

## Chapter 12

# 회원 관리 실습

[ 모델2와 JSP 빈즈를 이용]

## 회원 관리 개요

## ■ 회원 관리의 개요

- MVC 패턴을 이용한 개발

- ▶ 데이터베이스에 접근하여 비즈니스 로직을 수행하는 모델(Model)
  - 자바 클래스로 구성
  - 클래스 내부에는 비즈니스 로직을 수행하는 메서드들을 포함
- ▶ 요구되는 비즈니스 로직의 수행 결과를 브라우저에 출력하는 뷰(View)
  - JSP 문서로 개발
- ▶ 클라이언트의 요구에 따라 모델의 비즈니스 로직과 뷰 문서를 수행하도록하는 컨트롤러(Controller)
  - 모든 사용자의 요구는 컨트롤러를 통해 실행됨
  - 일반적으로 서블릿으로 개발
  - 본 실습에서는 빈즈를 사용하기 위해 JSP로 개발할 것임

- 빈즈를 이용한 데이터의 처리 수행

- ▶ 빈즈를 사용해 회원 정보를 문서들 사이에 공유
- ▶ DAO와 VO 클래스를 빈즈로 생성해 처리

# 회원 관리 개요

## ■ 관련 파일

구분	기능	설명	비고
View	memberInsertForm.jsp	회원 정보 입력 (이름, 전자우편, 전화번호, 부서명, 입사일, 직위)	이전 작성 문서 사용
	memberList.jsp	회원 목록 출력 (정보 수정과 삭제를 위한 링크를 포함)	이전 작성 문서 사용
	memberModifyForm.jsp	수정을 위한 회원 정보 입력 (이름을 제외한 정보 수정 가능)	이전 작성 문서 사용
	style.css	스타일 시트 문서	이전 작성 문서 사용
	index.htm	초기 페이지	
Model	MemberDAO.java	데이터베이스 등록/수정/삭제/ 조회 등을 담당하는 클래스	
	MemberVO.java	회원 정보 데이터 객체	
Controller	MemberController.jsp	컨트롤러 (제어의 이동, 뷰 실행, 데이터 객체 전달)	

## ■ 데이터베이스와 테이블

- 이전에 작성한 데이터베이스와 테이블을 사용

- ▶ 데이터베이스 : jspdb

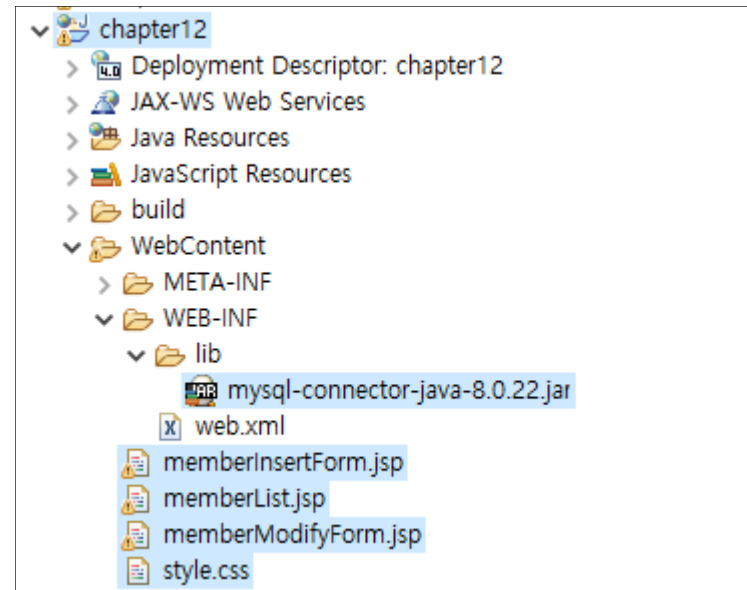
- ▶ 테이블 : members

컬럼명	자료형	기능	비고
memberID	int(5)	회원 레코드 번호	not null, primary key, auto_increment
memberName	varchar(10)	회원 이름	not null
memberMail	varchar(60)	회원 이메일 주소	not null
memberTel	varchar(20)	회원 연락처 (xxx-xxxx-xxxx)	not null
memberComp	varchar(20)	회원 회사명	
memberBirth	varchar(10)	회원 생년월일 (yyyy-mm-dd)	not null
memberPos	varchar(10)	회원 직위	

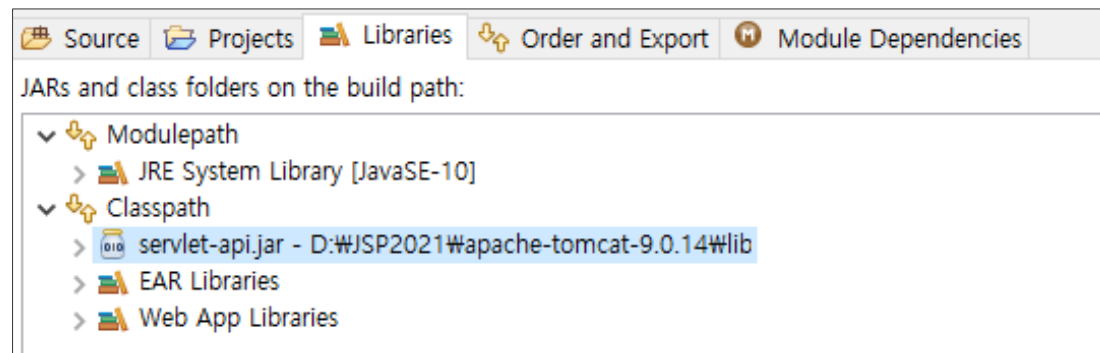
## ■ 프로젝트 생성

- 프로젝트 이름 : chapter12

- ▶ mysql connector 복사
- ▶ chapter11에서 작성한 4개의 파일을 chapter12로 사용
  - 일부 수정 필요



- servlet-api 라이브러리 추가



클래스 생성

# memberVO 클래스 생성

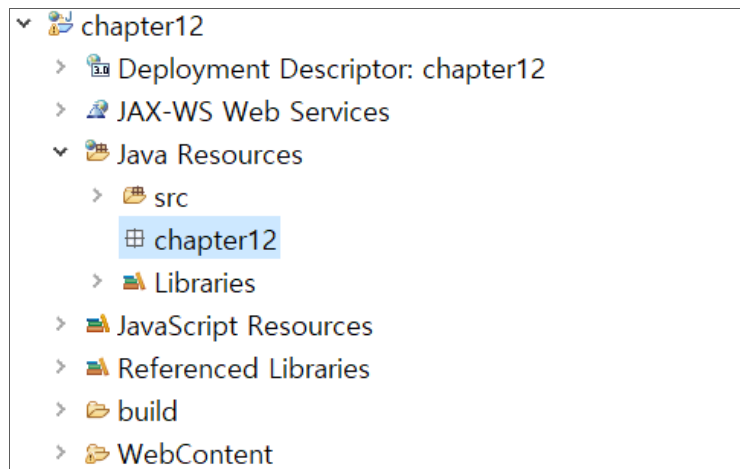
## ■ 패키지과 클래스 생성

- 패키지 이름

  - ▶ chapter12

- 클래스 이름

  - ▶ MemberVO.java



The 'New Java Class' dialog box is shown with the following configuration:

