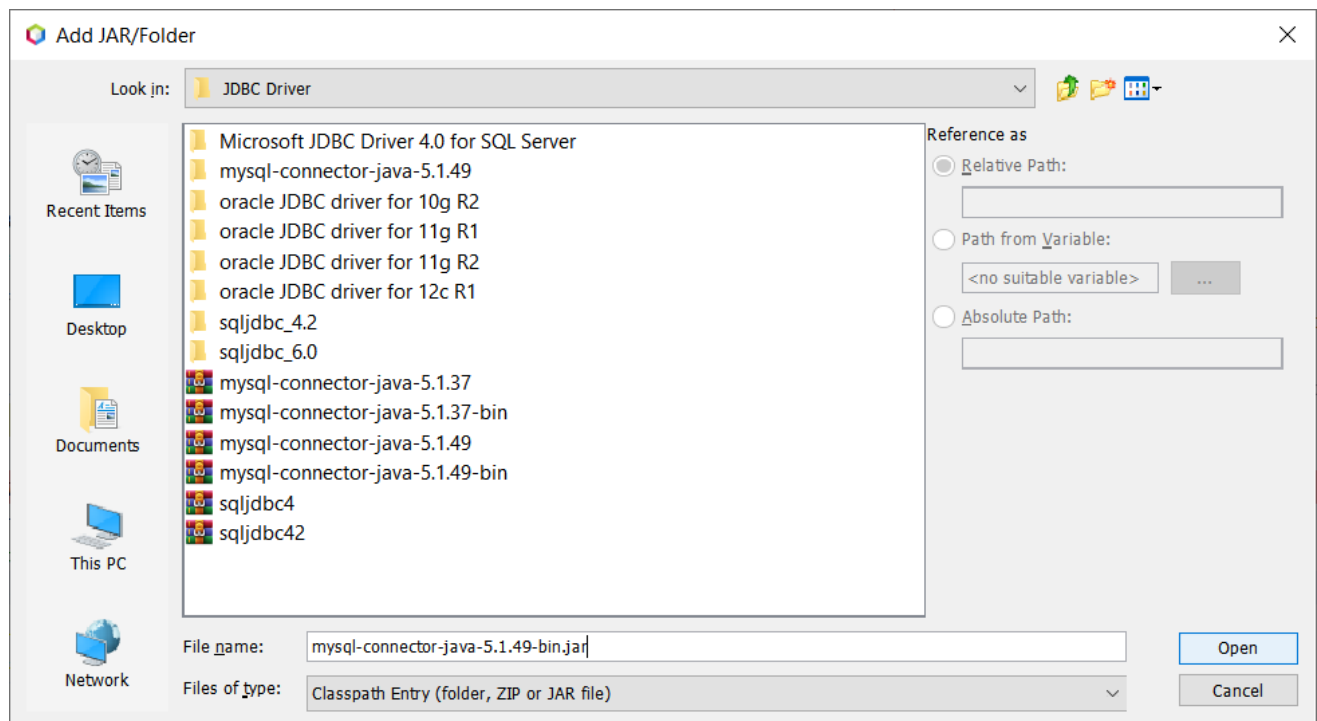
II. Thời gian

- Hướng dẫn chung và thực hành tại lớp: 5 tiết
- Bài tập về nhà: 10 tiết

III. Hướng dẫn chung

Câu 1: Kết nối với CSDL MySQL; sử dụng lớp **Connection** và cung cấp chuỗi kết nối.

- o **Bước 1:** Tải thư viện mysql-connector-java (tham khảo link: <https://dev.mysql.com/downloads/connector/j/5.1.html>). Thêm thư viện vào project bằng cách: chọn chuột phải tại thư mục Libraries của project => Add JAR/Folder...
Tại hộp thoại Add JAR/Folder, chọn thư viện mysql-connector-java (file .jar).



- **Bước 2:** Đăng ký driver cho project: `Class.forName("com.mysql.jdbc.Driver");`

- **Bước 3:** Tạo thông số kết nối CSDL với tên CSDL, và user, password nếu có.

```
String url = "jdbc:mysql://localhost:3306/world?user=root&password=&useUnicode=true&characterEncoding=utf8";
```

- **Bước 4:** Dùng phương thức `getConnection` của `DriverManager` để khởi tạo đối tượng kết nối `Connection`

```
Connection con = DriverManager.getConnection(url);
```

- Kết quả:

```
Connection con = null;
String url = "jdbc:mysql://localhost:3306/world?user=root&password=
&useUnicode=true&characterEncoding=utf8";
String driver = "com.mysql.jdbc.Driver";
try{
    Class.forName(driver);
    con = DriverManager.getConnection(url);
    System.out.println("Connected to the database");
    return true;
}catch(Exception e){
    e.printStackTrace();
    return false;
}
```

Câu 2: Sử dụng lớp `Statement` trong JDBC để tạo câu lệnh và thực thi các truy vấn thêm, xóa và sửa dữ liệu.

