**package** dbtest.dao;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement; // insert, delete, oracle 문장을 대신하는 가이드 역할

// 이 역시 interface. 직접 implement하는 건 바보같은 짓.

// 대신 implement해주는 클래스 역시 없으니... 메소드를 잡는 수밖에?

**import** java.sql.SQLException;

**import** java.util.Scanner;

**public** **class** InsertTest {

**private** Connection conn; // null값 갖고 있음. 고로 뉴 생성해줘야

**private** PreparedStatement pstmt;

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

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

**private** String username = "c##java";

**private** String password = "bit";

**public** InsertTest() {

**try** {

// driver loading

// Class.forName("풀네임, 패키지명.파일명"); 파일명을 주면 Class으로 생성해주겠다

Class.*forName*(driver);

System.***out***.println("드라이버 로딩 성공");

} **catch** (ClassNotFoundException e) {

e.printStackTrace(); // try부터 catch까지 발생하는 에러 내용을 출력

}

}

**public** **void** getConnection() {

**try** {

// DriverManager.getConnection("URL", "USERNAME", "PASSWORD");

// conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",

// "c##java", "bit");

conn = DriverManager.*getConnection*(url, username, password);

System.***out***.println("접속 성공");

} **catch** (SQLException e) {// SQLException이 떨어질 지 모른다

e.printStackTrace(); // 그걸 화면에 팍팍 찍어라

}

}

**public** **void** insertArticle() {

Scanner scan = **new** Scanner(System.***in***);

System.***out***.print("이름 입력 : ");

String name = scan.next();

System.***out***.print("나이 입력 : ");

**int** age = scan.nextInt();

System.***out***.print("키 입력 : ");

**double** height = scan.nextDouble();

**try** { // Exception 걸어주기

**this**.getConnection(); // 접속 - 접속(connection)한 뒤 sql문장 처리

// developer로 와서 기존에 데이터 다 지워버리기 (-> delete dbtest;)

// 이때 delete는 lock이 걸려있어 commit을 해주지 않으면 다른 곳에서는 삭제 됐는지 모른다

// 고로 지워졌다는 걸 갱신시켜주기 위해 commit; 입력

pstmt = conn.prepareStatement("insert into dbtest values(?, ?, ?, sysdate)");

// connection에 들어있는 prepareStatement 생성

// implement를 할 수도, implement를 대신 해주는 것도 없으니 메소드 잡아서 생성해준 것

// ?에 데이터 매핑

pstmt.setString(1, name); // Oracle은 java와 달리 index값이 1부터 시작.

pstmt.setInt(2, age);

pstmt.setDouble(3, height);

// 실행 - 개수 리턴

**int** su = pstmt.executeUpdate(); // db에서 insert into를 입력하면 항상 개수를 리턴시져줌

System.***out***.println(su + "개의 행이 만들어졌습니다");

// insert into dbtest values()

// 오라클 문장일뿐 자바코드가 아니니 가이드(PreparedStatement)를 따라가야

} **catch** (SQLException e) {

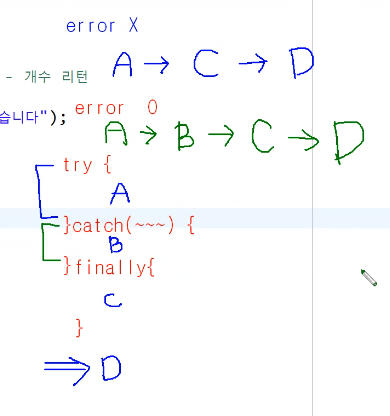
e.printStackTrace();

} **finally** {

// 에러가 있건 없건 무조건!! 접속했으면 끊어줘야!!

**try** {

**if** (pstmt != **null**)



pstmt.close();

**if** (conn != **null**)

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

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

InsertTest it = **new** InsertTest();

it.insertArticle();

}

}