- Source folder: chapter12/src
- Package: chapter12
- Enclosing type: (empty)
- Name: MemberVO
- Modifiers: ☒ public, ☐ package, ☐ private, ☐ protected, ☐ abstract, ☐ final, ☐ static
- Superclass: java.lang.Object
- Interfaces: (empty)
- Which method stubs would you like to create?
  - ☐ public static void main(String[] args)
  - ☐ Constructors from superclass
  - ☒ Inherited abstract methods
- Do you want to add comments? (Configure templates and default value [here](#))
  - ☐ Generate comments

Buttons: Finish, Cancel



```
1 package chapter12;
2
3 public class MemberVO {
4
5     //멤버 변수 선언
6     private int memberID;
7     private String memberName;
8     private String memberMail;
9     private String memberTel;
10    private String memberComp;
11    private String memberBirth;
12    private String memberPos;
13
14    // memberID의 Getter와 Setter
15    public int getMemberID() {
16        return memberID;
17    }
18    public void setMemberID(int memberID) {
19        this.memberID = memberID;
20    }
21
22    // memberName의 Getter와 Setter
23    public String getMemberName() {
24        return memberName;
25    }
26    public void setMemberName(String memberName) {
27        this.memberName = memberName;
28    }
```

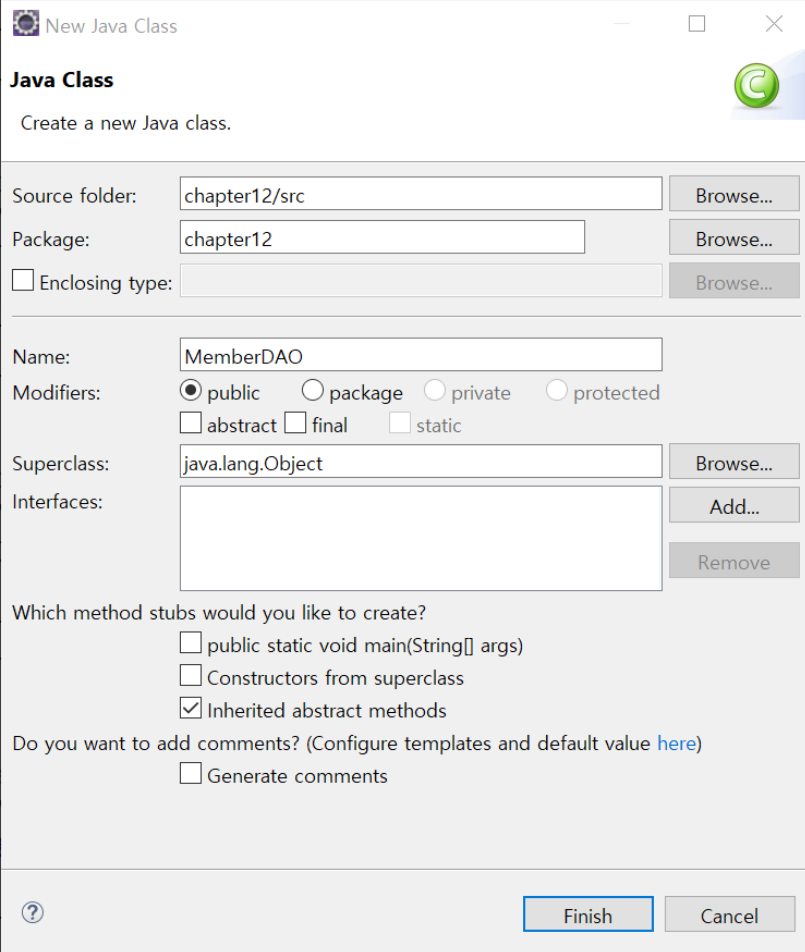
```
29
30 // memberMail의 Getter와 Setter
31 public String getMemberMail() {
32     return memberMail;
33 }
34 public void setMemberMail(String memberMail) {
35     this.memberMail = memberMail;
36 }
37
38 // memberTel의 Getter와 Setter
39 public String getMemberTel() {
40     return memberTel;
41 }
42 public void setMemberTel(String memberTel) {
43     this.memberTel = memberTel;
44 }
45
46 // memberComp의 Getter와 Setter
47 public String getMemberComp() {
48     return memberComp;
49 }
50 public void setMemberComp(String memberComp) {
51     this.memberComp = memberComp;
52 }
53
```

```
54 // memberBirth의 Getter와 Setter
55 public String getMemberBirth() {
56     return memberBirth;
57 }
58 public void setMemberBirth(String memberBirth) {
59     this.memberBirth = memberBirth;
60 }
61
62 // memberPos의 Getter와 Setter
63 public String getMemberPos() {
64     return memberPos;
65 }
66 public void setMemberPos(String memberPos) {
67     this.memberPos = memberPos;
68 }
69
70 }
```

# memberDAO 클래스 생성

## ■ memberDAO 클래스 생성

- 데이터베이스에 대한 연산을 담당하는 클래스
  - ▶ 데이터베이스 연결과 해제
  - ▶ 회원 정보 입력/수정/삭제/출력을 수행하는 메서드로 구성
- 데이터베이스 연결과 해제를 위한 메서드를 먼저 생성
  - ▶ connect() 메서드
  - ▶ disconnect() 메서드
  - ▶ 차후 다른 기능을 수행하는 메소드를 추가할 것임



The image shows the 'New Java Class' dialog box in an IDE. The title bar says 'New Java Class'. Below the title bar, it says 'Java Class' and 'Create a new Java class.' The dialog has several fields and buttons:

- Source folder:** chapter12/src (with a 'Browse...' button)
- Package:** chapter12 (with a 'Browse...' button)
- Enclosing type:** (empty, with a 'Browse...' button)
- Name:** MemberDAO
- Modifiers:** ☒ public, ☐ package, ☐ private, ☐ protected, ☐ abstract, ☐ final, ☐ static
- Superclass:** java.lang.Object (with a 'Browse...' button)
- Interfaces:** (empty list, with 'Add...' and 'Remove' buttons)
- Which method stubs would you like to create?**
  - ☐ public static void main(String[] args)
  - ☐ Constructors from superclass
  - ☒ Inherited abstract methods
- Do you want to add comments? (Configure templates and default value [here](#))**
  - ☐ Generate comments

At the bottom right, there are 'Finish' and 'Cancel' buttons. A help icon (?) is at the bottom left.

```
1 package chapter12;
2
3 import java.sql.*;
4
5 public class MemberDAO {
6
7
8     // 데이터베이스 접속 관련 객체 선언 -----
9     Connection conn = null;
10    PreparedStatement pstmt = null;
11    ResultSet rs = null;
12
13    // 데이터베이스 접속 관련 변수 선언 -----
14    String jdbc_driver = "com.mysql.cj.jdbc.Driver";
15    String jdbc_url = "jdbc:mysql://localhost:3306/jspdb?serverTimezone=UTC";
16
17    void connect( ) {
18
19        try {
20            //JDBC 드라이버 로드와 Connection 객체 생성 -----
21            Class.forName(jdbc_driver);
22            conn = DriverManager.getConnection(jdbc_url, "jspstudy", "jsppass");
23        } catch(Exception e) {
24            e.printStackTrace();
25        }
26
27    }
```

```
29 void disconnect() {
30
31     //ResultSet 객체 종료 -----
32     if(rs != null) {
33         try {
34             rs.close();
35         } catch(Exception e) {
36             e.printStackTrace();
37         }
38     }
39     //PreparedStatement 객체 종료 -----
40     if(pstmt != null) {
41         try {
42             pstmt.close();
43         } catch(Exception e) {
44             e.printStackTrace();
45         }
46     }
47     //Connection 객체 종료 -----
48     if(conn != null) {
49         try {
50             conn.close();
51         } catch(Exception e) {
52             e.printStackTrace();
53         }
54     }
55 }
56
57 }
```

