**Class**

**1.Class MonHoc**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package MonHoc;

/\*\*

\*

\* @author Anonymous

\*/

public class MonHoc {

private String mamh;

private int sotrinh;

private String tenmh;

private String hinhthucthi;

private int hocky;

private String phonghoc;

public MonHoc() {

}

public MonHoc(String mamh, int sotrinh, String tenmh, String hinhthucthi, int hocky, String phonghoc) {

this.mamh = mamh;

this.sotrinh = sotrinh;

this.tenmh = tenmh;

this.hinhthucthi = hinhthucthi;

this.hocky = hocky;

this.phonghoc = phonghoc;

}

public int getHocky() {

return hocky;

}

public void setHocky(int hocky) {

this.hocky = hocky;

}

public String getMamh() {

return mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public int getSotrinh() {

return sotrinh;

}

public void setSotrinh(int sotrinh) {

this.sotrinh = sotrinh;

}

public String getTenmh() {

return tenmh;

}

public void setTenmh(String tenmh) {

this.tenmh = tenmh;

}

public String getHinhthucthi() {

return hinhthucthi;

}

public void setHinhthucthi(String hinhthucthi) {

this.hinhthucthi = hinhthucthi;

}

public String getPhonghoc() {

return phonghoc;

}

public void setPhonghoc(String phonghoc) {

this.phonghoc = phonghoc;

}

}

**2.class SinhVien**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package SinhVien;

import java.util.Date;

/\*\*

\*

\* @author Anonymous

\*/

public class SinhVien {

private String masv;

private String tensv;

private String malop;

private String diachi;

private String hedaotao;

private Date ngaysinh;

private boolean gioitinh;

private String sodt;

public SinhVien() {

}

public SinhVien(String masv, String tensv, String malop, String diachi, String hedaotao, Date ngaysinh, boolean gioitinh, String sodt) {

this.masv = masv;

this.tensv = tensv;

this.malop = malop;

this.diachi = diachi;

this.hedaotao = hedaotao;

this.ngaysinh = ngaysinh;

this.gioitinh = gioitinh;

this.sodt = sodt;

}

public String getMasv() {

return masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getTensv() {

return tensv;

}

public void setTensv(String tensv) {

this.tensv = tensv;

}

public String getMalop() {

return malop;

}

public void setMalop(String malop) {

this.malop = malop;

}

public String getDiachi() {

return diachi;

}

public void setDiachi(String diachi) {

this.diachi = diachi;

}

public String getHedaotao() {

return hedaotao;

}

public void setHedaotao(String hedaotao) {

this.hedaotao = hedaotao;

}

public Date getNgaysinh() {

return ngaysinh;

}

public void setNgaysinh(Date ngaysinh) {

this.ngaysinh = ngaysinh;

}

public boolean isGioitinh() {

return gioitinh;

}

public void setGioitinh(boolean gioitinh) {

this.gioitinh = gioitinh;

}

public String getSodt() {

return sodt;

}

public void setSodt(String sodt) {

this.sodt = sodt;

}

}

**3.class BangDiem**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package BangDiem;

/\*\*

\*

\* @author Anonymous

\*/

public class BangDiem {

private String masv;

private String mamh;

private int lanthi;

private int heso;

private float diem;

private boolean trangthai;

public BangDiem() {

}

public BangDiem(String masv, String mamh, int lanthi, int heso, float diem, boolean trangthai) {

this.masv = masv;

this.mamh = mamh;

this.lanthi = lanthi;

this.heso = heso;

this.diem = diem;

this.trangthai = trangthai;

}

public String getMasv() {

return masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getMamh() {

return mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public int getLanthi() {

return lanthi;

}

public void setLanthi(int lanthi) {

this.lanthi = lanthi;

}

public int getHeso() {

return heso;

}

public void setHeso(int heso) {

this.heso = heso;

}

public float getDiem() {

return diem;

}

public void setDiem(float diem) {

this.diem = diem;

}

public boolean isTrangthai() {

return trangthai;

