

```

package connectDB;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

public class Database {

    public static Connection con = null;
    private static Database instance = new Database();
    public static Database getInstance() {
        return instance;
    }
    public void connect() {
        String url = "jdbc:sqlserver://localhost:1433;databasename=QLLop";
        String user = "sa";
        String password = "123";
        try {
            con = DriverManager.getConnection(url, user, password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    public static Connection getConnection() {
        return con;
    }
    public void disconnect() {
        if(con != null)
            try {
                con.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
    }
}
package entity;

```

```
public class GiaoVien {
    private String magv;
    private String tengv;

    public GiaoVien(String magv) {
        super();
        this.magv = magv;
    }

    public GiaoVien(String magv, String tengv) {
        super();
        this.magv = magv;
        this.tengv = tengv;
    }

    public String getMagv() {
        return magv;
    }

    public void setMagv(String magv) {
        this.magv = magv;
    }

    public String getTengv() {
        return tengv;
    }

    public void setTengv(String tengv) {
        this.tengv = tengv;
    }

    @Override
    public String toString() {
        return magv ;
    }

}
```

```
package entity;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JOptionPane;

public class LopHoc {
    private String maLop;
    private String tenLop;
    private GiaoVien giaoVien;
    private int siso;

    public LopHoc(String maLop, String tenLop, GiaoVien giaoVien, int siso) {
        this.maLop = maLop;
        this.tenLop = tenLop;
        this.giaoVien = giaoVien;
        this.siso = siso;
    }

    public LopHoc(String maLop) {
        super();
        this.maLop = maLop;
    }

    public LopHoc(GiaoVien giaoVien) {
        super();
        this.giaoVien = giaoVien;
    }

    public int getSiso() {
        return siso;
    }
}
```

```

    }
    public void setSiso(int siso) {
        this.siso = siso;
    }
    public String getMaLop() {
        return maLop;
    }

    public void setMaLop(String maLop) {
        this.maLop = maLop;
    }

    public String getTenLop() {
        return tenLop;
    }

    public void setTenLop(String tenLop) {
        this.tenLop = tenLop;
    }

    public GiaoVien getGiaoVien() {
        return giaoVien;
    }

    public void setGiaoVien(GiaoVien giaoVien) {
        this.giaoVien = giaoVien;
    }
}
@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((maLop == null) ? 0 : maLop.hashCode());
    return result;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;

```

```

        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        LopHoc other = (LopHoc) obj;
        if (maLop == null) {
            if (other.maLop != null)
                return false;
        } else if (!maLop.equals(other.maLop))
            return false;
        return true;
    }
}

package dao;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import connectDB.Database;
import entity.GiaoVien;

public class GiaoVien_DAO {
    List<entity.GiaoVien> dsgv;
    public GiaoVien_DAO(){
        dsgv= new ArrayList<entity.GiaoVien>();
    }
    public List<entity.GiaoVien> docTuBang() {

        try {
            Connection con = Database.getInstance().getConnection();

```

```

        String sql = "Select * from giaovien";
        Statement statement = con.createStatement();
        // Thực thi câu lệnh SQL trả về đối tượng ResultSet.
        ResultSet rs = statement.executeQuery(sql);
        // Duyệt trên kết quả trả về.
        while (rs.next()) { // Di chuyển con trỏ xuống bản ghi kế tiếp.
            String ma = rs.getString("maGiaoVien");
            String ten= rs.getString(2);

            GiaoVien s = new GiaoVien(ma,ten);

            dsgv.add(s);
        }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        // Đóng kết nối
    }
    return dsgv;
}

}
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JOptionPane;

import connectDB.Database;
import entity.GiaoVien;
import entity.LopHoc;

public class LopHoc_DAO {
    ArrayList<LopHoc> dslop;

