0405 조별과제 분석 및 소스 제출

# 분석

분석1) 개발환경

Oracle 데이터베이스와 객체지향인 Java를 자원을 인식시키는 과정을 통해 두 시스템 간의 연동이 이루어진 환경에서 개발한다.

분석2) dao dto역할

dao : 다른 시스템인 Oracle 데이터베이스에 접속하여 CRUD 작업 한다.

dto : JVM과 오라클을 연동해서 데이터를 전송하는 역할이다.

분석3) 클래스 설계 분석

Main : 프로그램의 메인을 담당한다.

GoodPlaceADM : 프로그램의 메뉴를 담당한다.

GoodPlaceDTO : 회원들에게 입력받은 맛집의 상호명, 별점, 지역명을 저장하여 오라클에 데이터를 전송한다.

GoodPlaceDAO : Oracle 데이터베이스에 연동하여 goodplace 테이블에 CRUD 작업한다.

GoodAttachDTO : 회원들에게 입력받은 맛집의 댓글을 저장하여 오라클에 데이터를 전송한다.

GoodAttachDAO : Oracle 데이터베이스에 연동하여 goodattach 테이블에 CRUD 작업한다.

분석4) 패키지란 무엇인가?

유사한 기능을 가진 클래스들을 모은 폴더와 같은 개념이다.

분석5) ojdbc란 무엇인가?

Oracle 데이터베이스를 인식할 수 있는 드라이브 이름이다.

분석6) 맛집을 등록하는 과정을 시작 메서드 부터 설명하시오

GoodPlaceDTO 클래스에서 접근제어자가 private이고, 자료형이 String인 상호명, 별점, 지역명을 각각 선언하고, 초기값을 null 값으로 잡는다.

GoodPlaceADM 클래스에서 상호명, 별점, 지역명을 입력받아 GoodPlaceDTO 객체를 생성하여 그 주소값을 변수 g에 저장하고,

setName(), setStarp(), setCity() 메서드를 통해 주소값을 저장하고,

GoodPlaceDAO 클래스에서 정의한 insert 메서드를 호출하여 DB에 insert 작업을 한다.

분석7) 맛집의 정보를 보는 과정을 시작 메서드 부터 설명하시오.

맛집의 정보들을 더 유동적으로 관리하기 위해 ArrayList를 사용하여 GoodPlaceDAO 클래스에서 정의한 selectAll() 메서드를 호출하여

glist에 저장된 값들을 하나씩 불러온다.

만약 selectAll()메서드를 실행할 때, boolean 자료형인 변수 flag로 참과 거짓을 받아와 거짓일 경우 select \* from goodplace 명령문을 실행한다.

맛집 리스트를 출력한 다음, 댓글을 입력 받는다.

만약 메뉴에서 별점순으로 보기를 선택할 경우 selectAll() 메서드의 flag는 true로 값을 넘겨 별점 오름차순으로 정렬하여 리스트를 출력한다.

분석8) statement와 preparedstatement의 공통점과 차이점을 설명하시오

statement와 preparedstatement의 공통점은 쿼리문을 만들기 위한 클래스이며,

statement는 매개변수는 없지만 리턴값은 있고, preparedstatement는 매개변수는 있고 리턴값은 있을 수도 있고, 없을 수도 있다.

즉 statement는 넘겨줄 데이터가 없을 경우에 사용하고, preparedstatement는 넘겨줄 데이터가 있을 때 사용한다.

분석9) 매핑이란 무엇인가?

