

Lab 06: LẬP TRÌNH CƠ SỞ DỮ LIỆU TRONG JAVA SWING

I. Tổng quan bài thực hành

- Biết cấu hình SQL Server cho phép kết nối cơ sở dữ liệu trong Java sử dụng JDBC.
- Sử dụng thành thạo các control: JTextField, JButton, JLabel, JComboBox, JRadioButton, JTextbox, JTable.

II. Bài tập thực hành

Bài 1: Xây dựng ứng dụng Quản lý sinh viên đơn giản

Title 1	Title 2	Title 3	Title 4

Yêu cầu:

- Tạo ra cơ sở dữ liệu với bảng dữ liệu như sau:

SinhVien			
	Column Name	Data Type	Allow Nulls
🔑	MaSV	char(7)	<input type="checkbox"/>
	HoTen	nvarchar(50)	<input type="checkbox"/>
	Email	nvarchar(50)	<input type="checkbox"/>
	SoDT	char(20)	<input checked="" type="checkbox"/>
	GioiTinh	smallint	<input type="checkbox"/>
	DiaChi	nvarchar(150)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- Tạo lớp Student.java để lưu thông tin 1 sinh viên, chứa các thuộc tính tương ứng với các trường trong CSDL.

```
public class Student {  
    private String maSV, hoTen, email, soDT, diaChi;  
    private int gioiTinh;  
  
    public Student() {  
    }  
  
    public Student(String maSV, String hoTen, String email, String soDT, String diaChi, int gioiTinh) {  
        this.maSV = maSV;  
        this.hoTen = hoTen;  
        this.email = email;  
        this.soDT = soDT;  
        this.diaChi = diaChi;  
        this.gioiTinh = gioiTinh;  
    }  
}
```

- Tạo lớp DatabaseHeplper cho phép chương trình kết nối đến SQL Server

```

public class DatabaseHelper {
    public static Connection openConnection() throws Exception{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        String connectionUrl = "jdbc:sqlserver://localhost;database=QLSinhVien;";
        String username = "demo";
        String password = "demo";
        Connection con = DriverManager.getConnection(connectionUrl, username, password);

        return con;
    }
}

```

- Tạo lớp **StudentDAO** cho phép tìm kiếm, thêm, xóa, cập nhật sinh viên: tạo các phương thức thêm, xóa và cập nhật cho đối tượng sinh viên. Tham khảo hàm **insert** như hình dưới.

```

public class StudentDao {
    public boolean insert(Student stud) throws Exception{
        String sql = "insert into student(masv, hoten, email, sodt, gioitinh, diachi) values(?, ?, ?, ?, ?, ?)";
        try{
            Connection con = DatabaseHelper.openConnection();
            PreparedStatement pstmt = con.prepareStatement(sql);
        }{
            pstmt.setString(1, stud.getMaSV());
            pstmt.setString(2, stud.getHoTen());
            pstmt.setString(3, stud.getEmail());
            pstmt.setString(4, stud.getSoDT());
            pstmt.setInt(5, stud.getGioiTinh());
            pstmt.setString(6, stud.getDiaChi());

            return pstmt.executeUpdate() > 0;
        }
    }
}


```

- Thiết kế giao diện như hình trên.
 - Tên các control phải đặt tên đúng quy định
 - Khi chạy chương trình lần đầu tiên thì luôn hiện cửa sổ chương trình ở giữa màn hình desktop.
- Nút **Nhập mới**: Reset tất cả các điều khiển trên màn hình chương trình về mặc định (với Text để rỗng, với các nút thì unSelected)
- Nút **Lưu**: Thêm mới 1 sinh viên vào bảng Student
 - Kiểm tra nếu **Mã sinh viên** trống thì **tô đỏ ô Mã sinh viên** và thông báo: “*Mã sinh viên không được để trống*”.
 - Khi thêm thành công hiện thị thoại thông báo cho người dùng biết sinh viên mới đã được lưu.

- Lưu dữ liệu xuống database đồng thời hiện thị lên JTable.
- Nút **Thoát**: thoát khỏi chương trình
- Nút **Tìm kiếm**: Cho phép tìm kiếm 1 sinh viên theo mã sinh viên
 - Kiểm tra nếu **Mã sinh viên** trống thì **tô đỏ ô Mã sinh viên** và thông báo: “*Mã sinh viên không được để trống*”.
 - Nếu tìm thấy thì hiện thị dữ liệu lên JTable, ngược lại hiện hội thoại thông báo không tìm thấy sinh viên.
- Nút **Cập nhật**
 - Kiểm tra nếu **Mã sinh viên** trống thì **tô đỏ ô Mã sinh viên** và thông báo: “*Mã sinh viên không được để trống*”.
 - Lấy thông tin đã sửa từ các Textbox để cập nhật xuống cơ sở dữ liệu
 - Hiện thị thông báo xác nhận việc cập nhật.
 - Cập nhật thành công thì hiện thị thông báo cho người dùng.
 - Cập nhật dữ liệu chỉnh sửa lên JTable.
- Nút **Xóa**
 - Kiểm tra nếu **Mã sinh viên** trống thì **tô đỏ ô Mã sinh viên** và thông báo: “*Mã sinh viên không được để trống*”.
 - Hiện thị thông báo xác nhận việc xóa sinh viên.
 - Xóa thành công thì hiện thị thông báo cho người dùng.
 - Cập dữ liệu mới lên JTable.

Bài 2: Xây dựng chức năng tìm kiếm và xóa trong ứng dụng quản lý sách

- Tạo ra cơ sở dữ liệu với bảng dữ liệu như sau:

	Column Name	Data Type	Allow Nulls
	BookId	int	<input type="checkbox"/>
	Title	nvarchar(50)	<input type="checkbox"/>
	Price	float	<input type="checkbox"/>
			<input type="checkbox"/>

- Thêm vào bảng 5 dòng dữ liệu.

- Thiết kế giao diện như hình.

- Tên các control phải đặt tên đúng quy định
- Khi chạy chương trình lần đầu tiên thì luôn hiện cửa sổ chương trình ở giữa màn hình desktop.
- Nút **Exit**: thoát khỏi chương trình
- Nút **Search**: nhập từ khóa tìm kiếm thì sẽ hiện thị kết quả ở bảng bên dưới. Sử dụng tìm kiếm tương đối (nghĩa là nhập vào 1 từ khóa thì chương trình trả về kết quả có xuất hiện từ khóa). Tham khảo mẫu code sau:

```
String searchCriteria = "%";
if (!txtSearchTitle.getText().equals("")) {
    searchCriteria += txtSearchTitle.getText() + "%";
}
```

- Nút **Delete**: Chọn dòng dữ liệu trong bảng và cho phép người dùng xóa. Hiện hộp thoại yêu cầu xác nhận trước khi xóa. Sau khi xóa thành công thì có hiện thị ra thông báo.
- Cần tạo ra lớp **BookDAO** để chứa các phương thức Tìm kiếm, xóa dữ liệu

III. Tổ chức code

- Tạo dự án với Package **Lab06_MSSV** để chứa các 2 Package con, tương ứng với 2 bài tập.

IV. Hướng dẫn nộp bài

- Nén toàn bộ thư mục và các tài nguyên cần thiết của dự án để giảng viên chấm điểm bằng file .zip
- Chọn File -> Export -> Project to Zip file.
- Nộp bài trên hệ thống LMS