}

public void setTrangthai(boolean trangthai) {

this.trangthai = trangthai;

}

}

**ClassTimKiem**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package Search;

/\*\*

\*

\* @author Anonymous

\*/

public class TimKiem {

private String malop;

private String masv;

public TimKiem() {

}

public TimKiem(String malop, String masv) {

this.malop = malop;

this.masv = masv;

}

public void setMasv(String masv) {

this.masv = masv;

}

public String getMasv() {

return masv;

}

public TimKiem(String malop) {

this.malop = malop;

}

public String getMalop() {

return malop;

}

public void setMalop(String malop) {

this.malop = malop;

}

}

**DAO**

1. **BangDiemDAO**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package BangDiem;

import JDBC.DBConnect;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Anonymous

\*/

public class BangDiemDAO implements IBangDiemDAO {

@Override

public ArrayList<BangDiem> getAll() {

//throw new UnsupportedOperationException("Not supported yet.");

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from Diem");

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findByIDSinhVien(String masv) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Diem where MaSV = ?");

ps.setString(1, masv);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findByIDMonHoc(String mamh) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from Diem where MaMH = ?");

ps.setString(1, mamh);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString(2));

bd.setLanthi(rs.getInt(3));

bd.setHeso(rs.getInt(4));

bd.setDiem(rs.getFloat(5));

bd.setTrangthai(rs.getBoolean(6));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public BangDiem addNew(BangDiem bd) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into Diem (MaSV,MaMH,LanThi,HeSo,Diem,TrangThai) values (?,?,?,?,?,?)");

ps.setString(1, bd.getMasv());

ps.setString(2, bd.getMamh());

ps.setInt(3, bd.getLanthi());

ps.setInt(4, bd.getHeso());

ps.setFloat(5, bd.getDiem());

ps.setBoolean(6, bd.isTrangthai());

int row = ps.executeUpdate();

if (row < 1) {

bd = null;

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps);

}

}

return bd;

}

@Override

public BangDiem upDate(BangDiem bd) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update Diem set HeSo =?,Diem = ?,TrangThai=? where MaSV =? and MaMH =? and LanThi=?");

ps.setInt(1, bd.getHeso());

ps.setFloat(2, bd.getDiem());

ps.setBoolean(3, bd.isTrangthai());

ps.setString(4, bd.getMasv());

ps.setString(5, bd.getMamh());

ps.setInt(6, bd.getLanthi());

int row = ps.executeUpdate();

if (row < 1) {

bd = null;

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close();

}

}

return bd;

}

public void delBangDiem(String MaSV, String MaMH, int LanThi) throws SQLException, ClassNotFoundException {

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from Diem where MaSV =? and MaMH =? and LanThi =?");

ps.setString(1, MaSV);

ps.setString(2, MaMH);

ps.setInt(3, LanThi);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public boolean CheckID(String masv, String mamh, int lanthi) {

PreparedStatement psCheck = null;

ResultSet rs = null;

boolean result = true;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \* from Diem where MaSV =? and MaMH =? and LanThi =?");

psCheck.setString(1, masv);

psCheck.setString(2, mamh);

psCheck.setInt(3, lanthi);

rs = psCheck.executeQuery();

result = rs.next();

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(psCheck, rs);

}

}

return result;

}

@Override

public ArrayList<BangDiem> findMaMH(String masv) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select DISTINCT mamh from Diem where trangthai is not null and masv=?");

ps.setString(1, masv);

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

bd.setMamh(rs.getString("mamh"));

// bd.setLanthi(rs.getInt("fldLanThi"));

// bd.setHeso(rs.getInt("fldHeSo"));

// bd.setDiem(rs.getFloat("fldDiem"));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

public ArrayList<BangDiem> loaddiem(String masv, String maMon) {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select lanthi, heso, diem from Diem where mamh =? and masv=?");

ps.setString(1, maMon);

ps.setString(2, masv);

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

//bd.setMasv(rs.getString(1));