- Thiết kế CSDL QLSP với bảng SANPHAM có các thuộc tính:

- MASP: char(4), primary key

- TENS: varchar(20)

- GIA: BigInt

- Xây dựng giao diện để hiển thị thông tin sản phẩm và các button thực hiện chức năng thêm, xóa, sửa và thoát

- Khi chạy chương trình, sẽ lấy dữ liệu trong CSDL hiển thị vào JTable như hình.

San Pham

Mã Sản Phẩm

Tên Sản Phẩm

Giá

Mã sản phẩm	Tên sản phẩm	Giá
1234	ABC	10000
2345	XYZ	20000

Thêm Xóa Sửa Thoát

```

public void TaoTable(){
    //Tạo jTable với DefaultTableModel
    tblModelTT = new DefaultTableModel();
    //Tạo tiêu đề cho DefaultTableModel
    String tieuDe[] = {"Mã sản phẩm", "Tên sản phẩm", "Giá"};
    tblModelTT.setColumnIdentifiers(tieuDe);
    Connection con = new ConnectDB().createCon();
    //Lấy dữ liệu từ CSDL gán vào DefaultTableModel tblModelTT
    try{
        String row[] = new String[3];
        String strSQL = "Select * from SANPHAM";
        Statement stat = con.createStatement();
        ResultSet rs = stat.executeQuery(strSQL);
        while (rs.next()){
            row[0] = rs.getString(1);
            row[1] = rs.getString(2);
            row[2] = rs.getString(3);
            tblModelTT.addRow(row);
        }
    }
    catch(SQLException e){
        System.out.println(e);
        System.out.println("Lỗi");
    }
    //Gán tblModelTT vào jTable tbThongtin
    tbThongtin.setModel(tblModelTT);
    setVisible(true);
}

```

○ Thêm sản phẩm:

- Nhập dữ liệu vào các JTextField **Mã Sản Phẩm**, **Tên Sản Phẩm** và **Giá**.
- Chọn nút “Thêm”, dữ liệu sẽ được đưa xuống jTable và lưu xuống CSDL.

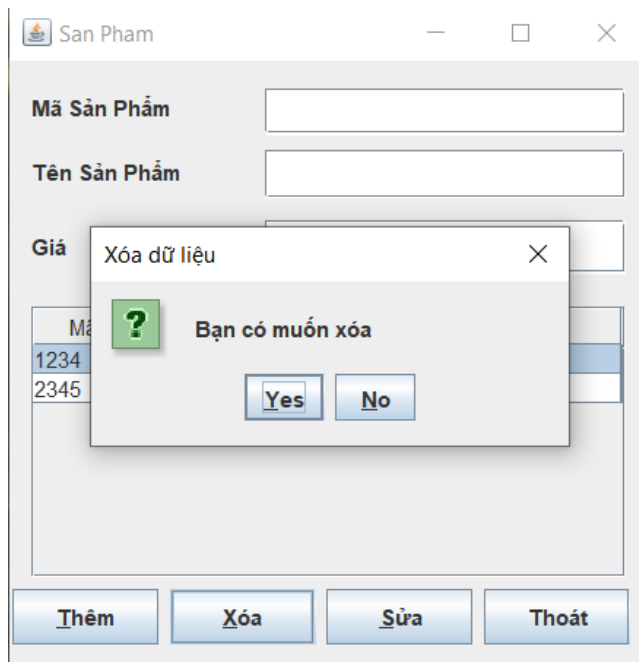
```
private void btThemActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String row[] = new String[3];  
    //Lấy các giá trị ở TextField để đưa vào mảng  
    row[0] = txtMaSP.getText();  
    row[1] = txtTenSP.getText();  
    row[2] = txtGia.getText();  
    //Thêm một dòng mới vào DefaultTableModel  
    tblModelTT.addRow(row);  
    //Thêm xuống CSDL  
    Connection con = new ConnectDB().createCon();  
    String sqlString = "insert SANPHAM Value(?,?,?)";  
    try{  
        PreparedStatement pres = con.prepareStatement(sqlString);  
        pres.setString(1,txtMaSP.getText());  
        pres.setString(2,txtTenSP.getText());  
        pres.setString(3,txtGia.getText());  
        pres.executeUpdate();  
        con.close();  
    }  
    catch (Exception e){  
        e.printStackTrace();  
        System.out.println("Lỗi!");  
    }  
}
```