```

```

LopHoc lh;
public LopHoc_DAO(){
    dslop= new ArrayList<LopHoc>();
}
public ArrayList<LopHoc> docTuBang() {

    try {
        Connection con = Database.getInstance().getConnection();
        String sql = "Select * from lophoc";
        Statement statement = con.createStatement();
        // Thực thi câu lệnh SQL trả về đối tượng ResultSet.
        ResultSet rs = statement.executeQuery(sql);
        // Duyệt trên kết quả trả về.
        while (rs.next()) { // Di chuyển con trỏ xuống bản ghi kế tiếp.
            String ma = rs.getString(1);
            String ten= rs.getString(2);
            String gv = rs.getString(3);
            int siso = rs.getInt(4);
            LopHoc s = new LopHoc(ma,ten,new GiaoVien(gv),siso);
            dslop.add(s);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    // Đóng kết nối
}
return dslop;
}

public ArrayList<LopHoc> getLopTheoMaLop(String mlop) {

    Connection con = Database.getInstance().getConnection();
    PreparedStatement stmt =null;
    try {
        String sql = "Select * from lophoc where maLop = ?";
        //String sql = "Select * from lophoc";
        stmt=con.prepareStatement(sql);
        stmt.setString(1, mlop);
        // Thả con trỏ về đầu SQL trả về đối tượng ResultSet.

```

```

ResultSet rs = stmt.executeQuery();
// Duyá»tt trÃn ká»t quá»f trá»f vá»».
while (rs.next()) { // Di chúyá»fn con trá»» xuá»'ng bá»fn ghi ká»t tiá»p.
    String ma = rs.getString(1);
    System.out.println(ma);
    String ten= rs.getString(2);
    String gv = rs.getString(3);
    int siso = rs.getInt(4);
    LopHoc s = new LopHoc(ma,ten,new GiaoVien(gv),siso);
    dslop.add(s);
}
} catch (SQLException e) {
    e.printStackTrace();
}

return dslop;
}

public boolean create(LopHoc p) {
    Connection con = Database.getInstance().getConnection();
    PreparedStatement stmt = null;
    int n = 0;
    try {
        stmt = con.prepareStatement("insert into LopHoc values(?, ?, ?, ?)");
        stmt.setString(1,p.getMaLop());
        stmt.setString(2,p.getTenLop());
        stmt.setString(3,p.getGiaoVien().getMagv());
        stmt.setInt(4,p.getSiso());
        n = stmt.executeUpdate();

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return n > 0;
}

//U: Update
public boolean update(LopHoc p) {
    Connection con = Database.getInstance().getConnection();
    PreparedStatement stmt = null;

```

```

        int n = 0;
        try {
            stmt = con.prepareStatement("update LopHoc set tenLop = ?, maGiaoVien = ?, siSo = ? where maLop = ?");
            stmt.setString(1, p.getTenLop());
            stmt.setString(2, p.getGiaoVien().getMagv());
            stmt.setInt(3, p.getSiso());
            stmt.setString(4, p.getMaLop());

            n = stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return n > 0;
    }

    //D: Delete
    public boolean delete(String mlop) {
        Connection con = Database.getInstance().getConnection();
        PreparedStatement stmt = null;
        int n = 0;
        try {
            stmt = con.prepareStatement("delete from LopHoc where maLop = ?");
            stmt.setString(1, mlop);
            n = stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return n > 0;
    }
}

package gui;

import java.awt.Color;
import java.awt.Font;
import java.util.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;

```

```

import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

import connectDB.Database;
import dao.GiaoVien_DAO;
import dao.LopHoc_DAO;
import entity.GiaoVien;
import entity.LopHoc;

public class FrmLopHoc extends JFrame implements ActionListener,MouseListener{

    private JTextField txtMaLop ;
    private JTextField txtTenLop;
    private JTextField txtSiSo;
    private JComboBox cboGVCN;
    private JButton btnThem;
    private JButton btnLuu;
    private JButton btnSua;
    private JButton btnXoa;
    private JButton btnTimGV,btnTimLop;
    private DefaultTableModel dataModel;
    private JTable table;
    LopHoc_DAO dslh = new LopHoc_DAO();
    public FrmLopHoc() {

