MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng:

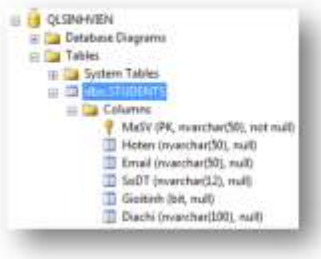
- ✓ Kết nối database, thực hiện các chức năng thêm, xóa, sửa, update, tiềm kiếm
- ✓ Hiểu được và ứng dụng các lớp tác vụ cơ bản của JDBC:
  - Statement
  - PreparedStatement
  - CallableStatement
  - ResultSet

NỘI DUNG:

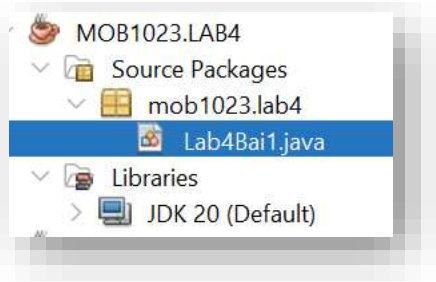
BÀI 1: KẾT NỐI CSDL SQL SERVER (2 ĐIỂM)

Yêu cầu: Viết chương trình đọc dữ liệu từ table hiển thị lên màn hình:

Database: tạo database và table có cấu trúc như hình

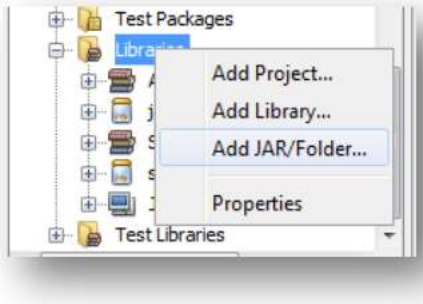


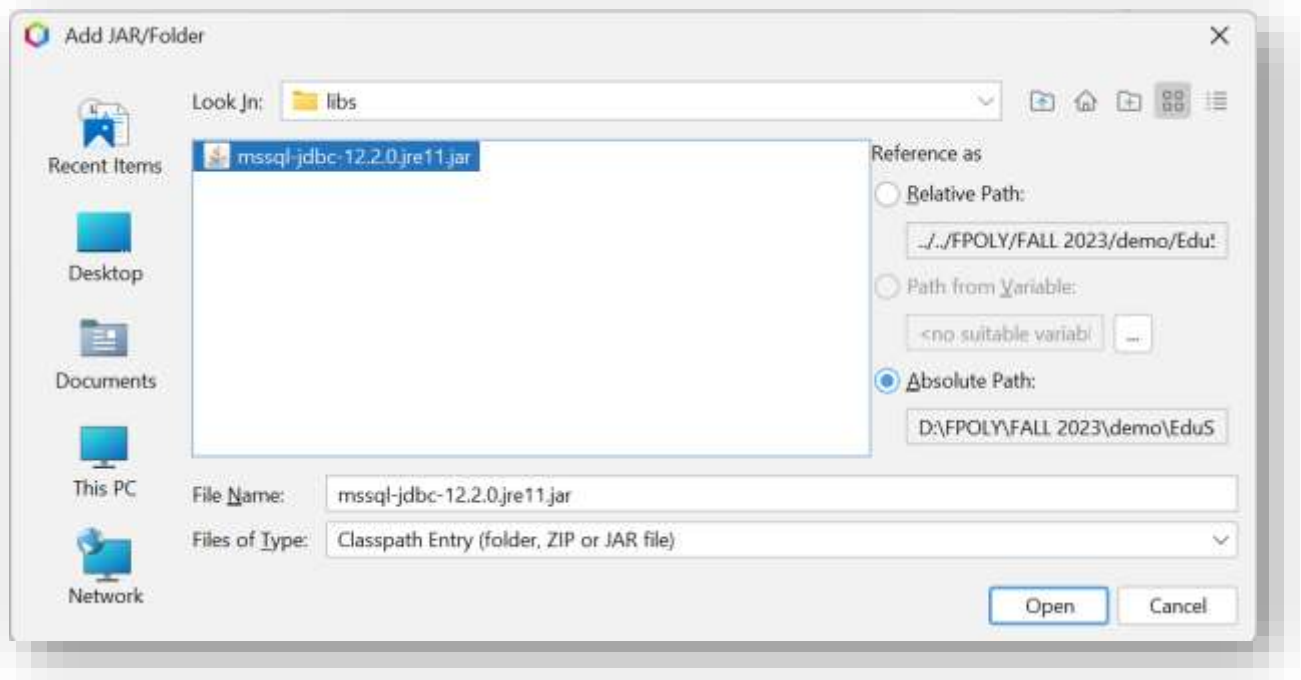
Tạo project Java:



