서버프로그램 구현 작업지시서

| **훈련 과정** | (디지털컨버전스)디지털 융합 자바 웹개발자 양성과정(백엔드, 프론트엔드) BL-2 | | | **훈련 기간** | 2023.06.28. - 2023.11.29 |
| --- | --- | --- | --- | --- | --- |
| **교 과 목** | 디지털SW융합 구현 | | | **능력단위명** | 서버프로그램 구현 |
| **평가 도구** | 평가자 체크리스트 | **평가 시간** | 60분 | **제출 방법** | 본인 LMS 직접 첨부 |
| **교 강 사** | 원윤희 | | | **훈련생명** | 이초희 |

**➜ 본 평가는 통합구현 1번-5번 문항과 이어지는 출제 문항이오니 참조바랍니다.**

| 회원정보 데이터의 항목은 아이디, 성명, 암호, 이메일, 전화번호 등이 있다.  아이디는 회원을 구분하기 위해 사용하는 데이터 항목이며 중복되어서는 안되고, 시스템에 접속하기 위해서는 아이디와 암호가 일치하는 경우만 접속이 가능하며 회원정보중 수정 가능한 데이터는 아이디를 제외한 모든 데이터는 수정 가능하다. |
| --- |

1. [난이도:하] 데이터 연계를 위한 회원정보를 위한 데이터를 정의하고 테이블(MEMBER)을 생성하시오. [배점:10]

CREATE TABLE MEMBER (

ID VARCHAR2(20) PRIMARY KEY,

NAME VARCHAR2(30) NOT NULL,

PASSWORD VARCHAR2(20) NOT NULL,

EMAIL VARCHAR2(20)

PHONE VARCHAR2(20)

);

2. [난이도:중] 공통으로 사용할 회원정보 데이터 저장용 클래스(MemberVO)를 작성하시오. [배점:10]

package com.mystudy.test.vo;

public class MemberVO {

String id;

String name;

String password;

String email;

String phone;

public MemberVO() {}

public MemberVO(String id, String name, String password, String email, String phone) {

this.id = id;

this.name = name;

this.password = password;

this.email = email;

this.phone = phone;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

@Override

public String toString() {

return "MemberVO [id=" + id + ", name=" + name + ", password=" + password + ", email=" + email + ", phone="

+ phone + "]";

}

}

3. [난이도:상] 데이터를 처리하기 위한 서버 공통 프로그램 MemberDAO의 데이터 검색 기능을 구현하시오. [배점:10]

- 전체데이터 조회 : selectAll()

import java.util.List;

import com.mystudy.test.vo.MemberVO;

public class MemberDAO {

private static final String DRIVER = "oracle.jdbc.OracleDriver";

private static final String URL = "jdbc:oracle:thin:@localhost:1521:xe";

private static final String USER = "mystudy";

private static final String PASSWORD = "mystudypw";

private Connection conn = null;

private PreparedStatement pstmt = null;

private ResultSet rs = null;

static {

try {

Class.forName(DRIVER);

System.out.println(">> JDBC 드라이버 로딩 성공");

} catch (ClassNotFoundException e) {

System.out.println("[예외발생] JDBC 드라이버 로딩 실패!!!");

}

}

//SELECT : 테이블 전체 데이터 조회 - selectAll : List<MemberVO>

public List<MemberVO> selectAll() {

List<MemberVO> list = null;

try {

conn = DriverManager.getConnection(URL, USER, PASSWORD);

StringBuilder sb = new StringBuilder();

sb.append("SELECT ID, NAME, PASSWORD, EMAIL, PHONE ");

sb.append(" FROM MEMBER ");

pstmt = conn.prepareStatement(sb.toString());

rs = pstmt.executeQuery();

list = new ArrayList<MemberVO>();

while (rs.next()) {

MemberVO vo = new MemberVO(

rs.getString("ID"),

rs.getString("NAME"),

rs.getString("PASSWORD"),

rs.getString("EMAIL"),

rs.getString("PHONE"));

list.add(vo);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

if(conn != null) conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

try {

if(rs != null) rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

try {

if(pstmt != null) pstmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

return list;

}

}

4. [난이도:상] 서버 공통 프로그램 MemberDAO의 데이터 1개를 검색하는 기능을 구현하시오. [배점:10]

- 1개 데이터 조회 : selectOne(String id)

public MemberVO selectOne(String id) {

MemberVO vo = null;

try {

conn = DriverManager.getConnection(URL, USER, PASSWORD);

StringBuilder sb = new StringBuilder();

sb.append("SELECT ID, NAME, PASSWORD, EMAIL, PHONE ");

sb.append(" FROM MEMBER ");

pstmt = conn.prepareStatement(sb.toString());

pstmt.setString(1, id);

rs = pstmt.executeQuery();

if (rs.next()) {

vo = new MemberVO(

rs.getString("ID"),

rs.getString("NAME"),

rs.getString("PASSWORD"),

rs.getString("EMAIL"),

rs.getString("PHONE"));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

if(conn != null) conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

try {

if(rs != null) rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

try {

if(pstmt != null) pstmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

return vo;

}

5. [난이도:상] 서버 공통 프로그램 MemberDAO의 회원정보 데이터 입력기능을 구현하시오. [배점:10]

- 입력기능 : insert(MemberVO vo)

public int insert(MemberVO vo) {

int result = 0;

try {

conn = DriverManager.getConnection(URL, USER, PASSWORD);

StringBuilder sql = new StringBuilder();

sql.append("INSERT INTO STUDENT ");

sql.append(" (ID, NAME, PASSWORD, EMAIL, PHONE ");

sql.append("VALUES (?, ?, ?, ?, ?) ");

pstmt = conn.prepareStatement(sql.toString());

int i = 1;

pstmt.setString(i++, vo.getId());

pstmt.setString(i++, vo.getName());

pstmt.setString(i++, vo.getPassword());

pstmt.setString(i++, vo.getEmail());

pstmt.setString(i++, vo.getPhone());

result = pstmt.executeUpdate();

} catch (SQLException e) {

//e.printStackTrace();

System.out.println("[예외발생] 작업중 예외가 발생 : " + e.getMessage());

result = -1;

} finally {

try {

if(conn != null) conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

try {

if(pstmt != null) pstmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

return result;

}