리팩토링

1. 싱글톤

: 객체를 한 개만 만들고 객체가 필요하면 주소를 주입받는다.

생성자에 private

싱글콘 클래스에 멤버변수로 자신의 변수 하나 만들기

getInstane메서드로 객체주소 리턴받기

1. 예외 처리

에러(error)와 예외(Exception)

에러는 실행이 되지 않는다.

예외는 실행 도중에 발생되는 오류

예외가 발생하면 프로그램이 오동작 할 수 있다.

그래서 예외가 발생하면 오동작이 나지 않도록

처리해 줘야 한다.

이런 구문이 try catch 문법입니다.

예외는 필수적으로 처리하는 경우도 있고

하지 않아도 되는 경우가 있다.

예를 들어 Thread.sleep() 기능은

예외를 필수적으로 처리해야 하는 기능이다.

예외를 처리하는 방법은 2가지 이다.

1가지는 예외를 직접처리하는 방법

다른 한가지는 예외를 떠 넘기는 방법

떠 넘길때는 호출한 메서드에 떠 넘긴다.

try{

//예외가 발생할 수 있는 코드

}catch ( ){

// 발생된 예외가 처리되는 코드

}catch ( ){

// 발생된 예외가 처리되는 코드

}finally{

// try 구문을 빠져나갈대 실행되는 코드

// 즉, 예외 발생 유무에 상관없이 실행

}

예외는 실행중에 발생하는 오류이다.

이 오류가 오작동이 되는 것을 방지하기 위해서

try구문을 사용한다.

try구문은 try catch finally가 있는데

try구문은 예외를 감지한다. 예외가 발생하면 catch 에서

처리해 준다. 예외가 발생하면 아래 코드는 실행하지 않는다.

finally 구문은 try 구문을 빠져 나갈때 실행된다..

**package** sample;

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

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

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

**new** UserAdmin();

}

}

**package** sample;

//책임 : 회원 한 명의 정보를 저장

**public** **class** UserDTO {

**private** String name;

**private** String id;

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** String getId() {

**return** id;

}

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

**this**.id = id;

}

**public** **void** prt() {

System.***out***.println("id : "+**this**.id);

System.***out***.println("name : "+**this**.name);

}

}

package sample;

import java.util.ArrayList;

import java.util.Scanner;

//책임 : 회원을 관리

public class UserAdmin {

private UserDAO userdao = UserDAO.getInstance();

UserAdmin(){

Scanner in = new Scanner(System.in);

boolean f = true;

while(f) {

System.out.println("1.등록 2.전체보기");

int a = in.nextInt();

in.nextLine();

switch (a) {

case 1: add();

break;

case 2: allList();

break;

default: f=false;

break;

}

}

}

private void add() {

Scanner in = new Scanner(System.in);

//사용자 추가

//회원의 이름과 아이디를 입력 받는다.

String name=in.nextLine();

String id = in.nextLine();

//회원 객체를 만든다..

UserDTO user = new UserDTO();

//회원 객체에 입력 받는 이름과 아이드를 매핑한다.

user.setName(name);

user.setId(id);

//회원 객체를 데이터베이스에 저장한다.

userdao.insert(user);

}

private void allList() {

// 전체보기 select \* from user1;

ArrayList<UserDTO> u = userdao.selectAll();

for(UserDTO uu: u) {

uu.prt();

}

}

}

package sample;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

//책임 : 데이터베이스에 CRUD작업 수행

public class UserDAO {

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

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

// DBMS주소:포트번호:데이터베이스 이름

private String id = "system11";

private String pass = "1111";

private Connection conn = null;

private static UserDAO userdao=null;

private UserDAO() {

init();

}

public static UserDAO getInstance() {

if(userdao == null) {

userdao = new UserDAO();

}

return userdao;

}

private Connection getConnection() {

try {

if(conn == null) {

conn = DriverManager.getConnection(url, id, pass);

}

System.out.println("연결성공");

} catch (Exception e) {

System.out.println("연결오류");

conn=null;

}

return conn;

}

private void init() {

try {

Class.forName(driver);

// 라이브러리를 로드하는 의미.

// 이 라이브러리는 오라클에 CURD를 위한 클래스들이 있음

System.out.println("클래스 load 성공");

} catch (ClassNotFoundException e) {

System.out.println("클래스 로드 실패");

e.printStackTrace(); //원인이 무엇인지 출력

}

}

public void insert(UserDTO u) {

PreparedStatement stmt = null;

try {

// 연결하기

if(getConnection()!=null) {

// 쿼리문정의

String sql = "insert into user1 values (?,?)";

// statement 생성

stmt = conn.prepareStatement(sql);

stmt.setString(1, u.getName());

stmt.setString(2, u.getId());

// 쿼리전송및 결과값 받기

int result = stmt.executeUpdate();

System.out.println(result + "건 삽입");

}

} catch (Exception e) {

// TODO: handle exception

} finally {

try {

// 연결해제-자원반납

stmt.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public ArrayList<UserDTO> selectAll() {

ArrayList<UserDTO> ulist = new ArrayList<>();

Statement stmt=null;

ResultSet rs=null;

try {

// 연결설정, 쿼리만들기, 매핑, 쿼리 실행,리턴값 받기

if(getConnection()!=null) {

System.out.println("연결성공");

String sql = "select \* from user1";

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

while (rs.next()) {

UserDTO u = new UserDTO();

u.setId(rs.getString("id")); // "id"는 컬럼명

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

ulist.add(u);

}

}

} catch (Exception e) {

} finally {

try {

// 연결해제-자원반납

rs.close();

stmt.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return ulist;

}

}