```

```

setTitle("Thông tin lớp học");
setSize(1000, 600);
setLocationRelativeTo(null);
setDefaultCloseOperation(EXIT_ON_CLOSE);

Box b, b1, b2, b3, b4, b44, b5, b6, b7;
//Dùng Box layout
add(b = Box.createVerticalBox()); //Box theo chiều?u d?c
b.add(Box.createVerticalStrut(10)); //Tạo khoảng cách theo chiều?u d?c
b.add(b1 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10)); //b1 -> b7: Box theo chiều?u ngang
b.add(b2 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10));
b.add(b3 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10));
b.add(b4 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10));
b.add(b44 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10));
b.add(b5 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10));
b.add(b6 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10));
b.add(b7 = Box.createHorizontalBox()); b.add(Box.createVerticalStrut(10));

JLabel lblTieuDe, lblMaLop, lblTenLop, lblGVCN, lblSiSo;
b1.add(lblTieuDe = new JLabel("THÔNG TIN LỚP HỌC", JLabel.CENTER));
lblTieuDe.setFont(new Font("Arial", Font.BOLD, 26));

b2.add(lblMaLop = new JLabel("Mã lớp: ", JLabel.RIGHT)); b2.add(txtMaLop = new JTextField());
b3.add(lblTenLop = new JLabel("Tên lớp: ", JLabel.RIGHT)); b3.add(txtTenLop = new JTextField());
b4.add(lblGVCN = new JLabel("Giáo viên chủ nhiệm: ", JLabel.RIGHT));
b4.add(cboGVCN = new JComboBox<String>());
cboGVCN.setEditable(false);
b44.add(lblSiSo = new JLabel("Số : ", JLabel.RIGHT)); b44.add(txtSiSo = new JTextField());

lblMaLop.setPreferredSize(lblGVCN.getPreferredSize());
lblTenLop.setPreferredSize(lblGVCN.getPreferredSize());
lblSiSo.setPreferredSize(lblGVCN.getPreferredSize());

b5.add(btnThem = new JButton("Thêm"));
b5.add(btnLuu= new JButton("Lưu"));
b5.add(btnSua = new JButton("Sửa"));
b5.add(btnXoa = new JButton("Xóa"));
b5.add(btnTimGV = new JButton("Tìm theo mã giáo viên"));
b5.add(btnTimLop = new JButton("Tìm theo mã lớp"));

