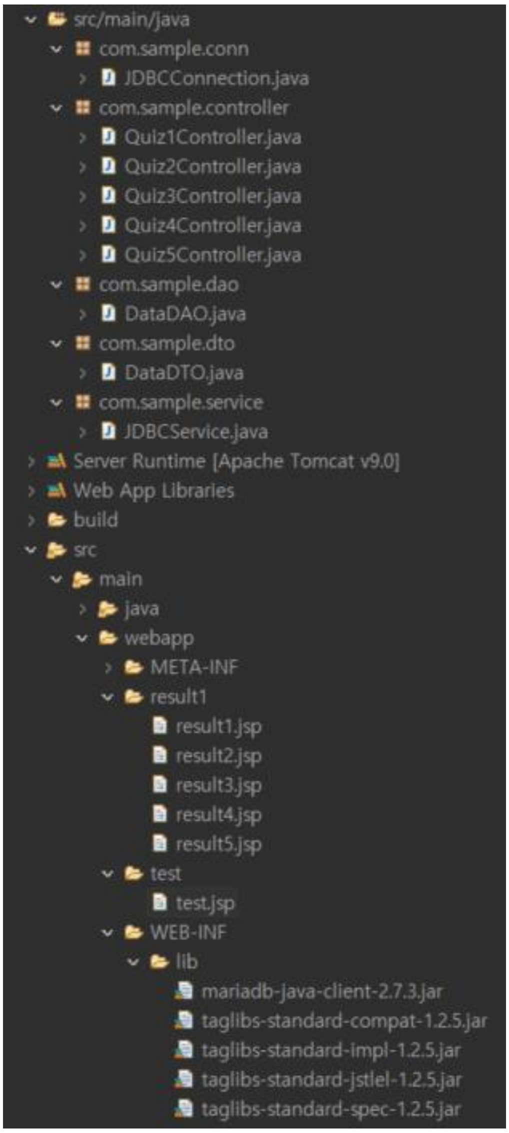
문제1. 다음과 같은 동작을 하는 페이지를 만들어라

<DB Set>

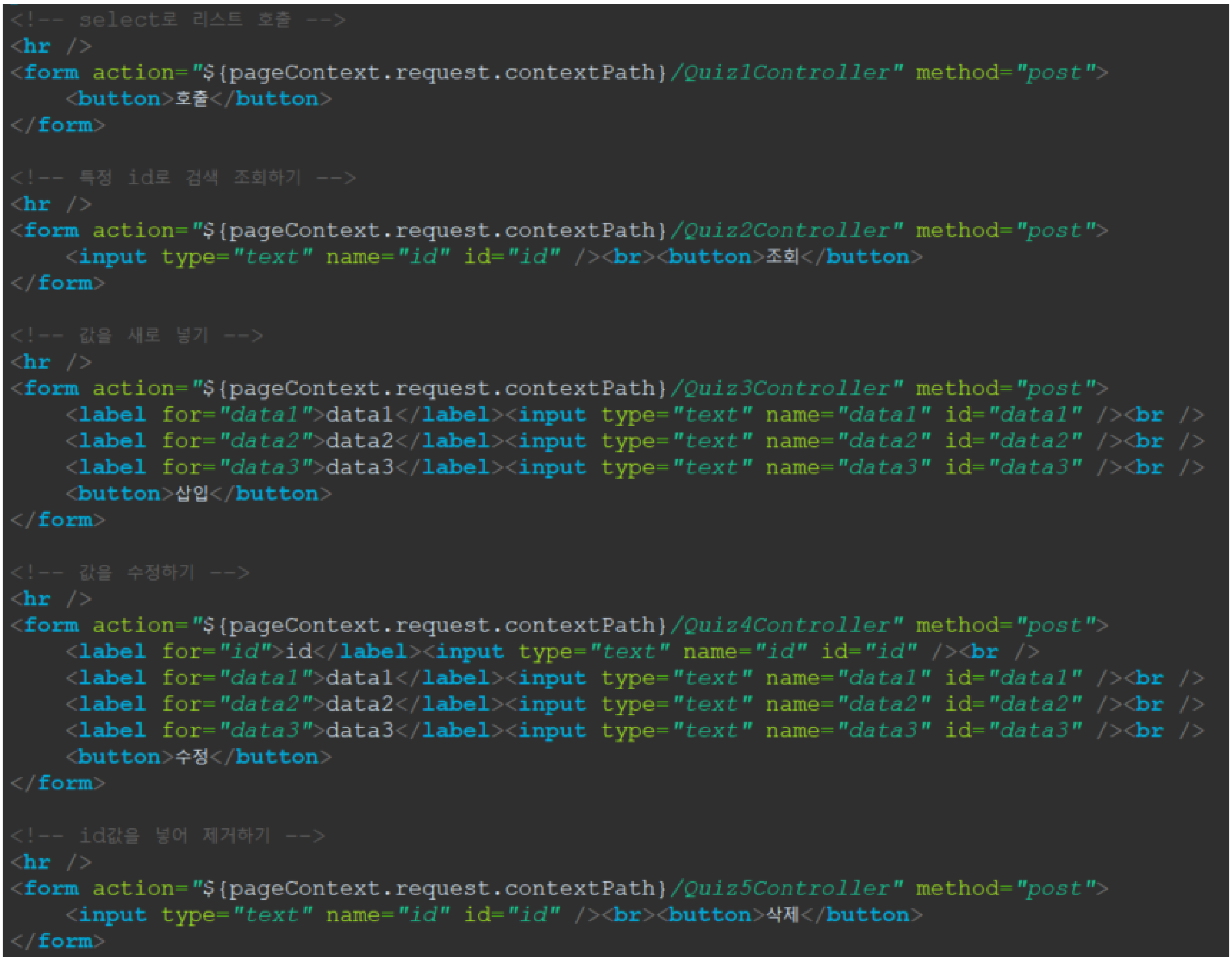
```
create table quiz3.sampletable
(
    id      bigint auto_increment comment 'id'
        primary key,
    data1 varchar(255) null comment 'data1',
    data2 varchar(255) null comment 'data2',
    data3 varchar(255) null comment 'data3'
)
    comment '샘플테이블';
insert into quiz3.sampletable (id, data1, data2, data3)
values  (1, 'aaa1', 'aaa2', 'aaa3'),
        (2, 'bbb1', 'bbb2', 'bbb3'),
        (3, 'ccc1', 'ccc2', 'ccc3');
```