○ Xóa sản phẩm:

- Chọn sản phẩm muốn xóa trên jTable => thông tin sản phẩm sẽ hiển thị vào các JTextField.

```
private void tbThongtinMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    int indexTB = tbThongtin.getSelectedRow();  
    if (indexTB < tblModelTT.getRowCount() && indexTB >=0)  
        txtMaSP.setText(tblModelTT.getValueAt(indexTB, 0).toString());  
        txtTenSP.setText(tblModelTT.getValueAt(indexTB, 1).toString());  
        txtGia.setText(tblModelTT.getValueAt(indexTB, 2).toString());  
}
```

- Chọn Xóa => hiển thị hộp thoại xác nhận việc xóa.



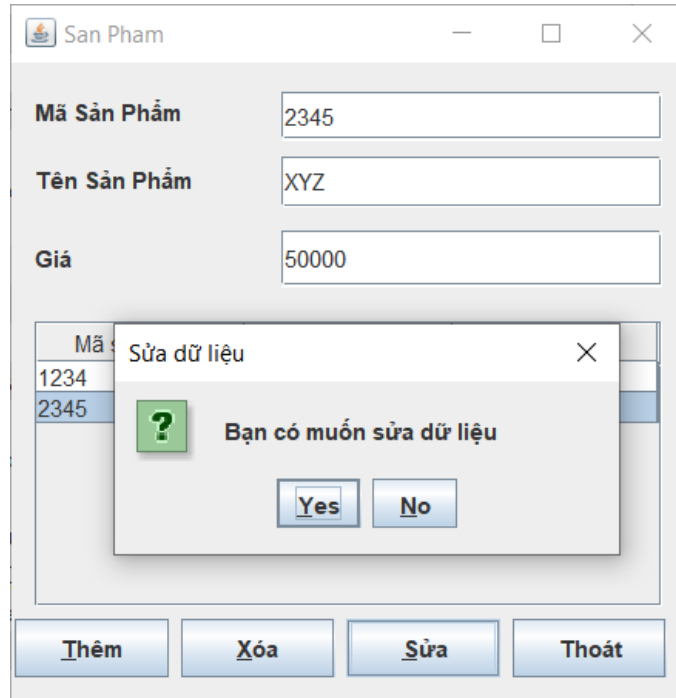
- Chọn “Yes” sẽ xóa khỏi CSDL.

```
private void btXoaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //Lấy vị trí đang chọn trên JTable
    int indexTB = tblThongtin.getSelectedRow();
    Connection con = new ConnectDB().createCon();
    int ret = JOptionPane.showConfirmDialog(null, "Bạn có muốn xóa",
        "Xóa dữ liệu", JOptionPane.YES_NO_OPTION);
    if (ret == JOptionPane.YES_OPTION) {
        //Xóa dòng đang chọn ra khỏi JTable
        if (indexTB < tblModelTT.getRowCount() && indexTB >=0)
            tblModelTT.removeRow(indexTB);
        String str = "Delete from SANPHAM where MASP = ?";
        try{
            PreparedStatement pres = con.prepareStatement(str);
            pres.setString(1, txtMaSP.getText());
            pres.executeUpdate();
            con.close();
        }
        catch (SQLException e){
            System.out.println(e);
        }
    }
}
```

○ Sửa sản phẩm

- Chọn sản phẩm muốn sửa trên JTable, thông tin sản phẩm sẽ hiển thị vào các JTextfield.

- Sửa thông tin trên JTextField (Lưu ý: Không sửa Mã Sản Phẩm) => Chọn nút Sửa => hiển thị hộp thoại xác nhận việc sửa.

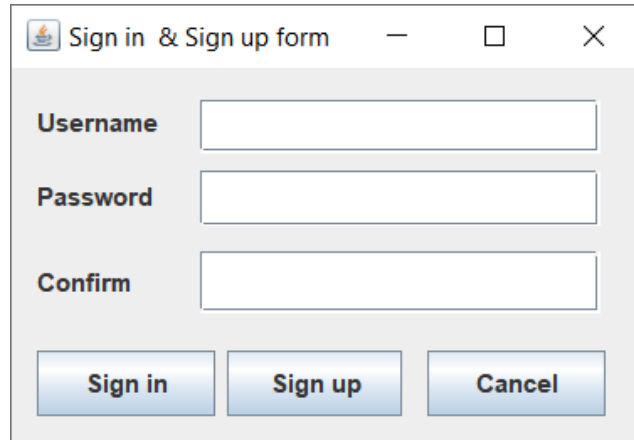


- Chọn "Yes" sẽ sửa dữ liệu trong CSDL.