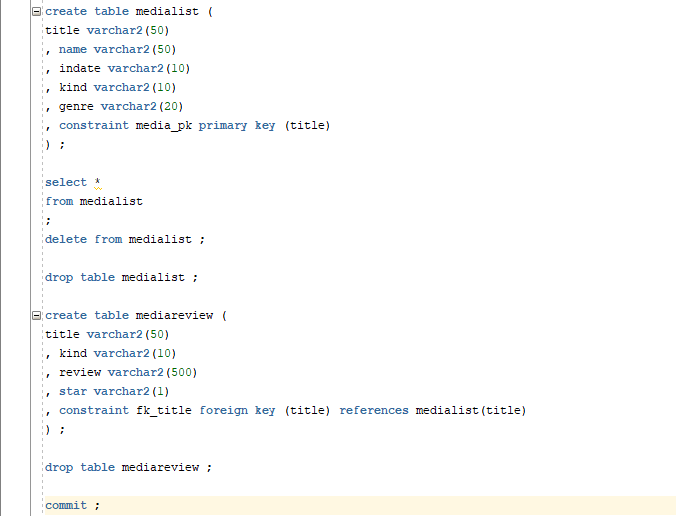
완성되지 않은 구문의 자리인 ?에 값을 지정해주는 것을 의미한다.

분석10) ResultSet 객체에 대해서 설명하시오. 이 객체를 어떻게 사용했는지 설명하시오.

쿼리문의 결과를 저장하기 위한 객체이다. Oracle 데이터베이스에 연결한 것에 변수 sql의 값을 넘긴다.

이때, 변수 sql의 값은 Oracle 데이터베이스의 쿼리문이다.

소스 첨부



**package** main;

**import** manager.MideaADM;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**new** MideaADM() ;

//MediaListDAO mldao = new MediaListDAO() ; // 연결 테스팅 코드

}

}

package manager;

import java.util.ArrayList;

import java.util.Scanner;

import dao.MediaListDAO;

import dao.MediaReviewDAO;

import dto.MediaListDTO;

import dto.MediaReviewDTO;

public class MediaADM {

private MediaListDAO mldao = new MediaListDAO();

private MediaReviewDAO mrdao = new MediaReviewDAO();

private Scanner in = new Scanner(System.in);

public MediaADM() {

while (true) {

System.out.println("[감상평 프로그램]");

System.out.println("1. 감상평 등록");

System.out.println("2. 감상평 수정");

System.out.println("3. 감상평 삭제");

System.out.println("4. 등록된 감상평 전체보기");

System.out.println("5. 분류별 감상평 보기(미완성)");

System.out.println("메뉴 번호를 입력해주세요.");

int selNum = in.nextInt();

in.nextLine();

if (selNum == 1) {

add();

} else if (selNum == 2) {

mod();

} else if (selNum == 3) {

del();

} else if (selNum == 4) {

allList();

} else if (selNum == 5) {

kindList();

} else {

break;

}

}

}

private void add() {

// TODO Auto-generated method stub

while (true) {

System.out.println("[관람평 등록]");

System.out.println("작성자명을 입력해주세요.");

System.out.println("영어+숫자 10자");

String id = in.nextLine();

System.out.println("비밀번호를 입력해주세요.");

System.out.println("비밀번호는 영어+숫자 최대 20자 입니다.");

String pw = in.nextLine();

System.out.println("관람평을 작성하실 분류명을 입력해주세요.");

System.out.println("(영화 / 책)");

String kind = in.nextLine();

if (kind.equals("영화") || kind.equals("책")) { // 분류명 제대로 입력했을 경우

System.out.println("작품명을 입력해주세요.");

System.out.println("한글 기준 최대 25자");

String title = in.nextLine();

System.out.println("감독 또는 지은이를 입력해주세요.");

System.out.println("한글 기준 최대 25자");

String name = in.nextLine();

System.out.println("기록 날짜를 입력해주세요.");

System.out.println("양식) yyyy-mm-dd");

String indate = in.nextLine();

System.out.println("장르를 입력해주세요.");

String genre = in.nextLine();

MediaListDTO mldto = new MediaListDTO();

mldto.setId(id); // MediaListDTO에 입력받은 내용 저장

mldto.setPw(pw);

mldto.setTitle(title);

mldto.setName(name);

mldto.setIndate(indate);

mldto.setKind(kind);

mldto.setGenre(genre);

mldao.insert(mldto); // 오라클로 입력 데이터 전송

review(id, title, kind); // review 메서드로 입력받은 아이디 제목 분류 넘김

break;

} else {

System.out.println("분류를 잘못입력하셨습니다.");