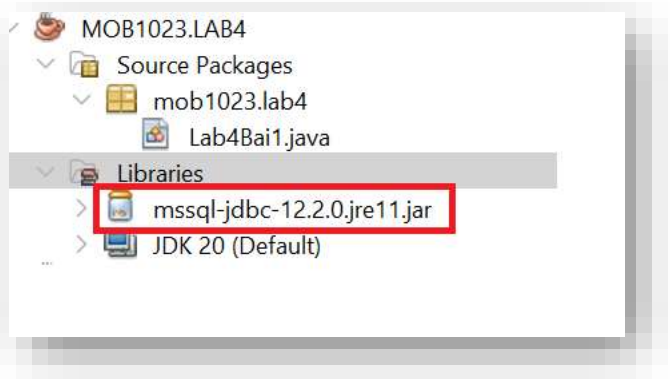
Add Library: click phải chọn Library\Add Jar/Folder…, chọn file

mssql-jdbc-12.2.0.jre11.jar





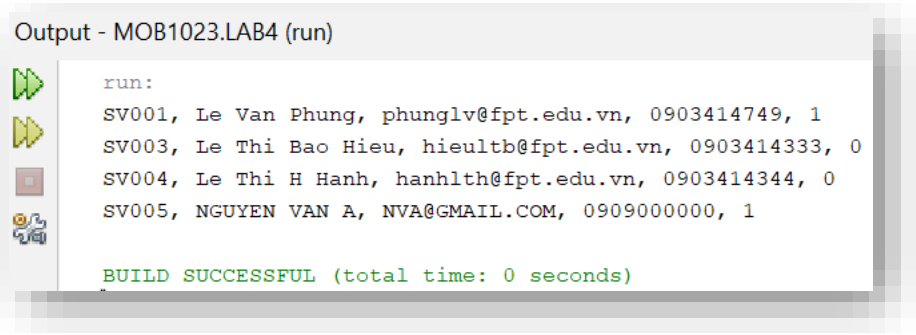
Kết quả như hình



Code:

```
/**
 *
 * @author nttai
 */
public class Lab4Bai1 {
    public static void main(String[] args) {
        try {
            String user = "sa";
            String pass = "123";
            Class.forName(className: "com.microsoft.sqlserver.jdbc.SQLServerDriver");
            String url = "jdbc:sqlserver://localhost:1433;database=QLSINHVIEN;encrypt=false";
            Connection con = DriverManager.getConnection(url, user, password: pass);
            String sql = "select * from students";
            Statement stm = con.createStatement();
            ResultSet rs = stm.executeQuery(string: sql);
            while (rs.next()) {
                System.out.print(rs.getString(string: "MaSV") + ", ");
                System.out.print(rs.getString(string: "Hoten") + ", ");
                System.out.print(rs.getString(string: "Email") + ", ");
                System.out.print(rs.getString(string: "SoDT") + ", ");
                System.out.println(x: rs.getString(string: "Gioitinh"));
            }
            con.close();
        } catch (Exception e) {
            System.out.println(x: e);
        }
    }
}
```

Kết quả:

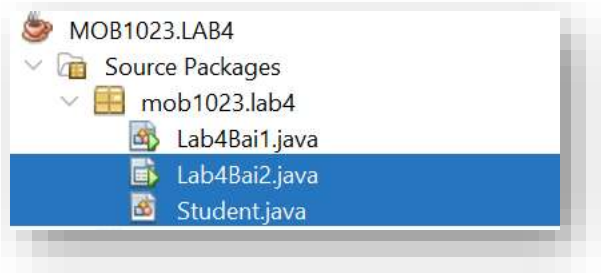


BÀI 2: THIẾT KẾ ỨNG DỤNG NHƯ HÌNH VÀ XỬ LÝ CHỨC NĂNG THEO YÊU CẦU (7 ĐIỂM)

Cho phép người dùng thêm, xoá, sửa, update dữ liệu của bảng Students. Ngoài ra chương trình còn cho phép duyệt danh sách sinh viên

Database: như bài 1

Cấu trúc chương trình gồm:



Tạo file Student.java có nội dung như sau:

```java
/**
 *
 * @author nttai
 */
public class Student {
    String maSV;
    String hoten;
    String email;
    String soDT;
    String diachi;
    boolean gioiTinh;

    public Student() {
    }

    public Student(String maSV, String hoten, String email, String soDT, String diachi, boolean gioiTinh) {
        this.maSV = maSV;
        this.hoten = hoten;
        this.email = email;
        this.soDT = soDT;
        this.diachi = diachi;
        this.gioiTinh = gioiTinh;
    }

    public String getMaSV() {
        return maSV;
    }

    public void setMaSV(String maSV) {
        this.maSV = maSV;
    }

    public String getHoten() {
        return hoten;
    }
    public void setHoten(String hoten) {
        this.hoten = hoten;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getSoDT() {
        return soDT;
    }

    public void setSoDT(String soDT) {
        this.soDT = soDT;
    }

    public String getDiachi() {
        return diachi;
    }

    public void setDiachi(String diachi) {
        this.diachi = diachi;
    }

    public boolean isGioiTinh() {
        return gioiTinh;
    }

    public void setGioiTinh(boolean gioiTinh) {
        this.gioiTinh = gioiTinh;
    }
}
```