## 회원 목록 출력

# 컨트롤러의 역할

## ■ 컨트롤러의 역할

- 컨트롤러는 뷰(view)와 모델(model) 사이의 인터페이스 역할을 수행

- ▶ 모든 요청은 컨트롤러를 통해 수행됨

- 모든 요청은 액션 코드(action code)를 컨트롤러에 전달하도록 함
    - 액션 코드를 통해 컨트롤러는 수행해야 하는 내용을 결정

- ▶ 컨트롤러는 요청된 액션 코드를 분석해 메서드를 수행하거나, 뷰 문서를 실행해 브라우저에 출력

- 실습에서 사용할 액션 코드

액션 코드	설명
list	데이터베이스로부터 회원 정보를 추출해 브라우저에 출력
insert	입력 문서를 통해 전달된 회원 정보를 데이터베이스에 저장
modify	수정할 회원 정보를 데이터베이스에서 추출
update	데이터베이스에 저장된 기존 레코드를 수정
delete	데이터베이스에 저장된 기존 레코드를 제거

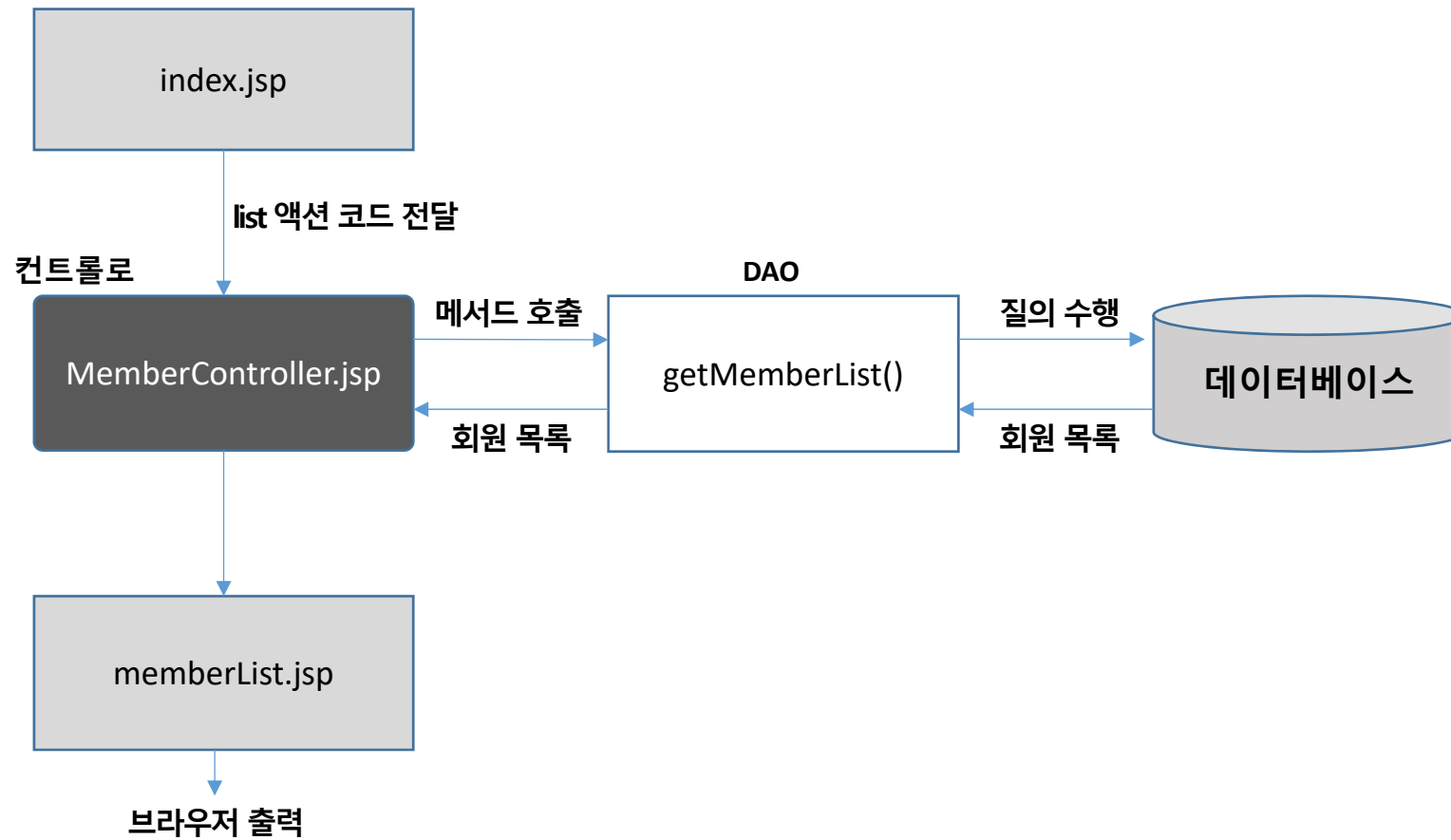


# 회원 목록 출력의 개요

## ■ 회원 목록 출력의 동작

- 회원 목록을 출력하는 문서는 memberList.jsp임
  - ▶ 하지만, 브라우저를 통해 바로 memberList.jsp를 요청할 수 없음
  - ▶ 모든 요청은 컨트롤러를 통해 이루어지며, 이는 memberList.jsp 문서도 마찬가지임
- memberList.jsp 문서를 로드하는 방법
  - ① list 액션 코드를 컨트롤러에 전달
  - ② 컨트롤러는 회원 목록을 추출하기 위해 MemberDAO의 getMemberList( )메서드를 호출함
  - ③ getMemberList( )메서드 데이터베이스로부터 회원 목록을 추출해 컨트롤러에 전달
  - ④ 컨트롤러는 회원 목록을 memberList.jsp에 전달해 브라우저에 출력함
- 컨트롤러에 list 액션 코드를 전달하는 방법
  - ▶ 컨트롤러에 list 액션 코드만을 전달하는 별도의 문서를 작성
  - ▶ 이 문서를 실행해 컨트롤러에 list 액션코 드를 전달

## 회원 목록 출력의 개요



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"  
2     pageEncoding="UTF-8"%>  
3  
4 <% response.sendRedirect("MemberController.jsp?action=list"); %>
```

### ■ MemberController.jsp 문서 생성

- MemberDAO 클래스의 빈즈 생성

- ▶ 생성된 빈즈에 접근해 MemberDAO 클래스에 선언된 메서드를 호출해 수행
- ▶ 기능이 추가될 때마다 MemberDAO 클래스의 메서드가 추가될 예정임

- 액션 코드 추출 및 getMemberList() 메서드 호출

- ▶ index.jsp 문서에서 전달받은 list 액션 코드를 추출
  - list 액션 코드는 데이터베이스에서 회원 목록을 추출하는 getMemberList() 메서드를 호출하는데 사용
- ▶ 데이터베이스에는 하나 이상의 회원 정보가 존재할 수 있음
  - 각각의 회원 정보를 MemberVO 클래스의 빈즈로 생성
  - 생성된 빈즈들을 배열 객체에 저장해 컨트롤러에 전달
- ▶ 컨트롤러는 전달받은 배열을 memberList.jsp로 전달해 브라우저에 출력할 수 있도록 함
  - 배열을 request 영역의 속성으로 지정해 memberList.jsp로 전달함

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <%@ page import="java.util.*" %>
5 <%@ page import="chapter12.*" %>
6
7 <!-- MemberDAO 클래스의 빈즈 생성 ----- -->
8 <jsp:useBean id="memberDao" class="chapter12.MemberDAO" />
9
10 <%
11
12 // memberInsertForm.jsp에서 전달된 액션 코드 추출 -----
13   String action = request.getParameter("action");
14
15 //회원의 목록을 출력하기 위한 getMemberList() 메서드 호출
16 //전달받은 회원 정보 객체 배열을 memberList.jsp에 전달해 브라우저에 출력
17   if( action.equals("list") ) {
18
19       ArrayList<MemberVO> memberlist = memberDao.getMemberList();
20       request.setAttribute("memberlist", memberlist);
21       pageContext.forward("memberList.jsp");
22
23   }
24
25 %>
```