}

}

}

private void review(String id, String title, String kind) {

// System.out.println("작성자명을 입력해주세요.");

// System.out.println("한글 기준 최대 5자");

// String id = in.nextLine();

//

// System.out.println("작품명을 입력해주세요.");

// System.out.println("한글 기준 최대 25자");

// String title = in.nextLine();

System.out.println("별점을 등록해주세요. (5점 만점)");

String star = in.nextLine();

System.out.println("감상평을 등록해주세요. (한글 기준 최대 250자)");

String review = in.nextLine();

// System.out.println("관람평을 작성하실 분류명을 입력해주세요.");

// System.out.println("(영화 / 책)");

// String kind = in.nextLine();

MediaReviewDTO mrdto = new MediaReviewDTO();

mrdto.setId(id);

mrdto.setTitle(title);

mrdto.setKind(kind);

mrdto.setReview(review);

mrdto.setStar(star);

mrdao.rvInsert(mrdto);

}

private void mod() {

// TODO Auto-generated method stub

allList(); // 등록된 리스트 출력

System.out.println("[감상평 수정]");

System.out.println("수정하고 싶은 글의 작성자명을 입력해주세요.");

String id = in.nextLine();

System.out.println("비밀번호를 입력해주세요.");

String pw = in.nextLine();

System.out.println("리뷰를 다시 입력해주세요.");

String review = in.nextLine();

mrdao.modRv(id, review);

}

private void del() {

// TODO Auto-generated method stub

allList() ;

System.out.println("[감상평 삭제]");

System.out.println("삭제하고 싶은 글의 작성자명을 입력해주세요.");

String id = in.nextLine();

mrdao.delRv(id);

mldao.delMl(id);

}

private void allList() {

// TODO Auto-generated method stub

int count = 1 ;

System.out.println("[등록된 관람평 전체보기]");

ArrayList<MediaListDTO> mList = mldao.selectAll(); // 등록된 작품명 보기.

for (MediaListDTO m : mList) {

System.out.println(count + "번 리뷰 ----------------------------------");

System.out.println("제목 : " + m.getTitle());

System.out.println("감독/지은이 : " + m.getName());

System.out.println("날짜 : " + m.getIndate());

System.out.println("분류 : " + m.getKind());

System.out.println("장르 : " + m.getGenre());

rvList(m.getTitle());

count++;

}

}

private void rvList(String title) {

MediaReviewDTO mr = new MediaReviewDTO() ;

// System.out.println("등록된 리뷰를 볼 작품명을 입력해주세요.");

// String title = in.nextLine();

ArrayList<MediaReviewDTO> rList = mrdao.reSelect(title);

System.out.println();

for (MediaReviewDTO m : rList) {

System.out.println("작성자명 : " + m.getId());

System.out.println("별점 : " + m.getStar());

System.out.println("내용 |");

System.out.println(": " + m.getReview());

System.out.println();

}

}

private void kindList() {

// TODO Auto-generated method stub

System.out.println("[분류별 리뷰 모음]");

System.out.println("원하시는 분류를 입력해주세요. (영화 / 책)");

System.out.println("리뷰는 별점순입니다.");

String kind = in.nextLine();

mrdao.kindStar(kind);

}

}

**package** dto;

**public** **class** MediaListDTO {

**private** String id = **null** ; // 작성자(한글 기준 5자)

**private** String pw = **null** ; // 비밀번호

**private** String title = **null** ; // 작품명(한글 기준 25자)

**private** String name = **null** ; // 감독 또는 지은이 (한글 기준 25자)

**private** String indate = **null** ; // 본 날짜, 혹은 완독 날짜 (양식 : xxxx-xx-xx)

**private** String kind = **null** ; // 분류(책, 영화) (한글기준 5글자)

**private** String genre = **null** ; // 장르 (한글기준 10글자)

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getPw() {

**return** pw;

}

**public** **void** setPw(String pw) {

**this**.pw = pw;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getIndate() {

**return** indate;

}

**public** **void** setIndate(String indate) {

**this**.indate = indate;

}

**public** String getKind() {

**return** kind;

}

**public** **void** setKind(String kind) {

**this**.kind = kind;

}

**public** String getGenre() {

**return** genre;

}

**public** **void** setGenre(String genre) {

**this**.genre = genre;

}

}

package dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import dto.MediaListDTO;

public class MediaListDAO {

private Connection conn = null ;

private String url = "jdbc:oracle:thin:@localhost:1521:orcl" ;

public MediaListDAO() {

String driver = "oracle.jdbc.driver.OracleDriver";

try {

Class.forName(driver);