```

```
String[] headers = {"Mã lớp", "Tên lớp", "Giáo viên CN", "Số số"};
dataModel = new DefaultTableModel(headers , 0);
JScrollPane scroll;
b6.add(scroll = new JScrollPane(table = new JTable(dataModel)));
scroll.setBorder(BorderFactory.createTitledBorder("Danh sách lớp học"));

b7.add(Box.createHorizontalStrut(600));
```

```
btnThem.addActionListener(this);
btnLuu.addActionListener(this);
btnXoa.addActionListener(this);
btnSua.addActionListener(this);
btnTimGV.addActionListener(this);
btnTimLop.addActionListener(this);
table.addMouseListener(this);
```

```
//Khi chương trình chạy, nạp toàn bộ danh sách lớp học lên bảng
Database.getInstance().connect();
updateComboBox();
updateTableData();
```

```
}
@Override
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    Object o = e.getSource();
    if(o.equals(btnThem)){
        xoaRongTextfields();
    }
    else if(o.equals(btnLuu)){
        LopHoc lh = reverSPFFromTextFile();
        if(dslh.create(lh)){
            Object[] rowData = {txtMaLop.getText(), txtTenLop.getText(),
                cboGVCN.getSelectedItem().toString(), txtSiSo.getText()};
            dataModel.addRow(rowData);
        }
    }
}
```

```

}
else if(o.equals(btnSua)){
    int row = table.getSelectedRow();
    if(row >= 0){
        LopHoc lh = reverSPFromTextFile();
        if(dslh.update(lh)){
            table.setValueAt(txtTenLop.getText(), row, 1);
            table.setValueAt(cboGVCN.getSelectedItem().toString(), row, 2);
            table.setValueAt(txtSiSo.getText(), row, 3);
        }
    }
}
else if(o.equals(btnXoa)){
    int row = table.getSelectedRow();
    if(row >= 0){
        String maLop = (String) table.getValueAt(row, 0);
        if(dslh.delete(maLop)){
            dataModel.removeRow(row);
            xoaRongTextfields();
        }
    }
}
else if(o.equals(btnTimLop)){
    LopHoc_DAO ds = new LopHoc_DAO();
    List<LopHoc> list = ds.getLopTheoMaLop(txtMaLop.getText());

    if (txtMaLop.getText().equals(""))
        JOptionPane.showMessageDialog(this, "Vui long nhap ma lop can tim!");
    else if(list.size()==0)
        JOptionPane.showMessageDialog(this, "Không tìm thấy!!!");
    else
    {
        dataModel.getDataVector().removeAllElements();
        for(LopHoc s : list){
            String [] rowData = {s.getMaLop(), s.getTenLop(), s.getGiaoVien() + "", s.getSiso() + ""};
            dataModel.addRow(rowData);
        }
        table.setModel(dataModel);
    }
}
}

```

```

}
}
private void xoaRongTextfields() {
    txtMaLop.setText("");
    txtTenLop.setText("");
    cboGVCN.setSelectedItem(null);
    txtSiSo.setText("");
    txtMaLop.requestFocus();
}
//đưa dữ liệu vào Combobox
private void updateComboBox() {
    GiaoVien_DAO ds = new GiaoVien_DAO();
    List<GiaoVien> list = ds.docTuBang();

    for(GiaoVien s : list)
        cboGVCN.addItem(s.getMagv());
}
private LopHoc reverSPFromTextFile() {
    String malop = txtMaLop.getText().toString();
    String tenlop = txtTenLop.getText().toString();
    String gv = cboGVCN.getSelectedItem().toString();
    int siso = Integer.parseInt(txtSiSo.getText());
    //kiem tra du lieu
    return new LopHoc(malop, tenlop, new GiaoVien(gv),siso);
}
//đưa dữ liệu vào table
private void updateTableData() {
    LopHoc_DAO ds = new LopHoc_DAO();
    List<LopHoc> list = ds.docTuBang();
    for(LopHoc s : list){
        String [] rowData = {s.getMaLop(), s.getTenLop(), s.getGiaoVien() + "", s.getSiso() + ""};
        dataModel.addRow(rowData);
    }
    table.setModel(dataModel);
}
public void mouseClicked(MouseEvent e){
    int row = table.getSelectedRow(); // lay dong dangchon tren table
    txtMaLop.setText(table.getValueAt(row,0).toString());
}
}
}

```

```

        txtTenLop.setText(table.getValueAt(row, 1).toString());
        cboGVCN.setSelectedItem(table.getValueAt(row, 2).toString());
        txtSiSo.setText(table.getValueAt(row, 3).toString());
    }
    @Override
    public void mouseEntered(MouseEvent e) {
        // TODO Auto-generated method stub
    }
    @Override
    public void mouseExited(MouseEvent e) {
        // TODO Auto-generated method stub
    }
    @Override
    public void mousePressed(MouseEvent e) {
        // TODO Auto-generated method stub
    }
    @Override
    public void mouseReleased(MouseEvent e) {
        // TODO Auto-generated method stub
    }
}

}
package gui;

public class TestLopHoc {
    public static void main(String[] args) {
        FrmLopHoc frm = new FrmLopHoc();
        frm.setVisible(true);
    }
}

```