## 회원 목록 출력 ( MemberDAO.java )

### ■ MemberDAO.java 수정

- `getMemberList()` 메서드 구현
  - ▶ 데이터베이스에는 하나 회원 정보가 존재할 수 있음
  - ▶ 각각의 회원 정보를 MemberVO 객체에 저장
  - ▶ 회원 정보들이 저장된 MemberVO 객체들을 배열 객체에 저장하여 컨트롤러에 전달

```
1 package chapter12;
2
3 import java.sql.*;
4 import java.util.*;
5
6 public class MemberDAO {
7
8     // 데이터베이스 접속 관련 객체 선언 -----
9     Connection conn = null;
10    PreparedStatement pstmt = null;
11    ResultSet rs = null;
12
13    // 데이터베이스 접속 관련 변수 선언 -----
14    String jdbc_driver = "com.mysql.cj.jdbc.Driver";
15    String jdbc_url = "jdbc:mysql://localhost:3306/jspdb?serverTimezone=UTC";
16
17    // 데이터베이스 연결 -----
18    void connect( ) {
19
20        try {
21            //JDBC 드라이버 로드와 Connection 객체 생성
22            Class.forName(jdbc_driver);
23            conn = DriverManager.getConnection(jdbc_url, "jspstudy", "jsppass");
24        } catch(Exception e) {
25            e.printStackTrace();
26        }
27
28    }
```

```
30 // 데이터베이스 연결 해제 -----
31 void disconnect() {
32
33     //ResultSet 객체 종료
34     if(rs != null) {
35         try {
36             rs.close();
37         } catch(Exception e) {
38             e.printStackTrace();
39         }
40     }
41     //PreparedStatement 객체 종료
42     if(pstmt != null) {
43         try {
44             pstmt.close();
45         } catch(Exception e) {
46             e.printStackTrace();
47         }
48     }
49     //Connection 객체 종료
50     if(conn != null) {
51         try {
52             conn.close();
53         } catch(Exception e) {
54             e.printStackTrace();
55         }
56     }
57 }
```



```
59 // 전체 회원 목록을 출력 ( 회원 정보들을 각각 객체로 생성해 배열로 저장한 다음 전달 )
60 public ArrayList<MemberVO> getMemberList() {
61
62     connect();
63
64     ArrayList<MemberVO> lists = new ArrayList<MemberVO>();
65
66     try {
67
68         String Query = "select * from members order by memberID desc ";
69         pstmt = conn.prepareStatement(Query);
70         rs = pstmt.executeQuery();
71
72         while( rs.next() ) {
73
74             MemberVO member = new MemberVO();
75
76             member.setMemberID(rs.getInt("memberID"));
77             member.setMemberName(rs.getString("memberName"));
78             member.setMemberMail(rs.getString("memberMail"));
79             member.setMemberTel(rs.getString("memberTel"));
80             member.setMemberComp(rs.getString("memberComp"));
81             member.setMemberBirth(rs.getString("memberBirth"));
82             member.setMemberPos(rs.getString("memberPos"));
83
84             lists.add(member);
85
86         }
```

```
87
88         } catch (Exception e) {
89             e.printStackTrace();
90         } finally {
91             disconnect();
92         }
93
94         return lists;
95
96     }
97
98 }
99
```

## 회원 목록 출력 ( memberList.jsp )

### ■ MemberList.jsp 수정

- 이전 chapter11에서 작성했던 문서를 일부 수정하여 사용
- 컨트롤러에서 전달받은 배열 객체를 빈즈로 생성
  - ▶ 생성된 빈즈는 각각의 요소마다 회원 정보가 저장된 객체가 존재함
- 각 요소 내에 존재하는 회원정보를 순환문을 사용해 출력
- 하단에 회원 정보 입력을 위한 [INSERT] 버튼의 하이퍼링크 수정

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <%@ page import="java.util.ArrayList" %>
5 <%@ page import="chapter12.*" %>
6
7 <jsp:useBean id="memberlist" scope="request" class="java.util.ArrayList" />
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12   <meta charset="UTF-8">
13   <link rel="stylesheet" type="text/css" href="style.css">
14 </head>
15 <body>
16
17   <h2 align=center>회원 목록</h2>
18   <hr align=center width=900 size=3 noshade>
19
20   <form name="memberList">
21
22     <table class=main>
23
24       <tr align=center>
25         <td class=title2 width=35>번호</td>
26         <td class=title2 width=80>이름</td>
27         <td class=title2 width=170>전자우편</td>

```

```

28         <td class=title2 width=120>전화번호</td>
29         <td class=title2 width=150>회사명</td>
30         <td class=title2 width=90>생년월일</td>
31         <td class=title2 width=70>직위</td>
32         <td class=title2 width=35>수정</td>
33         <td class=title2 width=35>삭제</td>
34     </tr>
35     <%
36         for( MemberVO mv : (ArrayList<MemberVO>) memberlist ) {
37     %>
38     <tr align=center>
39         <td class=content><%=mv.getMemberID()%></td>
40         <td class=content><%=mv.getMemberName()%></td>
41         <td class=content><%=mv.getMemberMail()%></td>
42         <td class=content><%=mv.getMemberTel() %></td>
43         <td class=content><%=mv.getMemberComp() %></td>
44         <td class=content><%=mv.getMemberBirth() %></td>
45         <td class=content><%=mv.getMemberPos() %></td>
46         <td class=content>수정</td>
47         <td class=content>삭제</td>
48     </tr>
49     <%
50         }
51     %>
52 </table>
53 <hr align=center width=900 size=1 noshade>

```