//System.out.println("로드 성공");

getConnection() ;

} catch (Exception e) {

// TODO: handle exception

System.out.println("로드 실패");

}

}

public Connection getConnection() {

// TODO Auto-generated method stub

try {

conn = DriverManager.getConnection(url, "system", "1111") ;

//System.out.println("오라클 연결 됨");

return conn ;

} catch (Exception e) {

// TODO: handle exception

System.out.println("오라클 연결 실패");

}

return null;

}

public void insert(MediaListDTO mldto) {

String sql = "insert into medialist values ( ?, ?, ?, ?, ?, ?, ? )" ;

if (getConnection() != null) {

try {

PreparedStatement psmt = null ;

psmt = conn.prepareStatement(sql) ;

psmt.setString(1, mldto.getId()) ;

psmt.setString(2, mldto.getPw());

psmt.setString(3, mldto.getTitle()) ;

psmt.setString(4, mldto.getName());

psmt.setString(5, mldto.getIndate());

psmt.setString(6, mldto.getKind());

psmt.setString(7, mldto.getGenre());

psmt.executeUpdate() ;

System.out.println("작품정보가 입력되었습니다.");

System.out.println("--------------리뷰--------------");

} catch (Exception e) {

// TODO: handle exception

System.out.println("insert문 오류");

}

}

}

public ArrayList<MediaListDTO> selectAll() {

ArrayList<MediaListDTO> ml = new ArrayList<>() ;

if (getConnection() != null) {

try {

String sql = "select \* from medialist" ;

Statement stmt = null ;

stmt = conn.createStatement() ;

ResultSet rs = null ;

rs = stmt.executeQuery(sql);

while(rs.next()) {

MediaListDTO mldto = new MediaListDTO() ;

mldto.setId(rs.getString("id"));

mldto.setPw(rs.getString("pw"));

mldto.setTitle(rs.getString("title"));

mldto.setName(rs.getString("name"));

mldto.setIndate(rs.getString("idate"));

mldto.setKind(rs.getString("kind"));

mldto.setGenre(rs.getString("genre"));

ml.add(mldto) ;

}

} catch (Exception e) {

// TODO: handle exception

System.out.println("등록된 리스트 출력 오류입니다.");

}

}

return ml;

}

public void delMl(String id) {

if (getConnection() != null) {

String sql = "delete from medialist where id = ?";

try {

PreparedStatement psmt = null ;

psmt = conn.prepareStatement(sql);

psmt.setNString(1, id);

psmt.executeUpdate();

System.out.println("삭제되었습니다.");

} catch (Exception e) {

// TODO: handle exception

System.out.println("리스트 삭제 오류");

}

}

}

}

**package** dto;

**public** **class** MediaReviewDTO {

**private** String id = **null** ; // 작성자명

**private** String title = **null** ; // 제목

**private** String kind = **null** ; // 분류

**private** String review = **null** ; // 리뷰(한글 기준 최대 250자)

**private** String star = **null** ; // 별점(5점만점)

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** String getKind() {

**return** kind;

}

**public** **void** setKind(String kind) {

**this**.kind = kind;

}

**public** String getReview() {

**return** review;

}

**public** **void** setReview(String review) {

**this**.review = review;

}

**public** String getStar() {

**return** star;

}

**public** **void** setStar(String star) {

**this**.star = star;

}

}

package dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import dto.MediaReviewDTO;

public class MediaReviewDAO {

private Connection conn = null ;

private String url = "jdbc:oracle:thin:@localhost:1521:orcl" ;

public MediaReviewDAO() {

String driver = "oracle.jdbc.driver.OracleDriver" ;

try {

Class.forName(driver) ;