<페이지 구조>



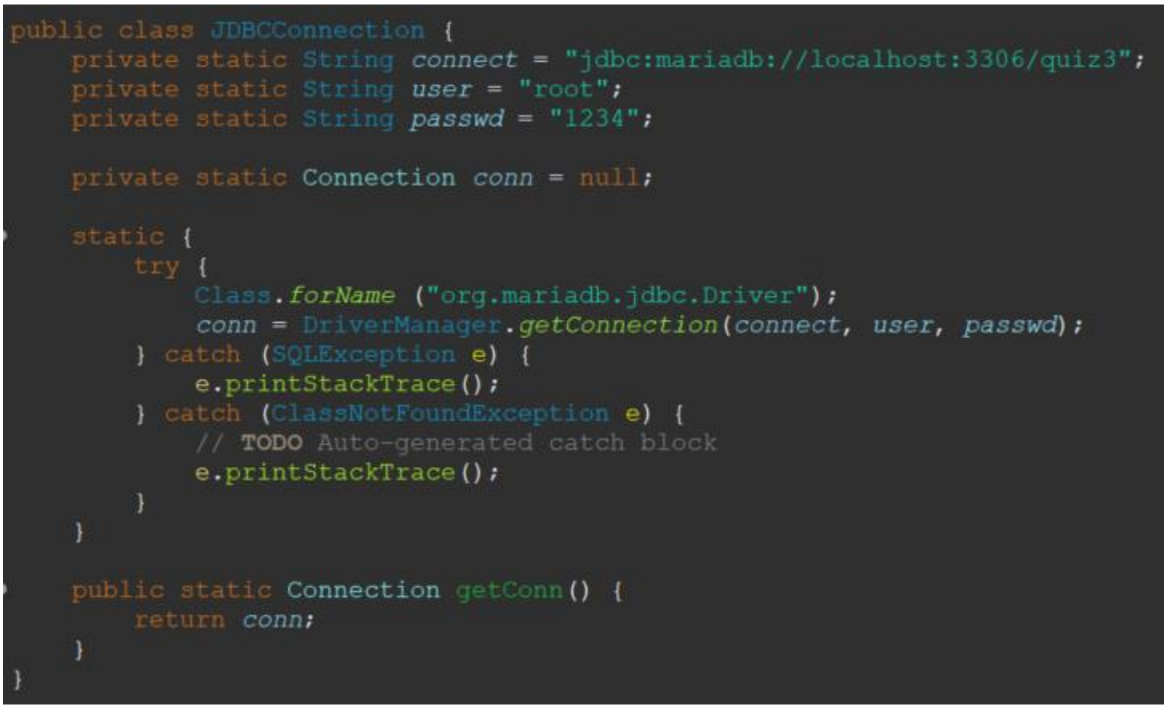
```
src/main/java
  com.sample.conn
    JDBCConnection.java
  com.sample.controller
    Quiz1Controller.java
    Quiz2Controller.java
    Quiz3Controller.java
    Quiz4Controller.java
    Quiz5Controller.java
  com.sample.dao
    DataDAO.java
  com.sample.dto
    DataDTO.java
  com.sample.service
    JDBCService.java
Server Runtime [Apache Tomcat v9.0]
Web App Libraries
build
src
  main
    java
    webapp
      META-INF
      result1
        result1.jsp
        result2.jsp
        result3.jsp
        result4.jsp
        result5.jsp
      test
        test.jsp
      WEB-INF
        lib
          mariadb-java-client-2.7.3.jar
          taglibs-standard-compat-1.2.5.jar
          taglibs-standard-impl-1.2.5.jar
          taglibs-standard-jstlel-1.2.5.jar
          taglibs-standard-spec-1.2.5.jar
```