```
54
55     <table align=center>
56         <tr>
57             <td align=center>
58                 <input type="button" value="INSERT"
59                     onClick="location.replace('memberInsertForm.jsp')">
60             </td>
61         </tr>
62     </table>
63
64 </form>
65 <hr align=center width=900 size=3 noshade>
66
67 </body>
68 </html>
```

## ■ index.jsp 문서 실행 결과

회원 목록								
번호	이름	전자우편	전화번호	회사명	생년월일	직위	수정	삭제
3	이순신	lee@abc.om	010-1111-2222	코리아네트워크	2021-04-05	대리	수정	삭제
1	이광	leeshine5455@gmail.com	010-5455-5268	한국교통대학교	2021-04-06	교수	수정	삭제

INSERT

### 회원정보 입력

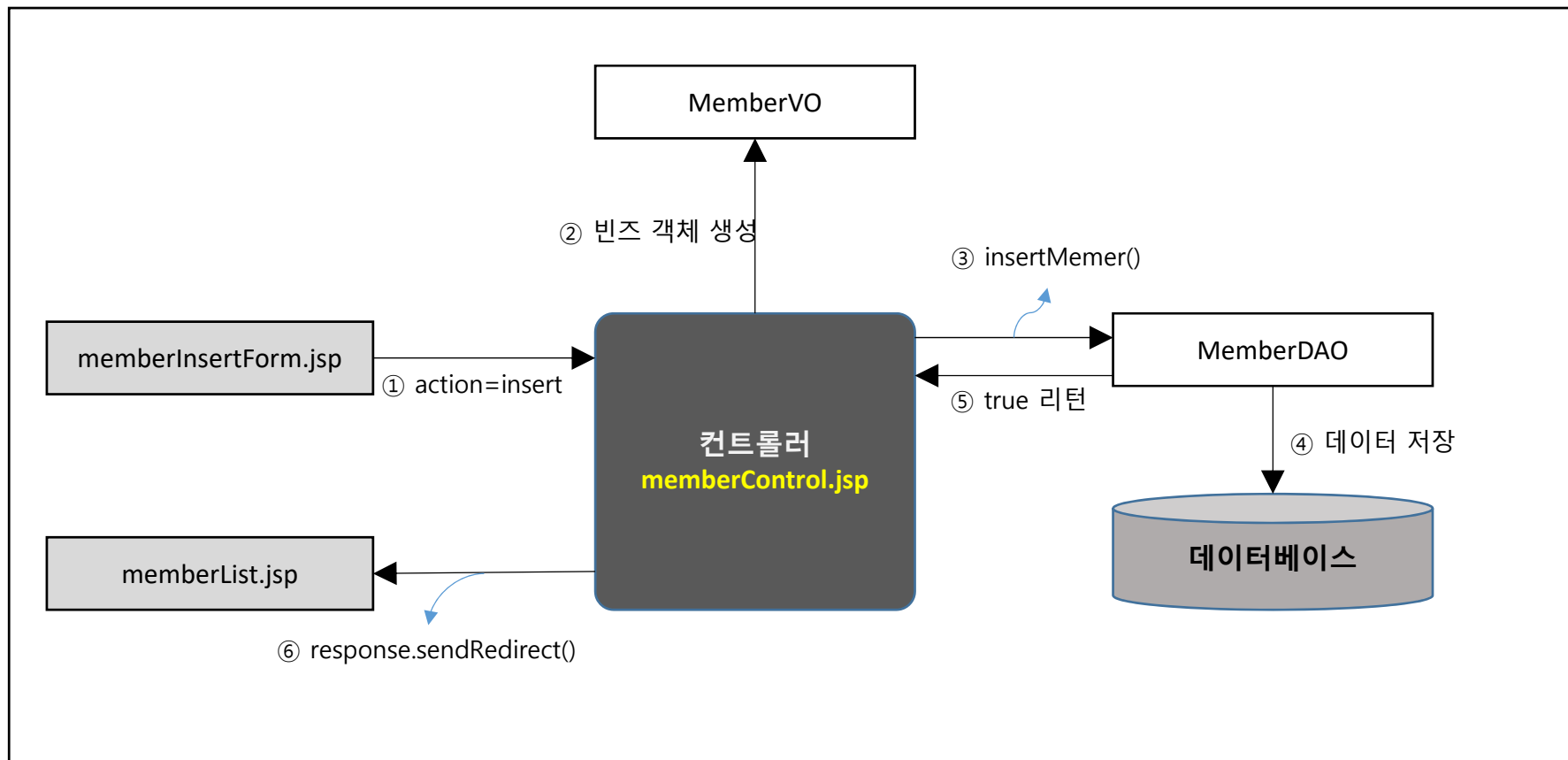
이름	<input type="text"/>
전자우편	<input type="text"/>
전화번호	<input type="text"/>
회사명	<input type="text"/>
생년월일	연도-월-일 <input type="text"/>
직위	<input type="text"/>

## 회원 정보 입력



# 회원 정보 입력의 개요

## ■ 회원 정보 입력 과정



### ■ memberInsertForm.jsp 문서 작성

- 이전 Chapter11에서 작성했던 내용을 일부 수정하여 사용
- <form> 태그의 action 속성 수정
  - ▶ 컨트롤러 문서인 MemberController.jsp 문서를 지정
- 컨트롤러에 insert 액션 코드를 전달
  - ▶ <input> 태그의 hidden 속성을 사용해 변수 action의 값에 insert를 지정하여 전달
- 버튼 수정
  - ▶ [LIST] 버튼을 추가

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5
6 <head>
7   <meta charset="UTF-8">
8   <link rel="stylesheet" type="text/css" href="style.css">
9 </head>
10
11 <body>
12
13   <h2 align=center>회원정보 입력</h2>
14   <hr align=center width=420 size=3 noshade>
15
16   <form name="memberInsert" method=post action="MemberController.jsp">
17
18     <input type=hidden name=action value="insert">
19
20     <table class=main width=400>
21
22       <tr>
23         <td class=title1 width=80>이름</td>
24         <td class=content width=320>
25           <input type=text name=memberName size=10 required>
26         </td>
27       </tr>
28
```

```
29<tr>
30    <td class=title1>전자우편</td>
31    <td class=content>
32        <input type=text name=memberMail size=30 required>
33    </td>
34</tr>
35
36<tr>
37    <td class=title1>전화번호</td>
38    <td class=content>
39        <input type=text name=memberTel size=20 required>
40    </td>
41</tr>
42
43<tr>
44    <td class=title1>회사명</td>
45    <td class=content>
46        <input type=text name=memberComp size=20>
47    </td>
48</tr>
49
50<tr>
51    <td class=title1>생년월일</td>
52    <td class=content>
53        <input type=date name=memberBirth size=20 required>
54    </td>
55</tr>
```

```
56
57     <tr>
58         <td class=title1>직위</td>
59         <td class=content>
60             <input type=text name=memberPos size=10>
61         </td>
62     </tr>
63
64 </table>
65 <hr align=center width=400 size=1 noshade>
66
67 <table align=center>
68     <tr>
69         <td align=left width=300>
70             <input type="submit" value="SAVE"> &nbsp;
71             <input type="reset" value="CANCEL">
72         </td>
73         <td align=right width=100>
74             <input type="button" value="LIST">
75         </td>
76     </tr>
77 </table>
78 </form>
79 <hr align=center width=420 size=3 noshade>
80
81 </body>
82 </html>
```

### ■ MemberController.jsp 문서 생성

- MemberVO 클래스의 빈즈 생성
  - ▶ memberInsertForm.jsp 문서에서 전달받은 파라미터 값을 저장하기 위한 빈즈 생성
  - ▶ 파라미터의 값은 빈즈를 통해 MemberDAO 클래스의 insertMember() 메서드에 전달되어 처리됨
- 액션 코드 추출 및 insertMember() 메서드 호출
  - ▶ memberInsertForm.jsp 문서에서 전달받은 insert 액션 코드를 추출
  - ▶ insert 액션 코드는 파라미터 값을 데이터베이스에 전달하는 insertMember() 메서드를 호출하는데 사용
- MemberDAO 클래스의 insertMember() 메서드 구현