//System.out.println("드라이브 로드 성공");

getConnection() ; // 테스트용

} catch (Exception e) {

// TODO: handle exception

System.out.println("드라이브 로드 실패");

}

}

public Connection getConnection() {

try {

conn = DriverManager.getConnection(url, "system", "1111") ;

//System.out.println("연결 성공했습니다.");

return conn ;

} catch (Exception e) {

// TODO: handle exception

System.out.println("연결이 실패되었습니다.");

}

return null;

}

public void rvInsert(MediaReviewDTO mrdto) { // 리뷰 입력 method 정의

String sql = "insert into mediareview values (?, ?, ?, ?, ?)" ;

try {

PreparedStatement pstm = null ;

pstm = conn.prepareStatement(sql);

pstm.setString(1, mrdto.getId());

pstm.setString(2, mrdto.getTitle());

pstm.setString(3, mrdto.getKind());

pstm.setString(4, mrdto.getReview());

pstm.setString(5, mrdto.getStar());

pstm.executeUpdate();

System.out.println("리뷰가 입력되었습니다.");

} catch (Exception e) {

// TODO: handle exception

System.out.println("리뷰 입력 오류");

}

}

public ArrayList<MediaReviewDTO> reSelect(String title) { // 리뷰 전체보기 method 정의

ArrayList<MediaReviewDTO> mr = new ArrayList<>() ; // 자료가 유동적으로 변경되므로 ArrayList 사용

if (getConnection() != null) {

try {

String sql = "select \* from mediareview where title = ?" ;

PreparedStatement psmt = null ;

psmt = conn.prepareStatement(sql);

psmt.setString(1, title);

ResultSet rs = null ;

rs = psmt.executeQuery();

while(rs.next()) {

MediaReviewDTO m = new MediaReviewDTO() ;

m.setId(rs.getString("id"));

m.setTitle(rs.getString("title")) ;

m.setKind(rs.getString("kind"));

m.setReview(rs.getString("review"));

m.setStar(rs.getString("star"));

mr.add(m); // 리뷰 데이터 받아온 걸 다시 객체로 변경해 ArrayList에 추가

}

} catch (Exception e) {

// TODO: handle exception

System.out.println("리뷰 출력 오류");

}

}

return mr;

}

public void modRv(String id, String review) { // 리뷰 수정 method 정의

if (getConnection() != null) {

try {

String sql = "update mediareview set review = ? where id = ?" ;

PreparedStatement psmt = null ;

psmt = conn.prepareStatement(sql);

psmt.setString(1, review) ;

psmt.setString(2, id);

ResultSet rs = null ;

rs = psmt.executeQuery() ; // 쿼리문 오류날 경우 여기 오류

psmt.executeUpdate() ;

System.out.println("리뷰가 수정되었습니다.");

} catch (Exception e) {

// TODO: handle exception

System.out.println("리뷰 수정 오류");

}

}

}

public void delRv(String id) {

if(getConnection() != null) {

try {

String sql = "delete from mediareview where id = ?";

PreparedStatement psmt = null ;

psmt = conn.prepareStatement(sql);

psmt.setString(1, id);

ResultSet rs = null ;

rs = psmt.executeQuery();

psmt.executeUpdate();

System.out.println();

} catch (Exception e) {

// TODO: handle exception

System.out.println("리뷰 삭제 오류");

}

}

}

public void kindStar(String kind) {

if (getConnection() != null) {

try {

String sql = "select \* from mediareview where kind = ? order by star" ;

PreparedStatement psmt = null ;

psmt = conn.prepareStatement(sql);

psmt.setString(1, kind);

ResultSet rs = null ;

rs = psmt.executeQuery();

psmt.executeUpdate();

} catch (Exception e) {

// TODO: handle exception

System.out.println("분류별 출력 오류");

}

}

}

}

**package** dao;

**public** **class** MideaReviewDAO {

}