// bd.setMamh(rs.getString("fldmamh"));

bd.setLanthi(rs.getInt("LanThi"));

bd.setHeso(rs.getInt("HeSo"));

bd.setDiem(rs.getFloat("Diem"));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<BangDiem> findMasv() {

ArrayList<BangDiem> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select DISTINCT masv from Diem where trangthai is not null");

rs = ps.executeQuery();

list = new ArrayList<BangDiem>();

while (rs.next()) {

BangDiem bd = new BangDiem();

bd.setMasv(rs.getString(1));

list.add(bd);

}

} catch (SQLException ex) {

Logger.getLogger(BangDiemDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

public static void main(String[] args) {

System.out.println(new BangDiemDAO().findMaMH("B103104111").get(0).getMasv());

}

}

**2.MonHocDAO**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package MonHoc;

import JDBC.DBConnect;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Anonymous

\*/

public class MonHocDAO implements IMonHocDAO {

@Override

public ArrayList<MonHoc> getAll() {

ArrayList<MonHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from MonHoc");

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

MonHoc mh = new MonHoc();

mh.setMamh(rs.getString(1));

mh.setTenmh(rs.getString(2));

mh.setSotrinh(rs.getInt(3));

mh.setHinhthucthi(rs.getString(4));

mh.setHocky(rs.getInt(5));

mh.setPhonghoc(rs.getString(6));

list.add(mh);

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<MonHoc> findIDMaMH(String maMH) {

ArrayList<MonHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from MonHoc where MaMH =?");

ps.setString(1, maMH);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

MonHoc mh = new MonHoc();

mh.setMamh(rs.getString(1));

mh.setTenmh(rs.getString(2));

mh.setSotrinh(rs.getInt(3));

mh.setHinhthucthi(rs.getString(4));

mh.setHocky(rs.getInt(5));

mh.setPhonghoc(rs.getString(6));

list.add(mh);

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public MonHoc addNew(MonHoc mh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into MonHoc(MaMH, TenMH,SoTrinh, HinhThucThi,HocKy,PhongHoc) values (?,?,?,?,?,?)");

ps.setString(1, mh.getMamh());

ps.setString(2, mh.getTenmh());

ps.setInt(3, mh.getSotrinh());

ps.setString(4, mh.getHinhthucthi());

ps.setInt(5, mh.getHocky());

ps.setString(6, mh.getPhonghoc());

int row = ps.executeUpdate();

if (row < 1) {

mh = null;

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

mh = null;

} finally {

DBConnect.close(ps);

}

}

return mh;

}

@Override

public MonHoc updateByID(MonHoc mh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update MonHoc set TenMH=?,SoTrinh=?, HinhThucThi =?,HocKy =?,PhongHoc =? where MaMH =?");

ps.setString(1, mh.getTenmh());

ps.setInt(2, mh.getSotrinh());

ps.setString(3, mh.getHinhthucthi());

ps.setInt(4, mh.getHocky());

ps.setString(5, mh.getPhonghoc());

ps.setString(6, mh.getMamh());

int row = ps.executeUpdate();

if (row < 1) {

mh = null;

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

mh = null;

} finally {

DBConnect.close();

}

}

return mh;

}

public void delMonHoc(String MonHocID) throws SQLException, ClassNotFoundException {

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from MonHoc where MaMH= ?");

ps.setString(1, MonHocID);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public ArrayList<MonHoc> CheckID(String mamh) {

// throw new UnsupportedOperationException("Not supported yet.");

ArrayList<MonHoc> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \*from MonHoc where MaMH=?");

psCheck.setString(1, mamh);

rs = psCheck.executeQuery();

list = new ArrayList<MonHoc>();

while (rs.next()) {

MonHoc monHoc = new MonHoc();

monHoc.setMamh(rs.getString(1));

list.add(monHoc);

}

} catch (SQLException ex) {

Logger.getLogger(MonHocDAO.class.getName()).log(Level.SEVERE, null, ex);

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}

**3. SinhVienDAO**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package SinhVien;

import JDBC.DBConnect;

import com.sun.corba.se.impl.protocol.giopmsgheaders.Message;

import com.sun.org.apache.xerces.internal.impl.dv.xs.YearDV;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Anonymous

\*/

public class SinhVienDAO implements ISinhVienDAO {

@Override

public ArrayList<SinhVien> getAll() {

ArrayList<SinhVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \*from SinhVien");

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

SinhVien sv = new SinhVien();

sv.setMasv(rs.getString(1));

sv.setTensv(rs.getString(2));

sv.setMalop(rs.getString(3));

sv.setHedaotao(rs.getString(4));

sv.setNgaysinh(new Date(rs.getDate(5).getTime()));

sv.setDiachi(rs.getString(6));

sv.setGioitinh(rs.getBoolean(7));

sv.setSodt(rs.getString(8));

list.add(sv);

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<SinhVien> findByIDLop(String maLop) {

ArrayList<SinhVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("select \* from SinhVien where MaLop = ?");

ps.setString(1, maLop);

rs = ps.executeQuery();

list = new ArrayList<>();

while (rs.next()) {

SinhVien sv = new SinhVien();

sv.setMasv(rs.getString(1));

sv.setTensv(rs.getString(2));

sv.setMalop(rs.getString(3));

sv.setHedaotao(rs.getString(4));

sv.setNgaysinh(new Date(rs.getDate(5).getTime()));

sv.setDiachi(rs.getString(6));

sv.setGioitinh(rs.getBoolean(7));

sv.setSodt(rs.getString(8));

list.add(sv);

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public SinhVien addNew(SinhVien sv) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into SinhVien(MaSV,HoTen,MaLop,HeDaoTao,NgaySinh,DiaChi,GioiTinh,SDT) values (?,?,?,?,?,?,?,?)");

ps.setNString(1, sv.getMasv());

ps.setNString(2, sv.getTensv());

ps.setNString(3, sv.getMalop());

ps.setNString(4, sv.getHedaotao());

ps.setDate(5, new java.sql.Date(sv.getNgaysinh().getTime()));

ps.setNString(6, sv.getDiachi());

ps.setBoolean(7, sv.isGioitinh());

ps.setNString(8, sv.getSodt());

int row = ps.executeUpdate();

if (row < 1) {

sv = null;

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

sv = null;

} finally {

DBConnect.close(ps);

}

}

return sv;

}

@Override

public SinhVien updateByID(SinhVien sv) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update SinhVien set HoTen = ?, MaLop = ?, HeDaoTao = ?, NgaySinh = ?, DiaChi = ?, GioiTinh = ?, SDT = ? where MaSV = ?");

ps.setString(1, sv.getTensv());

ps.setString(2, sv.getMalop());

ps.setString(3, sv.getHedaotao());

ps.setDate(4, new java.sql.Date(sv.getNgaysinh().getTime()));

ps.setString(5, sv.getDiachi());

ps.setBoolean(6, sv.isGioitinh());

ps.setString(7, sv.getSodt());

ps.setString(8, sv.getMasv());

if (ps.executeUpdate() < 1) {

sv = null;

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close();

}

}

return sv;

}

public void deleteIDSinhVien(String SinhVienID) throws SQLException, ClassNotFoundException {

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from SinhVien where MaSV= ?");

ps.setString(1, SinhVienID);

ps.executeUpdate();

DBConnect.close();

}

}

public static void main(String[] args) {

System.out.println(new SinhVienDAO().findByIDLop("").get(0).getMalop());

}

@Override

public ArrayList<SinhVien> CheckID(String masv) {

ArrayList<SinhVien> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \*from SinhVien where MaSV=?");

psCheck.setString(1, masv);

rs = psCheck.executeQuery();

list = new ArrayList<SinhVien>();

while (rs.next()) {

SinhVien sinhVien = new SinhVien();

sinhVien.setMasv(rs.getString(1));

list.add(sinhVien);

}

} catch (SQLException ex) {

Logger.getLogger(SinhVienDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

DBConnect.close(psCheck, rs);

}

}

return list;

}

}