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ page import="java.util.*" %>
4 <%@ page import="chapter12.*" %>
5
6 <% request.setCharacterEncoding("utf-8"); %>
7
8 <!-- MemberDAO 클래스의 빈즈 생성 ----- -->
9 <jsp:useBean id="memberDao" class="chapter12.MemberDAO" />
10
11 <!-- MemberVO 클래스의 빈즈 생성 ----- -->
12 <!-- memberInsertForm.jsp에서 전달된 모든 파라미터를 memberVo 빈즈에 저장 -->
13 <jsp:useBean id="memberVo" class="chapter12.MemberVO" />
14 <jsp:setProperty property="*" name="memberVo"/>
15 <%
16
17 // memberInsertForm.jsp에서 전달된 액션 코드 추출 -----
18     String action = request.getParameter("action");
19
20 //회원의 목록을 출력하기 위한 getMemberList() 메서드 호출
21 //전달받은 회원 정보 객체 배열을 memberList.jsp에 전달해 브라우저에 출력
22     if( action.equals("list") ) {
23
24         ArrayList<MemberVO> memberlist = memberDao.getMemberList();
25         request.setAttribute("memberlist", memberlist);
26         pageContext.forward("memberList.jsp");
27
28     }
```

```
29
30 // 회원 정보 저장을 위해 memberDao 빈즈의 insertMember() 메서드 호출
31 // insert() 메서드 수행 후 List문서 수행을 위해 list 액션 코드를 컨트롤러에 전달
32
33     if( action.equals("insert") ) {
34
35         if( memberDao.insertMember(memberVo) ) {
36             out.println("<script>");
37             out.println("alert('회원정보가 저장되었습니다')");
38             out.println("<script>");
39             response.sendRedirect("MemberController.jsp?action=list");
40         }
41
42     }
43
44 %>
```



### ■ MemberDAO.java 수정

- insertMember() 메서드 추가
  - ▶ 컨트롤러로부터 전달받은 memberVo 빈즈에 저장된 회원 정보를 getter()를 사용해 추출
  - ▶ 추출한 정보를 데이터베이스에 저장
  - ▶ 데이터베이스에 저장에 성공하면 컨트롤러에 true를 전달

```
1 package chapter12;
2
3 import java.sql.*;
4 import java.util.*;
5
6 public class MemberDAO {
7
8     // 데이터베이스 접속 관련 객체 선언 -----
9     Connection conn = null;
10    PreparedStatement pstmt = null;
11    ResultSet rs = null;
12
13    // 데이터베이스 접속 관련 변수 선언 -----
14    String jdbc_driver = "com.mysql.cj.jdbc.Driver";
15    String jdbc_url = "jdbc:mysql://localhost:3306/jspdb?serverTimezone=UTC";
16
17    // 데이터베이스 연결 -----
18*    void connect( ) {
19
20
21
22
23
24
25
26
27
28
29
30    // 데이터베이스 연결 해제 -----
31*    void disconnect() {
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59    // 전체 회원 목록을 출력 ( 회원 정보들을 각각 객체로 생성해 배열로 저장한 다음 전달 )
60*    public ArrayList<MemberVO> getMemberList() {
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
```

```
198 // 회원 정보 입력 (입력에 성공하면 true를 반환) -----
199 public Boolean insertMember( MemberVO memberVo ) {
200
201     connect();
202
203     try {
204
205         String Query = "insert into members ";
206         Query += "(memberName,memberMail,memberTel,memberComp,memberBirth,memberPos) ";
207         Query += "values (?, ?, ?, ?, ?, ?)";
208
209         pstmt = conn.prepareStatement(Query);
210         pstmt.setString(1, memberVo.getMemberName());
211         pstmt.setString(2, memberVo.getMemberMail());
212         pstmt.setString(3, memberVo.getMemberTel());
213         pstmt.setString(4, memberVo.getMemberComp());
214         pstmt.setString(5, memberVo.getMemberBirth());
215         pstmt.setString(6, memberVo.getMemberPos());
216         pstmt.executeUpdate();
217
218     } catch (Exception e) {
219         e.printStackTrace();
220         return false;
221     } finally {
222         disconnect();
223     }
224     return true;
225
226 }
227
228 }
```

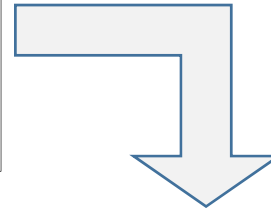
## 회원정보 입력

이름	<input type="text" value="강감찬"/>
전자우편	<input type="text" value="kang@bcd.com"/>
전화번호	<input type="text" value="010-3333-4444"/>
회사명	<input type="text" value="동아건설"/>
생년월일	<input type="text" value="2021-04-05"/> 
직위	<input type="text" value="과장"/>

SAVE

CANCEL

LIST



## 회원 목록

번호	이름	전자우편	전화번호	회사명	생년월일	직위	수정	삭제
6	강감찬	kang@bcd.com	010-3333-4444	동아건설	2021-04-05	과장	수정	삭제
3	이순신	lee@abc.om	010-1111-2222	코리아네트워크	2021-04-05	대리	수정	삭제
1	이광	leeshine5455@gmail.com	010-5455-5268	한국교통대학교	2021-04-06	교수	수정	삭제

INSERT

## 회원 정보 수정

# 회원 정보 수정의 개요

## ■ 수정할 회원 정보의 기존 내용을 출력

- memberList.jsp 문서 수정
  - ▶ [수정] 버튼의 하이퍼링크 생성
    - 각 회원 레코드의 [수정] 버튼을 클릭하면 modify 액션 코드와 memberId 값을 컨트롤러에 전달
- MemberController.jsp 문서 수정
  - ▶ memberList.jsp 문서에서 전달된 memberId 값을 MemberDAO의 getMember( ) 메서드에 전달
  - ▶ getMember( ) 메서드는 수정할 회원 정보를 추출하여 컨트롤러에 전달
- memberModifyForm.jsp 문서 수정
  - ▶ 전달받은 회원 정보를 추출하여 입력 양식에 출력
  - ▶ 실제 수정을 위해 update 액션 태그와 memberId 값을 컨트롤러에 전달

```

35<%
36    for( MemberVO mv : (ArrayList<MemberVO>) memberlist ) {
37    %>
38    <tr align=center>
39        <td class=content><%=mv.getMemberID()%></td>
40        <td class=content><%=mv.getMemberName()%></td>
41        <td class=content><%=mv.getMemberMail()%></td>
42        <td class=content><%=mv.getMemberTel() %></td>
43        <td class=content><%=mv.getMemberComp() %></td>
44        <td class=content><%=mv.getMemberBirth() %></td>
45        <td class=content><%=mv.getMemberPos() %></td>
46
47        <td class=content>
48            <a href="MemberController.jsp?action=modify&mId=<%=mv.getMemberID()%>">수정</a>
49        </td>
50
51        <td class=content>삭제</td>
52    </tr>
53    <%
54    }
55    %>

```

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.util.*" %>
4 <%@ page import="chapter12.*" %>
5
6 <% request.setCharacterEncoding("utf-8"); %>
7
8 <!-- MemberDAO 클래스의 빈즈 생성 ----- -->
9 <jsp:useBean id="memberDao" class="chapter12.MemberDAO" />
10
11 <!-- MemberVO 클래스의 빈즈 생성 ----- -->
12 <!-- memberInsertForm.jsp에서 전달된 모든 파라미터를 memberVo 빈즈에 저장 -->
13 <jsp:useBean id="memberVo" class="chapter12.MemberVO" />
14 <jsp:setProperty property="*" name="memberVo"/>
15 <%
16
17 // memberInsertForm.jsp에서 전달된 액션 코드 추출 -----
18     String action = request.getParameter("action");
19
20 //회원의 목록을 출력하기 위한 getMemberList() 메서드 호출
21 //전달받은 회원 정보 객체 배열을 memberList.jsp에 전달해 브라우저에 출력
22     if( action.equals("list") ) {
23
24         ArrayList<MemberVO> memberlist = memberDao.getMemberList();
25         request.setAttribute("memberlist", memberlist);
26         pageContext.forward("memberList.jsp");
27     }
28