Test1.jsp



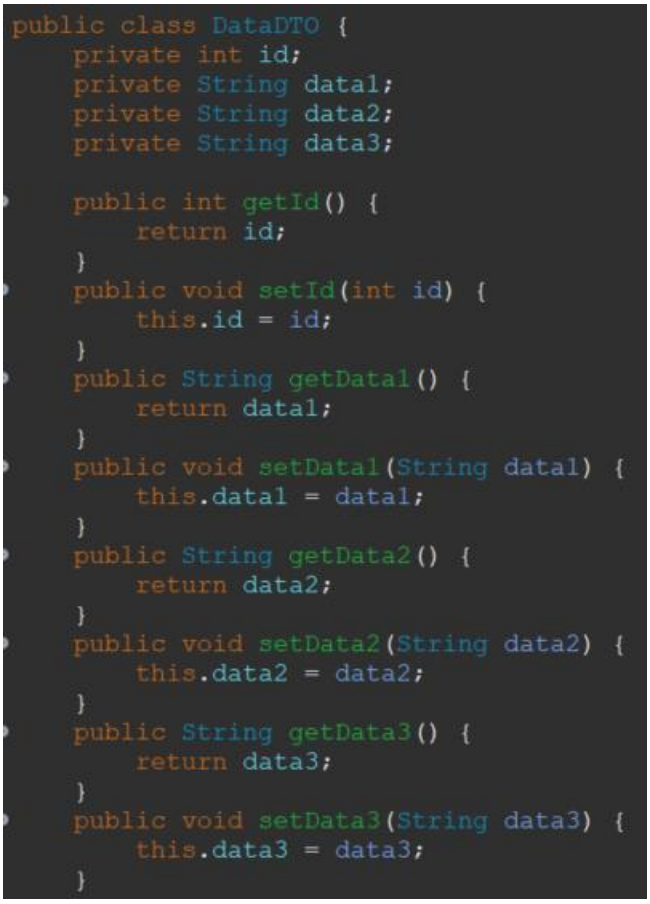
```
<!-- select로 리스트 호출 -->
<hr />
<form action="${pageContext.request.contextPath}/Quiz1Controller" method="post">
    <button>호출</button>
</form>

<!-- 특정 id로 검색 조회하기 -->
<hr />
<form action="${pageContext.request.contextPath}/Quiz2Controller" method="post">
    <input type="text" name="id" id="id" /><br><button>조회</button>
</form>

<!-- 값을 새로 넣기 -->
<hr />
<form action="${pageContext.request.contextPath}/Quiz3Controller" method="post">
    <label for="data1">data1</label><input type="text" name="data1" id="data1" /><br />
    <label for="data2">data2</label><input type="text" name="data2" id="data2" /><br />
    <label for="data3">data3</label><input type="text" name="data3" id="data3" /><br />
    <button>삽입</button>
</form>

<!-- 값을 수정하기 -->
<hr />
<form action="${pageContext.request.contextPath}/Quiz4Controller" method="post">
    <label for="id">id</label><input type="text" name="id" id="id" /><br />
    <label for="data1">data1</label><input type="text" name="data1" id="data1" /><br />
    <label for="data2">data2</label><input type="text" name="data2" id="data2" /><br />
    <label for="data3">data3</label><input type="text" name="data3" id="data3" /><br />
    <button>수정</button>
</form>

<!-- id값을 넣어 제거하기 -->
<hr />
<form action="${pageContext.request.contextPath}/Quiz5Controller" method="post">
    <input type="text" name="id" id="id" /><br><button>삭제</button>
</form>
```

JDBCConnection.java



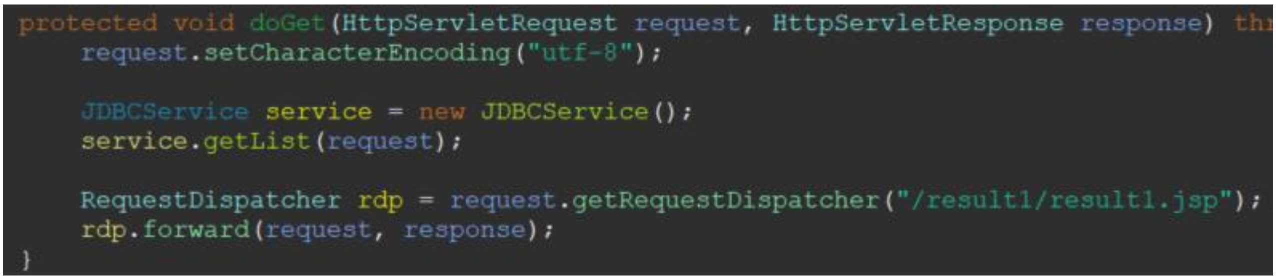
```
public class JDBCConnection {
    private static String connect = "jdbc:mariadb://localhost:3306/quiz3";
    private static String user = "root";
    private static String passwd = "1234";

    private static Connection conn = null;

    static {
        try {
            Class.forName ("org.mariadb.jdbc.Driver");
            conn = DriverManager.getConnection(connect, user, passwd);
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static Connection getConn() {
        return conn;
    }
}
```

DataDTO.java



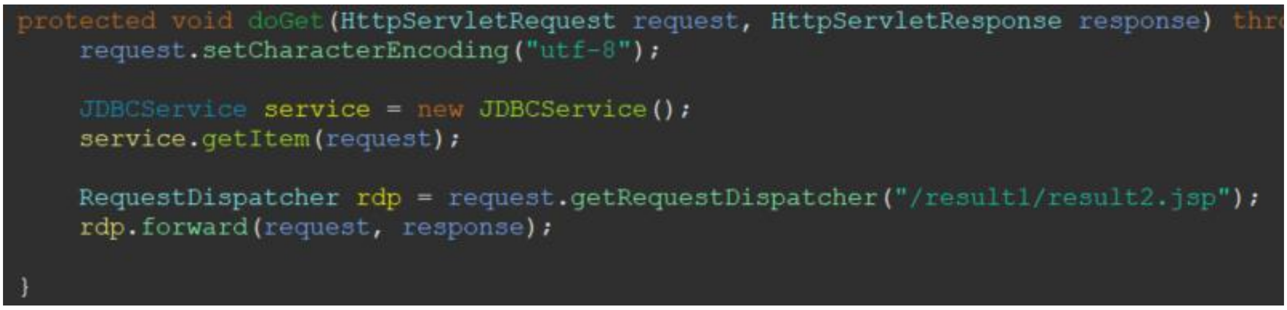
```java
public class DataDTO {
    private int id;
    private String data1;
    private String data2;
    private String data3;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getData1() {
        return data1;
    }
    public void setData1(String data1) {
        this.data1 = data1;
    }
    public String getData2() {
        return data2;
    }
    public void setData2(String data2) {
        this.data2 = data2;
    }
    public String getData3() {
        return data3;
    }
    public void setData3(String data3) {
        this.data3 = data3;
    }
```