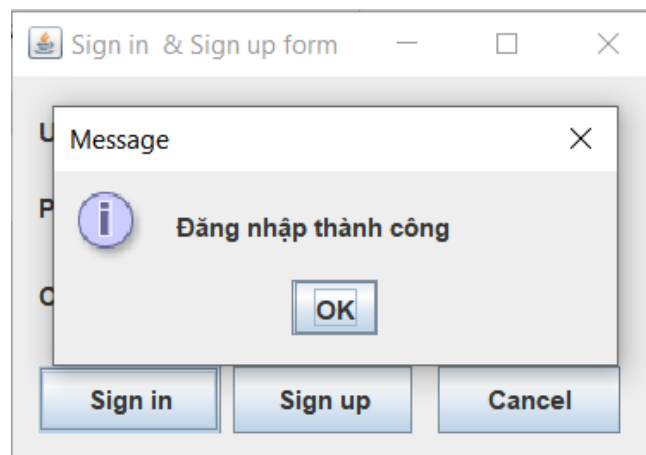
```
private void btSuaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Connection con = new ConnectDB().createCon();
    int ret = JOptionPane.showConfirmDialog(null, "Bạn có muốn sửa dữ liệu",
        "Sửa dữ liệu", JOptionPane.YES_NO_OPTION);
    if (ret == JOptionPane.YES_OPTION) {
        String sql = "Update SANPHAM Set TENSP = ?, GIA = ? where MASP = ?";
        try {
            PreparedStatement prest = con.prepareStatement(sql);
            prest.setString(1, txtTenSP.getText());
            prest.setInt(2, Integer.parseInt(txtGia.getText()));
            prest.setString(3, txtMaSP.getText());
            prest.executeUpdate();
            int indexTB = tbThongtin.getSelectedRow();
            if (indexTB < tblModelTT.getRowCount() && indexTB >= 0) {
                tblModelTT.setValueAt(txtTenSP.getText(), indexTB, 1);
                tblModelTT.setValueAt(txtGia.getText(), indexTB, 2);
            }
        } catch (SQLException e) {
            System.out.println(e);
        }
    }
}
```

IV. Thực hành

Câu 1: Xây dựng giao diện đăng ký và đăng nhập tài khoản theo mẫu.



- Thiết kế bảng USER trong CSDL QLSP có các thuộc tính:
 - USERNAME: varchar(10), primary key
 - PASSWORD: varchar(20)
- Chức năng Sign in:
 - Nhập thông tin Username, Password.
 - Chọn Sign in => Kiểm tra thông tin Username, Password trong bảng USER. Nếu có Username và Password xuất hiện thông báo “Đăng nhập thành công”; nếu không xuất hiện thông báo “Đăng nhập không thành công. Vui lòng nhập lại”



- Chức năng Sign up
 - Nhập thông tin Username, Password và thông tin Confirm để đăng ký. Password và Confirm phải giống nhau; nếu không giống sẽ xuất hiện thông báo.
 - Chọn Sign up để lưu xuống CSDL.
 - Nếu lưu thành công xuất hiện thông báo xác nhận “Bạn đã đăng ký tài khoản thành công”.

Câu 2: Xây dựng ứng dụng quản lý phòng khám theo yêu cầu sau:

- Thiết kế CSDL QLKB với các bảng sau:
 - **BACSI**(**MABS** char(4), TENBS varchar(20))
 - **BENHNHAN**(**MABN** char(4), TENBN varchar(20), NGSINH date, DCHI varchar(50), DTHOAI varchar(10), GIOITINH boolean)
 - **KHAMBENH**(**MAKB** char(6), MABN char(4), MABS char(4), NGÀYKHAM date, YEUCAUKHAM varchar(50), KETLUAN varchar(100), THANHTOAN boolean)
 - **DICHVU**(**MADV** char(6), TENDV, DONGIA).
 - **THUPHI**(**MAKB** char(6), **MADV** char(6), SOLUONG int, THANHTIEN bigint)
- Thêm thông tin bệnh nhân
 - Điền thông tin vào trong giao diện thêm bệnh nhân
 - Nhấn “Thêm” để thêm thông tin bệnh nhân vào bảng BENHNHAN