```



```
30 // 회원 정보 저장을 위해 memberDao 빈즈의 insertMember() 메서드 호출
31 // insertMember() 메서드 수행 후 List문서 수행을 위해 list 액션 코드를 컨트롤러에 전달
32
33     if( action.equals("insert") ) {
34
35         if( memberDao.insertMember(memberVo) ) {
36             out.println("<script>");
37             out.println("alert('회원정보가 저장되었습니다')");
38             out.println("<script>");
39             response.sendRedirect("MemberController.jsp?action=list");
40         }
41     }
42 }
43
44 // 회원 정보 수정을 위한 기본 정보 추출을 위해 memberDao 빈즈의 getMember() 메서드 호출
45 // getMember() 메서드 수행 후 전달받은 회원 정보를 memberModifyForm.jsp에 전달
46     if( action.equals("modify") ) {
47
48         int mId = Integer.parseInt(request.getParameter("mId"));
49
50         MemberVO member = memberDao.getMember(mId);
51         request.setAttribute("member", member);
52         pageContext.forward("memberModifyForm.jsp");
53     }
54 }
55
56 %>
```

```
1 package chapter12;
2
3 import java.sql.*;
4
5
6 public class MemberDAO {
7
8     // 데이터베이스 접속 관련 객체 선언 -----
9     Connection conn = null;
10    PreparedStatement pstmt = null;
11    ResultSet rs = null;
12
13    // 데이터베이스 접속 관련 변수 선언 -----
14    String jdbc_driver = "com.mysql.cj.jdbc.Driver";
15    String jdbc_url = "jdbc:mysql://localhost:3306/jspdb?serverTimezone=UTC";
16
17    // 데이터베이스 연결 -----
18    void connect( ) {
19
20
21
22
23
24
25
26
27
28
29
30    // 데이터베이스 연결 해제 -----
31    void disconnect() {
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59    // 전체 회원 목록을 출력 ( 회원 정보들을 각각 객체로 생성해 배열로 저장한 다음 전달 )
60    public ArrayList<MemberVO> getMemberList() {
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
```

```
98 // 회원 정보 입력 (입력에 성공하면 true를 반환 ) -----
99*   public Boolean insertMember( MemberVO memberVo ) {
127
128 // 수정하고자 하는 회원 정보 추출
129*   public MemberVO getMember(int mId) {
130
131       connect();
132
133       String Query = "select * from members where memberID=?";
134       MemberVO member = new MemberVO();
135
136       try {
137
138           pstmt = conn.prepareStatement(Query);
139           pstmt.setInt(1, mId);
140           ResultSet rs = pstmt.executeQuery();
141
142           rs.next();
143
144           member.setMemberID(rs.getInt("memberID"));
145           member.setMemberName(rs.getString("memberName"));
146           member.setMemberMail(rs.getString("memberMail"));
147           member.setMemberTel(rs.getString("memberTel"));
148           member.setMemberComp(rs.getString("memberComp"));
```

```
149         member.setMemberBirth(rs.getString("memberBirth"));
150         member.setMemberPos(rs.getString("memberPos"));
151
152     }catch(Exception e) {
153         e.printStackTrace();
154     } finally {
155         disconnect();
156     }
157
158     return member;
159
160 }
161
162 }
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <%@ page import="chapter12.*" %>
5
6 <jsp:useBean id="member" class="chapter12.MemberVO" scope="request" />
7
8 <!DOCTYPE html>
9 <html>
10 <head>
11   <meta charset="UTF-8">
12   <link rel="stylesheet" type="text/css" href="style.css">
13 </head>
14 <body>
15
16   <h2 align=center>회원정보 수정</h2>
17   <hr align=center width=420 size=3 noshade>
18
19   <form name="memberModify" method=post action="MemberController.jsp">
20
21     <input type=hidden name="action" value="update">
22     <input type=hidden name="mId" value="<%=member.getMemberID()%>">
23
24     <table class=main width=400>
25
26       <tr>
27         <td class=title1 width=80>이름</td>
28         <td class=content width=320><%=member.getMemberName()%></td>
29       </tr>
```

```

30
31     <tr>
32         <td class=title1>전자우편</td>
33         <td class=content>
34             <input type=text name=memberMail size=30
35                 value="<%=member.getMemberMail() %>" required>
36         </td>
37     </tr>
38
39     <tr>
40         <td class=title1>전화번호</td>
41         <td class=content>
42             <input type=text name=memberTel size=20
43                 value="<%=member.getMemberTel() %>" required>
44         </td>
45     </tr>
46
47     <tr>
48         <td class=title1>회사명</td>
49         <td class=content>
50             <input type=text name=memberComp
51                 value="<%=member.getMemberComp() %>" size=20>
52         </td>
53     </tr>
54
55     <tr>
56         <td class=title1>생년월일</td>
57         <td class=content>

```

```

58         <input type=date name=memberBirth size=20
59             value="<%=member.getMemberBirth() %>" required>
60     </td>
61 </tr>
62
63 <tr>
64     <td class=title1>직위</td>
65     <td class=content>
66         <input type=text name=memberPos
67             value="<%=member.getMemberPos() %>" size=10>
68     </td>
69 </tr>
70
71 </table>
72 <hr align=center width=400 size=1 noshade>
73
74 <table align=center>
75     <tr>
76         <td align=center>
77             <input type="submit" value="MODIFY"> &nbsp;
78             <input type="reset" value="CANCEL">
79         </td>
80     </tr>
81 </table>
82
83 </form>
84 <hr align=center width=420 size=3 noshade>
85
86 </body>
87 </html>

```



## 회원 목록

번호	이름	전자우편	전화번호	회사명	생년월일	직위	수정	삭제
6	강감찬	kang@bcd.com	010-3333-4444	동아건설	2021-04-05	과장	<a href="#">수정</a>	<a href="#">삭제</a>
3	이순신	lee@abc.com	010-1111-2222	코리아네트웍	2021-04-05	대리	<a href="#">수정</a>	<a href="#">삭제</a>
1	이광	leeshine5455@gmail.com	010-5455-5268	한국교통대학교	2021-04-06	교수	<a href="#">수정</a>	<a href="#">삭제</a>

INSERT

## 회원정보 수정

이름	강감찬
전자우편	<input type="text" value="kang@bcd.com"/>
전화번호	<input type="text" value="010-3333-4444"/>
회사명	<input type="text" value="동아건설"/>
생년월일	<input type="text" value="2021-04-05"/> <input type="button" value="📅"/>
직위	<input type="text" value="과장"/>

MODIFY

CANCEL



# 회원 정보 수정

## ■ 회원 정보 수정 처리

- MemberController.java 수정

- ▶ meberModifyForm.jsp에서 전달받은 액션 코드를 사용해 데이터베이스를 수정하는 updateMember() 메서드 호출
- ▶ updateMember() 메서드에 전달받은 memberID 값을 전달

- MemberDAO.java 수정

- ▶ 데이터베이스를 수정하는 updateMember() 메서드를 구현
- ▶ updateMember() 메서드는 회원 정보를 수정한 후 컨트롤러에 true를 전달
- ▶ 컨트롤러는 컨트롤러 자신에게 list 액션 속성을 전달해 memberList.jsp 문서를 실행함

