DTO, DAO, VO 실습

부제 : 부서관리시스템

1. DBConn.java

- 클래스 용도
- 1. JDBC 드라이버 로드
  2. Database Connection 연결
  3. Database Connection 해지 → 여기에선 수행하지 않음

```
package com.my;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConn {

  // Connection 메소드
  public static Connection getConnection() throws SQLException{
    // SQLException = 함수 예외처리
    // static : 공용공간에 올려놓고 계속 사용

    Connection conn = null; // DB 연결 위한 객체

    String jdbcDriver = "com.mysql.cj.jdbc.Driver";
    String jdbcUrl = "jdbc:mysql://localhost/empdb";
    String dbUser = "root"; // DB ID
    String dbPwd = "1234"; // DB PW

    conn = DriverManager.getConnection(jdbcUrl,dbUser, dbPwd);

    return conn;
  }
}
```

2. deptVO.java

- 클래스 용도
- 1. 멤버 필드 정의
  2. 생성자 정의
  3. 멤버 메소드 : getter

```
package com.my;
```

```
public class DeptVO {
  // 멤버 필드
  private int deptno;
  private String dname;
  private String loc;
  // 생성자
  DeptVO(){}  // 은닉 드러내야함
  DeptVO(int deptno, String dname, String loc){
    this.deptno = deptno; this.dname = dname; this.loc = loc;}
  // 멤버 메소드 : getter      (source -> Generate getter setter)
  public int getDeptno() { return deptno;}
  public String getDname() {return dname;}
  public String getLoc() {return loc;}

  @Override        // (source -> generate toString -> inheritence? -> toString)
  public String toString() {
    return "[ "+ deptno + " | " + dname + " | " + loc + " ]";


  }

}
```

3. deptDAO.java

- 클래스 용도
- CRUD (Create, Research, Update, Delete)
- DAO : 대부분 싱글톤* 으로 생성

```
package com.my;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class DeptDAO {
  // Singleton
  private static DeptDAO dao = new DeptDAO();
  private DeptDAO(){ }
  public static DeptDAO getInstance() { return dao; }
  /////////////////
  DeptVO selectDept(int deptno, Connection conn) { // selectDept() {} 부서명 -> 부서번호, 부서명, 부서위치

    DeptVO dept = null;
    try {
      String sql = "select * from dept10 where deptno = ?";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setInt(1, deptno);
      ResultSet rs = pstmt.executeQuery();

      while(rs.next()) {
        dept = new DeptVO(rs.getInt(1),rs.getString(2),rs.getString("loc"));
        System.out.printf("%d | %-10s | %-10s %n",rs.getInt("deptno"),rs.getString(2),rs.getString("loc"));
      }
      pstmt.close();
```

```
    }
    catch (SQLException e) {
      System.out.println("연결 안됨");
    }
    return dept;
  }

//  insertDept( ) { }
  int insertDept(DeptVO deptObj, Connection conn) {

    int resultCount = 0;

    try {
      String sql = "insert into dept10 values(?,?,?)";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setInt(1, deptObj.getDeptno());
      pstmt.setString(2, deptObj.getDname());
      pstmt.setString(3, deptObj.getLoc());

      resultCount = pstmt.executeUpdate();

      pstmt.close();
    }
    catch (SQLException e) {
      System.out.println("연결 안됨");
    }
    return resultCount;
  }

  //  updateDept( ) { }
  int updateDept(DeptVO deptObj, Connection conn) {

    int resultCount = 0;

    try {
      String sql = "update Dept10 d set d.loc =?, where d.deptno = ?";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setInt(2, deptObj.getDeptno());
      pstmt.setString(1, deptObj.getLoc());

      resultCount = pstmt.executeUpdate();

      pstmt.close();
    }
    catch (SQLException e) {
      System.out.println("연결 안됨");
    }
    return resultCount;
  }

  //  deleteDept( ) { }
  int deleteDept(int deptno, Connection conn) {

    int resultCount = 0;

    try {
      String sql = "delete from dept10 d where d.deptno = ?)";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setInt(1, deptno);  // 차이
      resultCount = pstmt.executeUpdate();

      pstmt.close();
    }
    catch (SQLException e) {
      System.out.println("연결 안됨");
    }
    return resultCount;
  }
```

```
  //  listDept() { }  부서명
  List <DeptVO> listDept(Connection conn) { //  selectDept() {} 부서명 -> 부서번호, 부서명, 부서위치

    /*array list 쓰는 이유 : array는 몇 행 몇 열 칸이 fix 늘렸다 줄였다 못함 , array list 는 유동적으 칸 크기를 조절 */
    List<DeptVO> deptlist = new ArrayList<DeptVO>();
    try {
      String sql = "select * from dept10";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      ResultSet rs = pstmt.executeQuery();

      while(rs.next()) {
        deptlist.add(new DeptVO(rs.getInt(1),rs.getString(2),rs.getString("3")));
        System.out.printf("%d | %-10s | %-10s %n",rs.getInt("deptno"),rs.getString(2),rs.getString("loc"));
      }
      pstmt.close();
    }
    catch (SQLException e) {
      System.out.println("연결 안됨");
    }
    return deptlist;
  }
}
```