Quiz1Controller.java



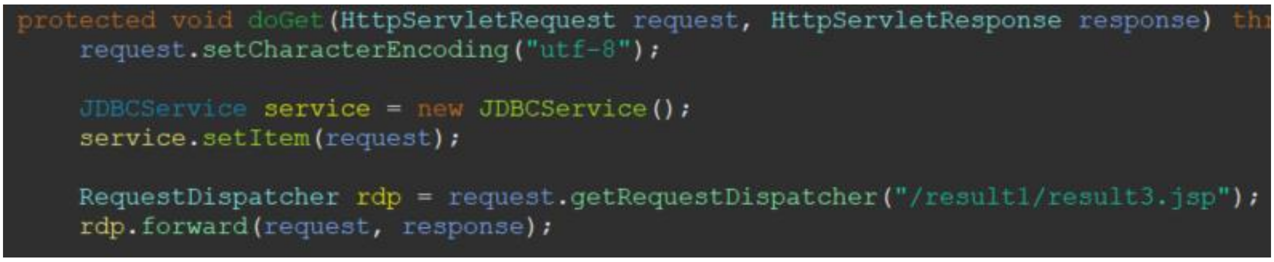
```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) th
    request.setCharacterEncoding("utf-8");

    JDBCService service = new JDBCService();
    service.getList(request);

    RequestDispatcher rdp = request.getRequestDispatcher("/result1/result1.jsp");
    rdp.forward(request, response);
}
```

Quiz2Controller.java



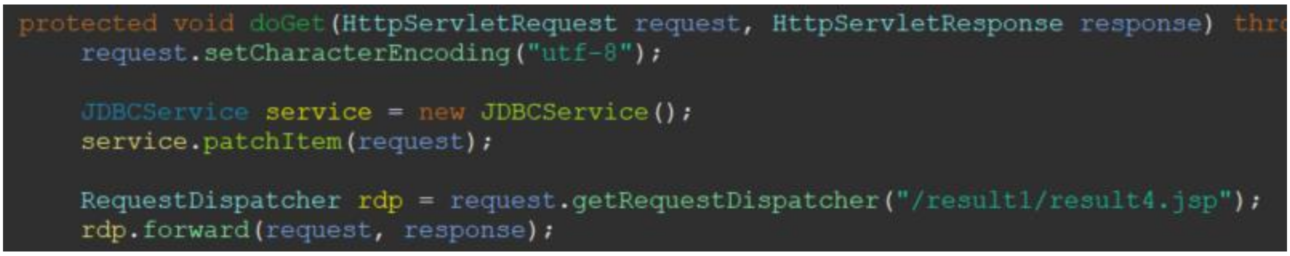
```java
protected void doGet(HttpServletRequest request, HttpServletResponse response) thr
    request.setCharacterEncoding("utf-8");

    JDBCService service = new JDBCService();
    service.getItem(request);

    RequestDispatcher rdp = request.getRequestDispatcher("/result1/result2.jsp");
    rdp.forward(request, response);

}
```

Quiz3Controller.java



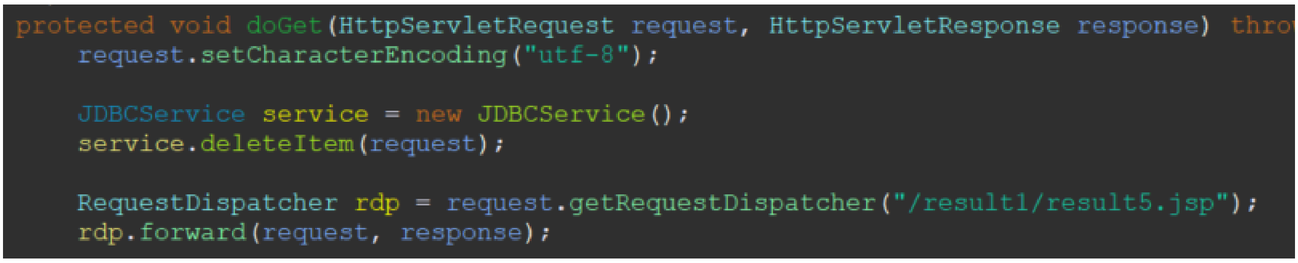
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) thr
    request.setCharacterEncoding("utf-8");

    JDBCService service = new JDBCService();
    service.setItem(request);