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ page import="java.util.*" %>
4 <%@ page import="chapter12.*" %>
5
6 <% request.setCharacterEncoding("utf-8"); %>
7
8 <!-- MemberDAO 클래스의 빈즈 생성 ----- -->
9 <jsp:useBean id="memberDao" class="chapter12.MemberDAO" />
10
11 <!-- MemberVO 클래스의 빈즈 생성 ----- -->
12 <!-- memberInsertForm.jsp에서 전달된 모든 파라미터를 memberVo 빈즈에 저장 -->
13 <jsp:useBean id="memberVo" class="chapter12.MemberVO" />
14 <jsp:setProperty property="*" name="memberVo"/>
15 <%
16
17 // memberInsertForm.jsp에서 전달된 액션 코드 추출 -----
18     String action = request.getParameter("action");
19
20 //회원의 목록을 출력하기 위한 getMemberList() 메서드 호출
21 //전달받은 회원 정보 객체 배열을 memberList.jsp에 전달해 브라우저에 출력
22     if( action.equals("list") ) {
23
24         ArrayList<MemberVO> memberlist = memberDao.getMemberList();
25         request.setAttribute("memberlist", memberlist);
26         pageContext.forward("memberList.jsp");
27
28     }

```

```
29
30 // 회원 정보 저장을 위해 memberDao 빈즈의 insertMember() 메서드 호출
31 // insertMember() 메서드 수행 후 List문서 수행을 위해 list 액션 코드를 컨트롤러에 전달
32
33     if( action.equals("insert") ) {
34
35         if( memberDao.insertMember(memberVo) ) {
36             out.println("<script>");
37             out.println("alert('회원정보가 저장되었습니다')");
38             out.println("<script>");
39             response.sendRedirect("MemberController.jsp?action=list");
40         }
41     }
42
43
44 // 회원 정보 수정을 위한 기본 정보 추출을 위해 memberDao 빈즈의 getMember() 메서드 호출
45 // getMember() 메서드 수행 후 전달받은 회원 정보를 memberModifyForm.jsp에 전달
46     if( action.equals("modify") ) {
47
48         int mId = Integer.parseInt(request.getParameter("mId"));
49
50         MemberVO member = memberDao.getMember(mId);
51         request.setAttribute("member", member);
52         pageContext.forward("memberModifyForm.jsp");
53
54     }
55
```

```
56 // 회원 정보 수정하기 위해 memberDao 빈즈의 updateMember() 메서드 호출
57 // updateMember()메서드 수행 후 메서드 수행 후 List문서 수행을 위해 list 액션 코드를 컨트롤러에 전달
58     if( action.equals("update") ) {
59
60         int mId = Integer.parseInt(request.getParameter("mId"));
61
62         if( memberDao.updateMember(mId, memberVo) ) {
63             response.sendRedirect("MemberController.jsp?action=list");
64         }
65
66     }
67
68 %>
```

```

1 package chapter12;
2
3 import java.sql.*;
4
5
6 public class MemberDAO {
7
8 // 데이터베이스 접속 관련 객체 선언 -----
9     Connection conn = null;
10    PreparedStatement pstmt = null;
11    ResultSet rs = null;
12
13 // 데이터베이스 접속 관련 변수 선언 -----
14    String jdbc_driver = "com.mysql.cj.jdbc.Driver";
15    String jdbc_url = "jdbc:mysql://localhost:3306/jspdb?serverTimezone=UTC";
16
17 // 데이터베이스 연결 -----
18 void connect( ) {
19
20
21
22
23
24
25
26
27
28
29
30 // 데이터베이스 연결 해제 -----
31 void disconnect() {
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59 // 전체 회원 목록을 출력 ( 회원 정보들을 각각 객체로 생성해 배열로 저장한 다음 전달 )
60 public ArrayList<MemberVO> getMemberList() {
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98 // 회원 정보 입력 (입력에 성공하면 true를 반환 ) -----
99 public Boolean insertMember( MemberVO memberVo ) {

```

```
127
128 // 수정하고자 하는 회원 정보 추출
129*   public MemberVO getMember(int mId) {
161
162 // 회원정보 수정
163
164*   public boolean updateMember(int mId, MemberVO memberVo) {
165
166     connect();
167
168     try {
169
170         String Query = "update members set ";
171         Query += "memberMail=?, memberTel=?, memberComp=?, ";
172         Query += "memberBirth=?, memberPos=? ";
173         Query += "where memberId=?";
174
175         pstmt = conn.prepareStatement(Query);
176         pstmt.setString(1, memberVo.getMemberMail());
177         pstmt.setString(2, memberVo.getMemberTel());
178         pstmt.setString(3, memberVo.getMemberComp());
179         pstmt.setString(4, memberVo.getMemberBirth());
180         pstmt.setString(5, memberVo.getMemberPos());
181         pstmt.setInt(6, mId);
182
```

```
183         pstmt.executeUpdate();
184
185     } catch(Exception e) {
186         e.printStackTrace();
187         return false;
188     } finally {
189         disconnect();
190     }
191
192     return true;
193 }
194 }
```

## 회원 목록

번호	이름	전자우편	전화번호	회사명	생년월일	직위	수정	삭제
6	강감찬	kang@bcd.com	010-3333-4444	동아건설	2021-04-05	과장	수정	삭제
3	이순신	lee@abc.om	010-1111-2222	코리아네트웍	2021-04-05	대리	수정	삭제
1	이광	leeshine5455@gmail.com	010-5455-5268	한국교통대학교	2021-04-06	교수	수정	삭제

INSERT

## 회원정보 수정

이름	강감찬
전자우편	<input type="text" value="kang@gmail.com"/>
전화번호	<input type="text" value="010-5555-6666"/>
회사명	<input type="text" value="동아시스템"/>
생년월일	<input type="text" value="2021-04-05"/> <input type="button" value="📅"/>
직위	<input type="text" value="부장"/>

MODIFY

CANCEL

## 회원 목록

번호	이름	전자우편	전화번호	회사명	생년월일	직위	수정	삭제
6	강감찬	kang@gmail.com	010-5555-6666	동아시스템	2021-04-05	부장	수정	삭제
3	이순신	lee@abc.om	010-1111-2222	코리아네트웍	2021-04-05	대리	수정	삭제
1	이광	leeshine5455@gmail.com	010-5455-5268	한국교통대학교	2021-04-06	교수	수정	삭제

INSERT



## 회원 정보 삭제

# 회원 정보 삭제 개요

## ■ 회원 정보 삭제

- memberList.jsp 문서 수정
  - ▶ [삭제] 버튼의 하이퍼링크 생성
  - ▶ 각 회원 레코드의 [삭제] 버튼을 클릭하면 delete 액션 코드와 memberID 값을 컨트롤러에 전달
- MemberController.jsp 문서 수정
  - ▶ memberList.jsp 문서에서 전달된 memberID 값을 MemberDAO의 deleteMember( ) 메서드에 전달
  - ▶ deleteMember( ) 메서드는 해당 레코드를 삭제
  - ▶ 회원 정보 삭제 후 컨트롤러에 true를 반환
  - ▶ 컨트롤러는 컨트롤러 자신에 list를 전달해 memberList.jsp 문서를 출력함

# 과제

## ■ 과제

- 삭제 부분을 추가로 작성한 다음 프로젝트를 압축하여 제출
  - 워크스페이스의 Chapter12 폴더를 압축하여 제출
  - 압축 파일의 이름 : 학번-성명.zip
- 제출 방법
  - e-캠퍼스에 제출
  - 제출 일시는 e-캠퍼스에서 참조

**수고하셨습니다**