- Nếu thêm thành công xuất hiện thông báo xác nhận “Thêm bệnh nhân thành công”; ngược lại “Thêm bệnh nhân không thành công”.
- Đặt lịch khám của bệnh nhân
 - Khi nhập mã bệnh nhân và nhấn phím enter thì sẽ hiển thị tên bệnh nhân vào textfield “Tên bệnh nhân” (không chỉnh sửa được). Nếu không tìm thấy mã bệnh nhân sẽ xuất hiện thông báo “Bệnh nhân chưa đăng ký”.
 - Tên bác sĩ khám sẽ được load từ CSDL vào combobox “Tên bác sĩ”.
 - Chọn ngày khám bệnh (ngày khám phải lớn hơn hoặc bằng ngày hiện tại).
 - Nhập yêu cầu khám.

- Chọn bác sĩ khám trong combobox.
- Nhấn “Đặt lịch khám” toàn bộ thông tin trên sẽ lưu xuống bảng KHAMBENH.
- Nếu đặt lịch thành công xuất hiện thông báo xác nhận “Đặt lịch khám thành công”; ngược lại “Đặt lịch khám không thành công”.

Đặt lịch khám

Mã bệnh nhân:

Tên bệnh nhân:

Ngày khám:

Yêu cầu khám:

Bác sĩ khám:

Đặt lịch khám

○ Thêm chi tiết khám bệnh

Khám bệnh

Bác sĩ khám:

Ngày khám:

Tên bệnh nhân:

Yêu cầu khám:

Kết luận:

Danh sách dịch vụ

Tên dịch vụ
Dịch vụ a
Dịch vụ b
Dịch vụ c
Dịch vụ d

Danh sách dịch vụ bác sĩ chọn

Tên dịch vụ	Số lượng
Dịch vụ a	2
Dịch vụ b	1

Thêm

- Tên bác sĩ được load từ CSDL vào combobox.
- Danh sách dịch vụ được load từ CSDL vào JTable “Danh sách dịch vụ”.
- Chọn bác sĩ trong combobox, nhập ngày tháng và nhấn enter => danh sách bệnh nhân đặt lịch hẹn với bác sĩ sẽ hiển thị ở combobox “Tên bệnh nhân”. Lưu ý: chỉ load

bệnh nhân chưa thanh toán cho ngày khám đã được chọn.

- Chọn tên bệnh nhân sẽ hiển thị yêu cầu khám ở textfield “Yêu cầu khám” (không được chỉnh sửa yêu cầu khám).
- Nhập kết luận ở textfield “Kết luận”.
- Chọn các dịch vụ, mỗi lần chọn dịch vụ bên “Danh sách dịch vụ” thì dịch vụ đó sẽ được hiển thị bên JTable “Danh sách dịch vụ bác sĩ chọn” và bên “Danh sách dịch vụ” sẽ không còn hiển thị dịch vụ đó nữa. Thêm số lượng cho từng dịch vụ.
- Nhấn “Thêm” để thêm toàn bộ thông tin vào các bảng tương ứng trong CSDL.

○ Thanh toán tiền khám bệnh

Thanh toán khám bệnh

Mã bệnh nhân: 1234 Ngày khám: May 17, 2020

Tên bệnh nhân: ABC Yêu cầu khám: Khám tai mũi họng

Kết luận: Viêm họng Tổng tiền: 80000

Danh sách dịch vụ đã khám ☐ Đã thanh toán

Tên dịch vụ	Số lượng	Thành tiền
Dịch vụ a	2	30000
Dịch vụ b	1	50000

Thanh toán

- Khi nhập mã bệnh nhân và nhấn phím enter thì sẽ hiển thị tên bệnh nhân vào textfield “Tên bệnh nhân” (không chỉnh sửa được).
- Chọn ngày khám => hiển thị thông tin khám bệnh của bệnh nhân trong ngày đã chọn (tất cả đều không chỉnh sửa được) bao gồm: Yêu cầu khám, Kết luận, Danh sách dịch vụ đã khám (Tên dịch vụ, số lượng, thành tiền), Tổng tiền.
- Checkbox “Đã thanh toán” nếu được chọn thì nút “Thanh toán” sẽ bị mờ (Không hoạt động được); ngược lại, thì nút “Thanh toán” sẽ hiển thị, và khi chọn “Thanh toán” trạng thái “Thanh toán” trong CSDL sẽ được chuyển thành “True” để xác định bệnh nhân đã thanh toán cho đơn khám bệnh của mình.

Câu 3: (Optional) Cài đặt câu 2 phần hướng dẫn chung và câu 2 phần thực hành theo mô hình 3 lớp trong Java