    RequestDispatcher rdp = request.getRequestDispatcher("/result1/result3.jsp");
    rdp.forward(request, response);
```

Quiz4Controller.java



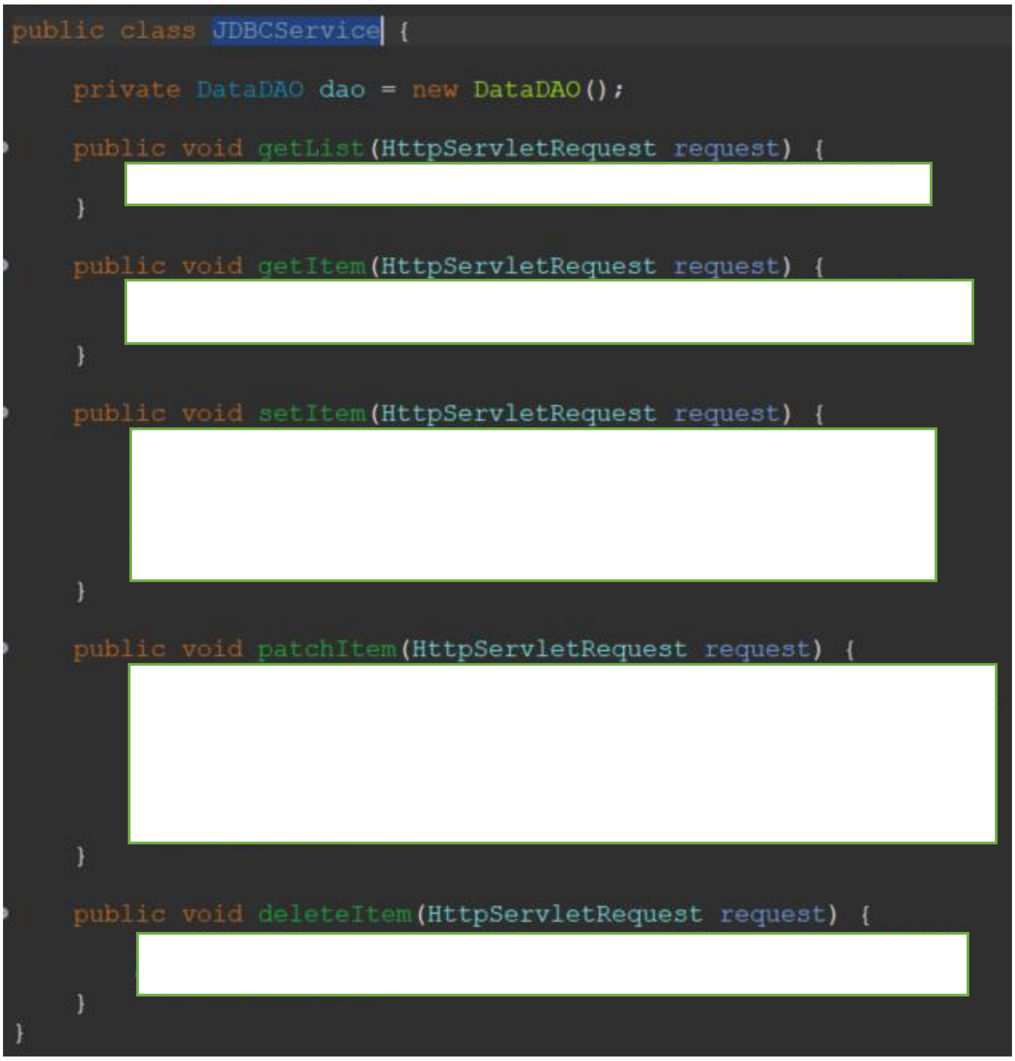
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) thr
    request.setCharacterEncoding("utf-8");

    JDBCService service = new JDBCService();
    service.patchItem(request);

    RequestDispatcher rdp = request.getRequestDispatcher("/result1/result4.jsp");
    rdp.forward(request, response);
```

Quiz5Controller.java



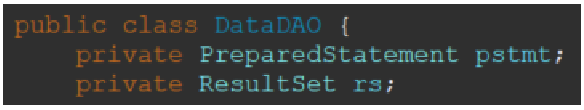
```
protected void doGet(HttpServletRequest request, HttpServletResponse response) thro
    request.setCharacterEncoding("utf-8");

    JDBCService service = new JDBCService();
    service.deleteItem(request);

    RequestDispatcher rdp = request.getRequestDispatcher("/result1/result5.jsp");
    rdp.forward(request, response);
```

JDBCService.java



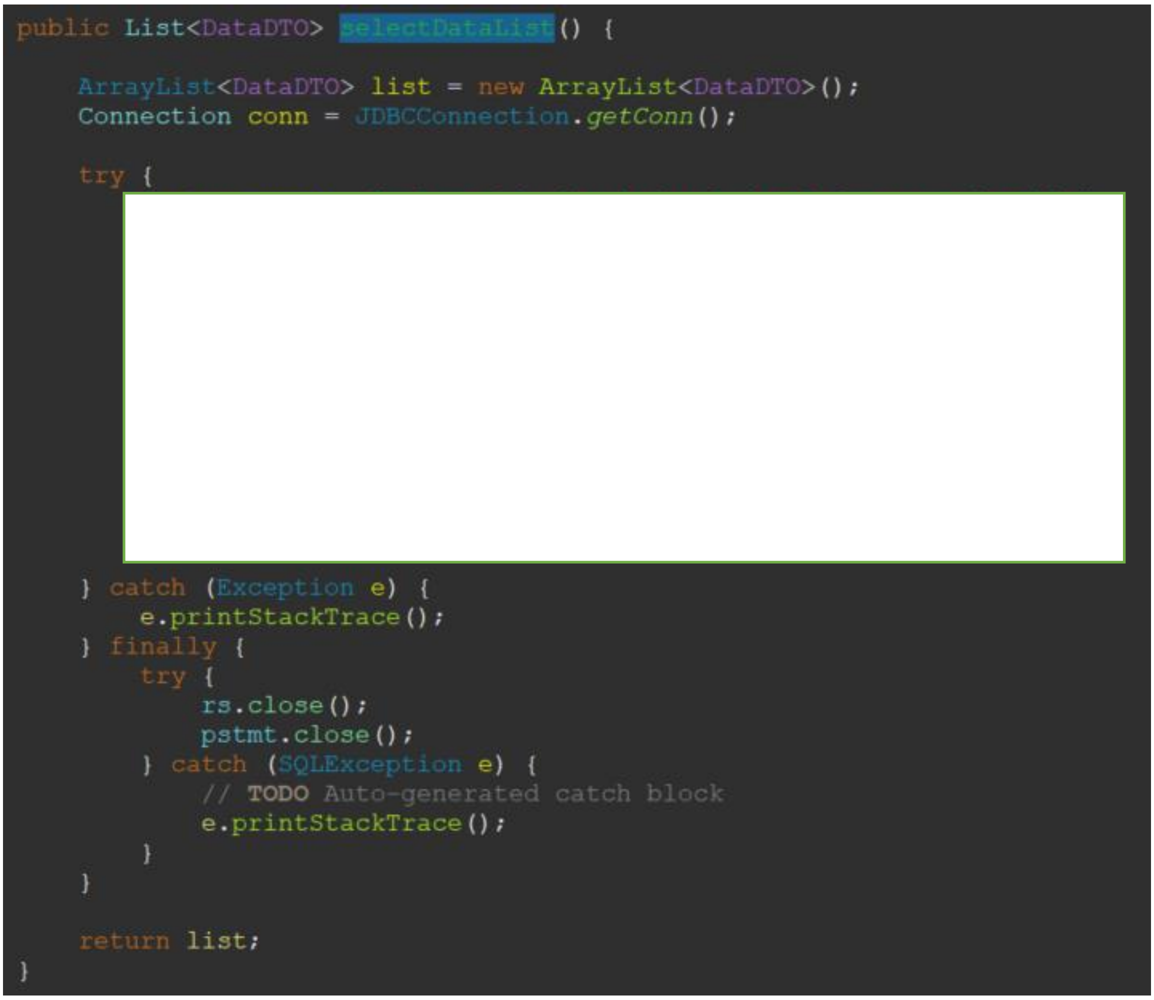
```java
public class JDBCService {

    private DataDAO dao = new DataDAO();

    public void getList(HttpServletRequest request) {

    }

    public void getItem(HttpServletRequest request) {

    }

    public void setItem(HttpServletRequest request) {

    }

    public void patchItem(HttpServletRequest request) {

    }

    public void deleteItem(HttpServletRequest request) {

    }
}
```

DataDAO.java



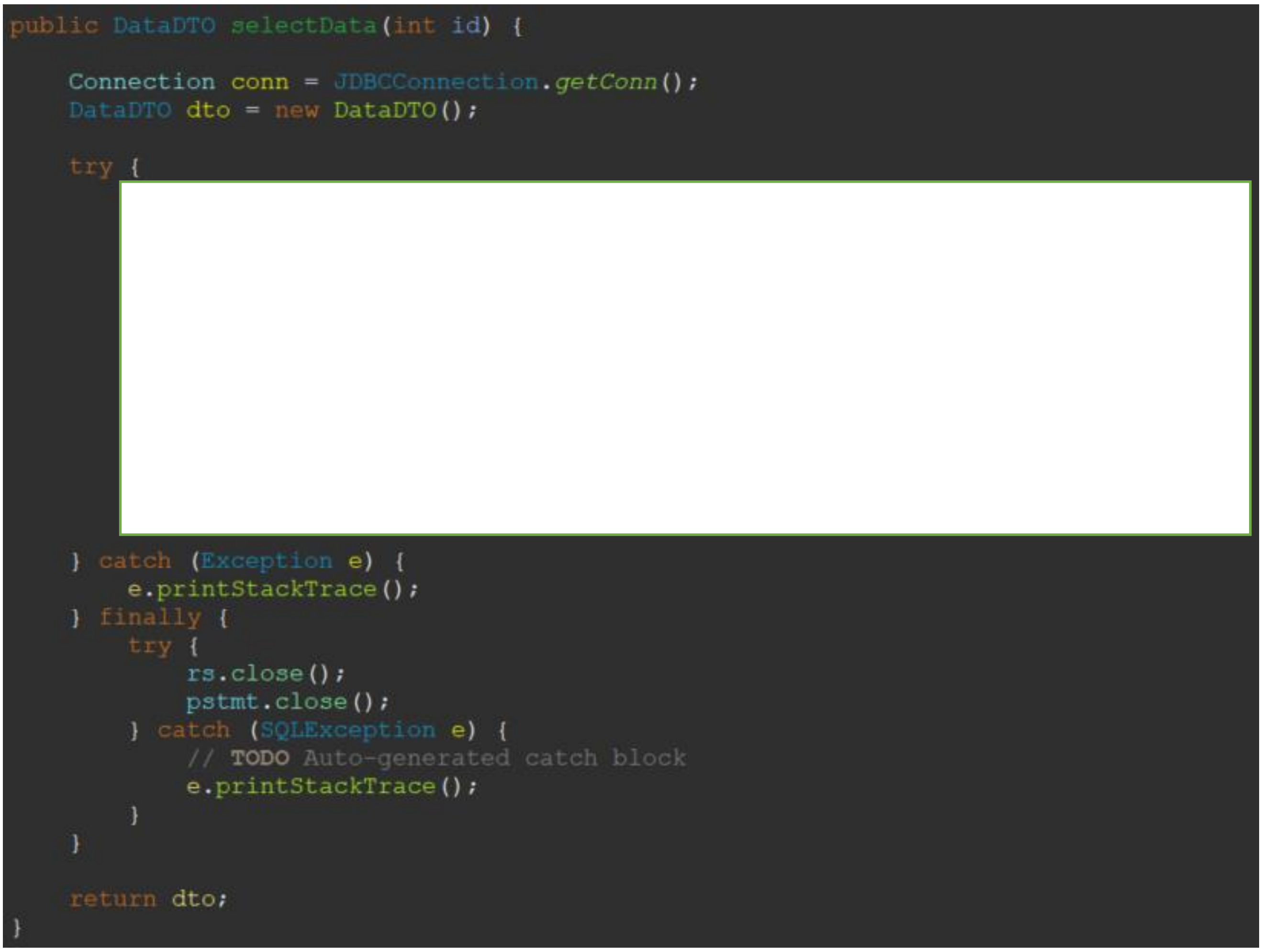
```java
public class DataDAO {
    private PreparedStatement pstmt;
    private ResultSet rs;
```

DataDAO.java - selectDataList



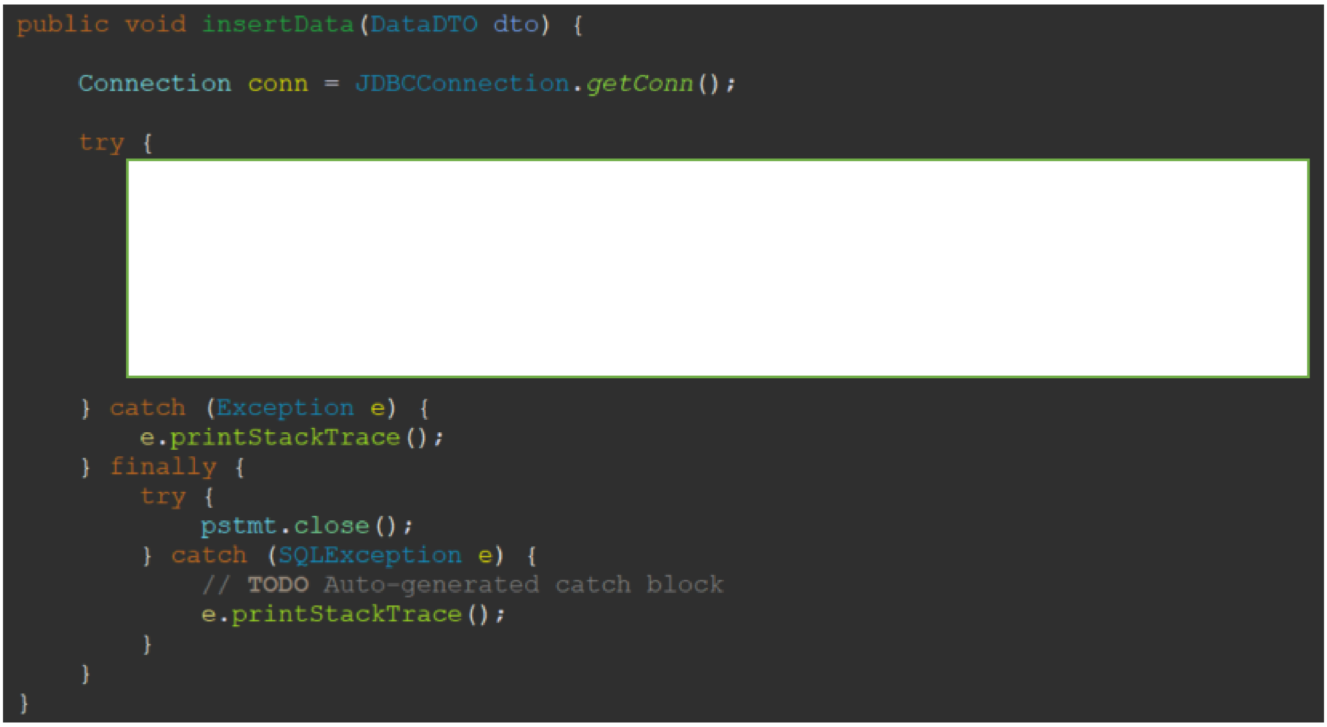
```java
public List<DataDTO> selectDataList() {

    ArrayList<DataDTO> list = new ArrayList<DataDTO>();
    Connection conn = JDBCConnection.getConn();

    try {

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            rs.close();
            pstmt.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    return list;
}
```

DataDAO.java - selectData



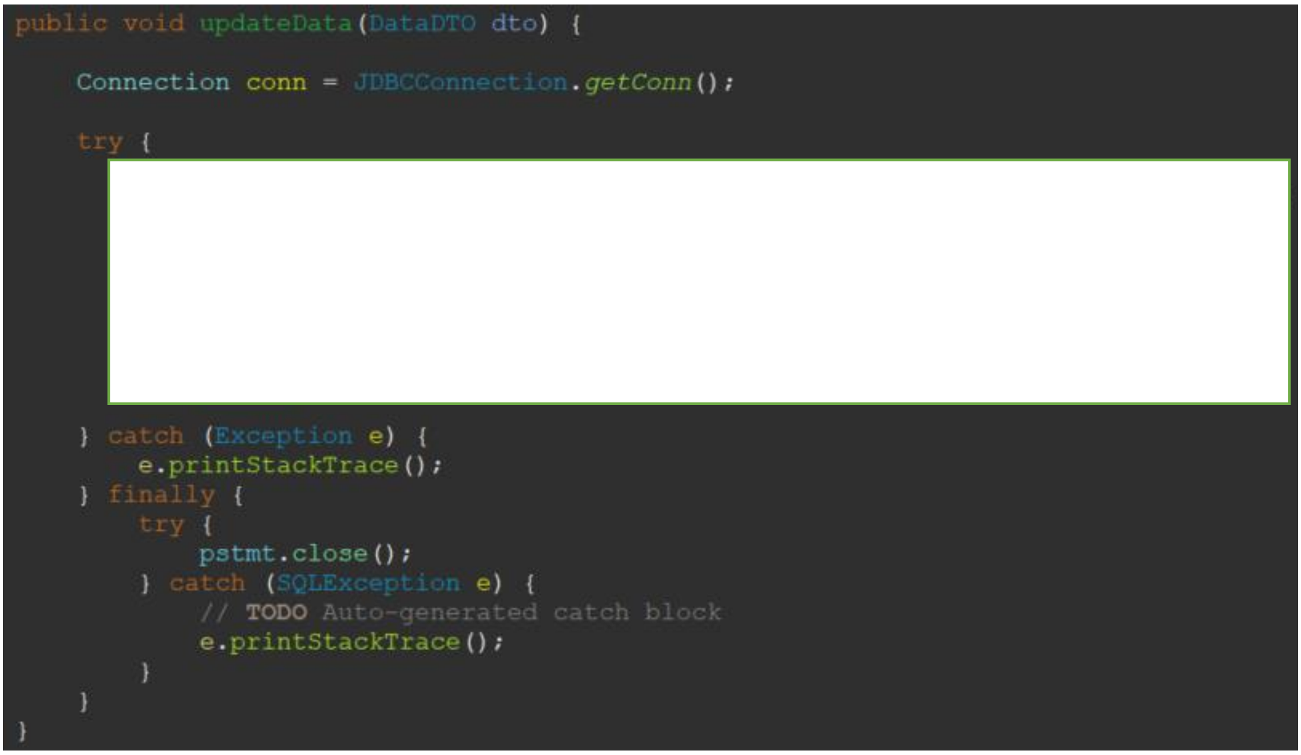
```
public DataDTO selectData(int id) {

    Connection conn = JDBCConnection.getConn();
    DataDTO dto = new DataDTO();

    try {









    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            rs.close();
            pstmt.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    return dto;
}
```

DataDAO.java – insertData



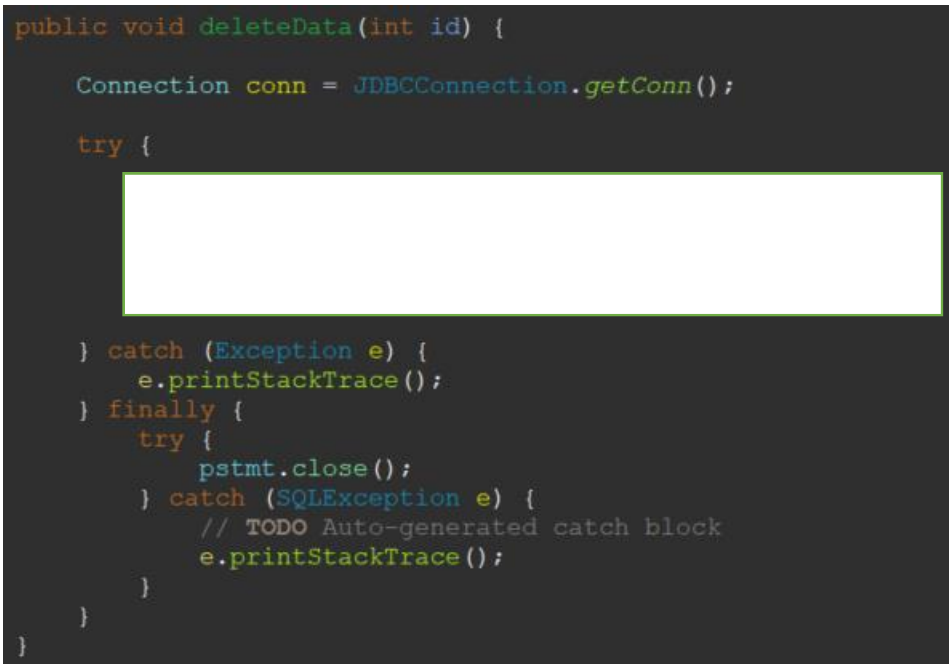
```
public void insertData(DataDTO dto) {

    Connection conn = JDBCConnection.getConn();

    try {







    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            pstmt.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

DataDAO.java – updateData



```
public void updateData(DataDTO dto) {

    Connection conn = JDBCConnection.getConn();

    try {







    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            pstmt.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

DataDAO.java – deleteData



```
public void deleteData(int id) {

    Connection conn = JDBCConnection.getConn();

    try {




    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            pstmt.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

Result1.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

Result2.jsp

Result3.jsp

```
<h1>저장되었습니다</h1>
```

Result4.jsp

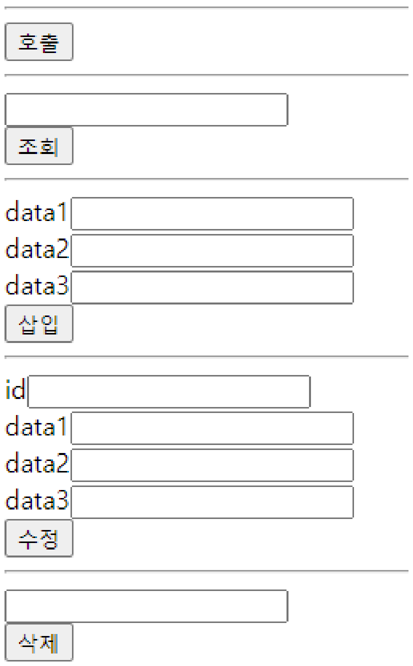
```
<h1>수정되었습니다</h1>
```

Result5.jsp

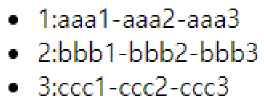
```
<h1>삭제되었습니다</h1>
```

<결과창>

Test.jsp

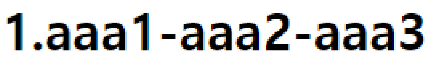


Result1.jsp



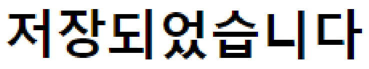
- 1:aaa1-aaa2-aaa3
- 2:bbb1-bbb2-bbb3
- 3:ccc1-ccc2-ccc3

Result2.jsp



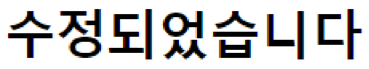
1.aaa1-aaa2-aaa3

Result3.jsp



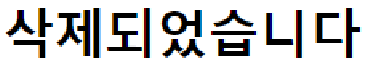
저장되었습니다

Result4.jsp



수정되었습니다

Result5.jsp



삭제되었습니다

문제2. 위의 문제를 MyBatis를 사용하여 풀어볼 것