Khai báo các biến toàn cục:

Hàm LoadDataToArray(): đọc tất cả dữ liệu trong table students vào list.

```
public void loadDataToArray(){
    try {
        Class.forName(className: "com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection conn = DriverManager.getConnection(url, user: userName, password);
        String sql = "select * from students";
        Statement stm = conn.createStatement();
        ResultSet rs = stm.executeQuery(string: sql);
        list.clear();
        while (rs.next()) {
            Student sv = new Student();
            sv.setMaSV(maSV: rs.getString(string: "MaSV"));
            sv.setHoten(hoten: rs.getString(string: "Hoten"));
            sv.setEmail(email: rs.getString(string: "Email"));
            sv.setSoDT(soDT: rs.getString(string: "SoDT"));
            sv.setGioiTinh(gioiTinh: rs.getBoolean(string: "Gioitinh"));
            sv.setDiachi(diachi: rs.getString(string: "Diachi"));
            list.add(e: sv);
        }

        conn.close();
    } catch (Exception e) {
        System.out.println(x: e);
    }
}
```

Hàm display(int i): dùng để hiển thị sv thứ i trong arraylist lên Frame

```
public void display(int i){
    Student sv = list.get(index: i);
    txtMaSV.setText(t: sv.getMaSV());
    txtHoten.setText(t: sv.getHoten());
    txtEmail.setText(t: sv.getEmail());
    txtSoDT.setText(t: sv.getSoDT());
    txtDiachi.setText(t: sv.getDiachi());
    boolean gt = sv.isGioiTinh();
    if(gt){
        rdoNam.setSelected(b: true);
    }else{
        rdoNu.setSelected(b: true);
    }
}
```

Viết xử lý cho các nút: First, Next, Previous, Last

```
private void btnFirstActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    current = 0;
    display(i: current);
}
```

```
private void btnPrevActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    if(current == 0){
        JOptionPane.showMessageDialog(parentComponent:null, message:"Đang ở đầu danh sách!");
        return;
    }
    current--;
    display(i: current);
}
```

```
private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    if(current == list.size() - 1){
        JOptionPane.showMessageDialog(parentComponent:null, message:"Đang ở cuối!");
        return;
    }
    current++;
    display(i: current);
}
```

```
private void btnLastActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    current = list.size() - 1;
    display(i: current);
}
```

Viết xử lý cho nút "Add":

```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtMaSV.setText(t: "");
    txtHoten.setText(t: "");
    txtEmail.setText(t: "");
    txtSoDT.setText(t: "");
    txtDiachi.setText(t: "");
    txtMaSV.requestFocus();
}
```

Viết xử lý cho nút "Delete":

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(txtMaSV.getText().equals(anObject: "")){
        JOptionPane.showMessageDialog(parentComponent: this, message: "Nhập Mã SV");
        txtMaSV.requestFocus();
        return;
    }
    try {
        Class.forName(className: "com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection conn = DriverManager.getConnection(url, user: userName, password);
        String sql = "delete from students where MaSV = ?";
        PreparedStatement st = conn.prepareStatement(string: sql);
        st.setString(i: 1, string: txtMaSV.getText());
        st.execute();
        JOptionPane.showMessageDialog(parentComponent: this, message: "Delete thành công!");
        conn.close();
        display(current--);
    } catch (Exception e) {
        System.out.println(x: e);
    }
}
```

Viết xử lý cho nút "Update":

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection conn = DriverManager.getConnection(url, userName, password);
        String sql = "update students set Hoten = ?, Email = ?, SoDT = ?, Gioitinh = ?, Diachi = ? where MaSV = ?";
        PreparedStatement st = conn.prepareStatement(sql);

        st.setString(1, txtHoten.getText());
        st.setString(2, txtEmail.getText());
        st.setString(3, txtSoDT.getText());
        st.setBoolean(4, rdoNam.isSelected());
        st.setString(5, txtDiachi.getText());
        st.setString(6, txtMaSV.getText());
        st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Update thành công!");
        conn.close();
        loadDataToArray();
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

Viết xử lý cho nút "Save":

```
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection conn = DriverManager.getConnection(url, userName, password);
        String sql = "insert into students values(?, ?, ?, ?, ?, ?)";
        PreparedStatement st = conn.prepareStatement(sql);
        st.setString(1, txtMaSV.getText());
        st.setString(2, txtHoten.getText());
        st.setString(3, txtEmail.getText());
        st.setString(4, txtSoDT.getText());
        st.setBoolean(5, rdoNam.isSelected());
        st.setString(6, txtDiachi.getText());
        st.executeUpdate();
        JOptionPane.showMessageDialog(this, "Save thành công!");
        conn.close();
        loadDataToArray();
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

BÀI 3: GIẢNG VIÊN CHO THÊM (1 ĐIỂM)

***** YÊU CẦU NỘP BÀI:**

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết ---