Model class - Các lớp mô tả dữ liệu;

***1.Giáo viên***

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package GiaoVien;

import MonHoc.MonHoc;

import java.util.ArrayList;

import java.util.Date;

/\*\*

\*

\* @author Anonymous

\*/

public class GiaoVien {

private String magv;

private String hotengv;

private Date ngaysinh;

private boolean gioitinh;

private String email;

private String diachi;

private String sdt;

private String mamh;

private ArrayList<MonHoc> listMh;

public ArrayList<MonHoc> getListMh() {

return listMh;

}

public void setListMh(ArrayList<MonHoc> listMh) {

this.listMh = listMh;

}

public GiaoVien() {

}

public GiaoVien(String magv, String hotengv, Date ngaysinh, boolean gioitinh, String email, String diachi, String sdt, String mamh) {

this.magv = magv;

this.hotengv = hotengv;

this.ngaysinh = ngaysinh;

this.gioitinh = gioitinh;

this.email = email;

this.diachi = diachi;

this.sdt = sdt;

this.mamh = mamh;

}

public void setMamh(String mamh) {

this.mamh = mamh;

}

public String getMamh() {

return mamh;

}

public String getMagv() {

return magv;

}

public void setMagv(String magv) {

this.magv = magv;

}

public String getHotengv() {

return hotengv;

}

public void setHotengv(String hotengv) {

this.hotengv = hotengv;

}

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 getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getDiachi() {

return diachi;

}

public void setDiachi(String diachi) {

this.diachi = diachi;

}

public String getSdt() {

return sdt;

}

public void setSdt(String sdt) {

this.sdt = sdt;

}

}

***2. Khoa***

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package Khoa;

/\*\*

\*

\* @author Anonymous

\*/

public class Khoa {

private String makhoa;

private String tenkhoa;

private String sdt;

public Khoa() {

}

public Khoa(String makhoa, String tenkhoa, String sdt) {

this.makhoa = makhoa;

this.tenkhoa = tenkhoa;

this.sdt = sdt; }

public String getMakhoa() {

return makhoa;

}

public void setMakhoa(String makhoa) {

this.makhoa = makhoa;

}

public String getTenkhoa() {

return tenkhoa;

}

public void setTenkhoa(String tenkhoa) {

this.tenkhoa = tenkhoa;

}

public String getSdt() {

return sdt;

}

public void setSdt(String sdt) {

this.sdt = sdt;

}

}

***3. Lớp học***

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package LopHoc;

/\*\*

\*

\* @author Anonymous

\*/

public class LopHoc {

private String malop;

private String tenlop;

private String makhoa;

private String khoahoc;

public LopHoc() {

}

public LopHoc(String malop, String tenlop, String makhoa, String khoahoc) {

this.malop = malop;

this.tenlop = tenlop;

this.makhoa = makhoa;

this.khoahoc = khoahoc;

}

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 String getMakhoa() {

return makhoa;

}

public void setMakhoa(String makhoa) {

this.makhoa = makhoa;

}

public String getKhoahoc() {

return khoahoc;

}

public void setKhoahoc(String khoahoc) {

this.khoahoc = khoahoc;

}

}

DAO Class - Các lớp truy xuất dữ liệu

1. ***GiaovienDAO***

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package GiaoVien;

import ConnectDatabase.DBConnect;

import MonHoc.MonHoc;

import MonHoc.MonHocDAO;

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;

/\*\*

\*

\* @author Anonymous

\*/

public class GiaoVienDAO implements IGiaoVienDAO {

@Override

public ArrayList<GiaoVien> getAll() {

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

ArrayList<GiaoVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

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

rs = ps.executeQuery();

list = new ArrayList<GiaoVien>();

while (rs.next()) {

GiaoVien gv = new GiaoVien();

gv.setMagv(rs.getString(1));

gv.setHotengv(rs.getString(2));

gv.setMamh(rs.getString(3));

gv.setNgaysinh(new Date(rs.getDate(4).getTime()));

gv.setGioitinh(rs.getBoolean(5));

gv.setEmail(rs.getString(6));

gv.setDiachi(rs.getString(7));

gv.setSdt(rs.getString(8));

list.add(gv);

}

} catch (SQLException ex) {

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

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<GiaoVien> findByIDMonHoc(String mamh) {

ArrayList<GiaoVien> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

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

ps.setString(1, mamh);

rs = ps.executeQuery();

list = new ArrayList<GiaoVien>();

while (rs.next()) {

GiaoVien gv = new GiaoVien();

gv.setMagv(rs.getString(1));

gv.setHotengv(rs.getString(2));

gv.setMamh(rs.getString(3));

gv.setNgaysinh(new Date(rs.getDate(4).getTime()));

gv.setGioitinh(rs.getBoolean(5));

gv.setEmail(rs.getString(6));

gv.setDiachi(rs.getString(7));

gv.setSdt(rs.getString(8));

list.add(gv);

}

} catch (SQLException ex) {

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

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public GiaoVien addNew(GiaoVien gv) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

// ps = DBConnect.cnn.prepareStatement("insert into tblGiangVien.fldMaGV, tblGiangVien.fldHoTenGV,tblGVMH.fldMaMH, tblGiangVien.fldNgaySinh, tblGiangVien.fldGioiTinh, tblGiangVien.fldEmail, tblGiangVien.fldDiaChi, tblGiangVien.fldSDT from tblGiangVien inner join tblGVMH on tblGiangVien.fldMaGV=tblGVMH.fldMaGV values (?,?,?,?,?,?,?,?)");

ps = DBConnect.cnn.prepareStatement("INSERT INTO GiangVien values (?,?,?,?,?,?,?,?)");

ps.setString(1, gv.getMagv());

ps.setString(2, gv.getHotengv());

ps.setString(3, gv.getMamh());

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

ps.setBoolean(5, gv.isGioitinh());

ps.setString(6, gv.getEmail());

ps.setString(7, gv.getDiachi());

ps.setString(8, gv.getSdt());

int row = ps.executeUpdate();

if (row < 1) {

gv = null;

}

} catch (SQLException ex) {

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

gv = null;

} finally {

DBConnect.close(ps);

}

}

return gv;

}

@Override

public GiaoVien updateByID(GiaoVien gv) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update GiangVien set HoTenGV =?,"

+ "MaMH= ?,NgaySinh=?,"

+ "GioiTinh=?, Email = ?, DiaChi = ?, "

+ "SDT = ? where MaGV = ?");

ps.setString(1, gv.getHotengv());

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

//ps.setDate(3, new java.sql.Date(new Date().getTime()));

ps.setDate(3, new java.sql.Date(gv.getNgaysinh().getTime()));

ps.setBoolean(4, gv.isGioitinh());

ps.setString(5, gv.getEmail());

ps.setString(6, gv.getDiachi());

ps.setString(7, gv.getSdt());

ps.setString(8, gv.getMagv());

int row = ps.executeUpdate();

if (row < 1) {

gv = null;

}

} catch (SQLException ex) {

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

gv = null;

} finally {

DBConnect.close();

}

}

return gv;

}

public void deleteIDGV(String GiaoVienID)throws SQLException,ClassNotFoundException{

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from GiangVien where MaGV= ?");

ps.setString(1, GiaoVienID);

ps.executeUpdate();

DBConnect.close();

}

}

public static void main(String[] args) {

System.out.println(new GiaoVienDAO().findByIDMonHoc("GV01").get(0).getMamh());

}

@Override

public ArrayList<GiaoVien> CheckID(String magv) {

ArrayList<GiaoVien> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

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

psCheck.setString(1, magv);

rs = psCheck.executeQuery();

list = new ArrayList<GiaoVien>();

while (rs.next()) {

GiaoVien giaoVien = new GiaoVien();

giaoVien.setMagv(rs.getString(1));

list.add(giaoVien);

}

} catch (SQLException ex) {

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

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}

***2. KhoaDAO***

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package Khoa;

import ConnectDatabase.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 KhoaDAO implements IKhoaDAO {

@Override

public ArrayList<Khoa> getAll() {

ArrayList<Khoa> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

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

rs = ps.executeQuery();

list = new ArrayList<Khoa>();

while (rs.next()) {

Khoa k = new Khoa();

k.setMakhoa(rs.getString(1));

k.setTenkhoa(rs.getString(2));

k.setSdt(rs.getString(3));

list.add(k);

}

} catch (SQLException ex) {

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

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public Khoa addNew(Khoa khoa) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("INSERT INTO Khoa values (?,?,?)");

ps.setString(1, khoa.getMakhoa());

ps.setString(2, khoa.getTenkhoa());

ps.setString(3, khoa.getSdt());

int row = ps.executeUpdate();

if (row < 1) {

khoa = null;

}

} catch (SQLException ex) {

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

khoa = null;

}finally{

DBConnect.close(ps);

}

}

return khoa;

}

@Override

public Khoa updateByID(Khoa khoa) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update Khoa set TenKhoa =?, SDT = ? where MaKhoa = ?");

ps.setString(1, khoa.getTenkhoa());

ps.setString(2, khoa.getSdt());

ps.setString(3, khoa.getMakhoa());

int row = ps.executeUpdate();

if (row < 1) {

khoa = null;

}

} catch (SQLException ex) {

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

khoa = null;

}finally{

DBConnect.close();

}

}

return khoa;

}

public void deleteKhoa(String KhoaID) throws SQLException, ClassNotFoundException{

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from Khoa where MaKhoa = ?");

ps.setString(1, KhoaID);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public ArrayList<Khoa> checkID(String makhoa) {

ArrayList<Khoa> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

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

psCheck.setString(1, makhoa);

rs = psCheck.executeQuery();

list = new ArrayList<Khoa>();

while (rs.next()) {

Khoa k = new Khoa();

k.setMakhoa(rs.getString(1));

list.add(k);

}

} catch (SQLException ex) {

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

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}

***3. LophocDAO***

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package LopHoc;

import ConnectDatabase.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 LopHocDAO implements ILopHocDAO {

@Override

public ArrayList<LopHoc> getAll() {

ArrayList<LopHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

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

rs = ps.executeQuery();

list = new ArrayList<LopHoc>();

while (rs.next()) {

LopHoc lh = new LopHoc();

lh.setMalop(rs.getString(1));

lh.setTenlop(rs.getString(2));

lh.setMakhoa(rs.getString(3));

lh.setKhoahoc(rs.getString(4));

list.add(lh);

}

} catch (SQLException ex) {

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

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public ArrayList<LopHoc> findByIDKhoa(String maKhoa) {

ArrayList<LopHoc> list = null;

PreparedStatement ps = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

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

ps.setString(1, maKhoa);

rs = ps.executeQuery();

list = new ArrayList<LopHoc>();

while (rs.next()) {

LopHoc lh = new LopHoc();

lh.setMalop(rs.getString(1));

lh.setTenlop(rs.getString(2));

lh.setMakhoa(rs.getString(3));

lh.setKhoahoc(rs.getString(4));

list.add(lh);

}

} catch (SQLException ex) {

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

} finally {

DBConnect.close(ps, rs);

}

}

return list;

}

@Override

public LopHoc addNew(LopHoc lh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("insert into Lop(MaLop,TenLop,MaKhoa,KhoaHoc) values(?,?,?,?)");

ps.setString(1, lh.getMalop());

ps.setString(2, lh.getTenlop());

ps.setString(3, lh.getMakhoa());

ps.setString(4, lh.getKhoahoc());

int row = ps.executeUpdate();

if (row < 1) {

lh = null;

}

} catch (SQLException ex) {

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

lh = null;

} finally {

DBConnect.close(ps);

}

}

return lh;

}

@Override

public LopHoc updateByID(LopHoc lh) {

PreparedStatement ps = null;

if (DBConnect.open()) {

try {

ps = DBConnect.cnn.prepareStatement("update Lop set TenLop = ?,MaKhoa=?, KhoaHoc = ? where MaLop = ?");

ps.setString(1, lh.getTenlop());

ps.setString(2, lh.getMakhoa());

ps.setString(3, lh.getKhoahoc());

ps.setString(4, lh.getMalop());

int row = ps.executeUpdate();

if (row < 1) {

lh = null;

}

} catch (SQLException ex) {

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

lh = null;

} finally {

DBConnect.close();

}

}

return lh;

}

public void deleteLopHoc(String LopID)throws SQLException, ClassNotFoundException{

PreparedStatement ps = null;

if (DBConnect.open()) {

ps = DBConnect.cnn.prepareStatement("delete from Lop where MaLop= ?");

ps.setString(1, LopID);

ps.executeUpdate();

DBConnect.close();

}

}

@Override

public ArrayList<LopHoc> checkID(String malop) {

ArrayList<LopHoc> list = null;

PreparedStatement psCheck = null;

ResultSet rs = null;

if (DBConnect.open()) {

try {

psCheck = DBConnect.cnn.prepareStatement("select \* from Lop where MaLop");

psCheck.setString(1, malop);

rs = psCheck.executeQuery();

list = new ArrayList<LopHoc>();

while (rs.next()) {

LopHoc lh = new LopHoc();

lh.setMalop(rs.getString(1));

list.add(lh);

}

} catch (SQLException ex) {

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

}finally{

DBConnect.close(psCheck, rs);

}

}

return list;

}

}