

JSP 연습문제 정답

연습문제 2-1 서블릿 클래스 작성 연습

[예제 2-8]과 [예제 2-9]를 각각 다음과 같이 수정하면 됩니다.

[예제 2-8] 개인 정보를 입력받는 HTML 문서 - brain/PinfoInput.html

```
1 <HTML>
2   <HEAD>
3     <META http-equiv="Content-Type" content="text/html; charset=euc-kr">
4     <TITLE>개인 정보 입력 화면</TITLE>
5   </HEAD>
6   <BODY>
7     <H2>개인 정보 입력</H2>
8     <FORM ACTION=/brain/pinfo-result METHOD=POST>
9       이름: <INPUT TYPE=TEXT NAME=NAME><BR>
10      아이디: <INPUT TYPE=TEXT NAME=ID><BR>
11      패스워드: <INPUT TYPE=PASSWORD NAME=PASSWORD><BR><BR>
12      성별:
13        남<INPUT TYPE=RADIO NAME=GENDER VALUE=MALE>
14        여<INPUT TYPE=RADIO NAME=GENDER VALUE=FEMAIL><BR><BR>
15      메일 수신 거부:
16        공지 메일<INPUT TYPE=CHECKBOX NAME=INOTICE>
17        광고 메일<INPUT TYPE=CHECKBOX NAME=CNOTICE>
18        배송 확인 메일<INPUT TYPE=CHECKBOX NAME=DNOTICE><BR><BR>
19      직업:
20      <SELECT NAME=JOB>
21        <OPTION>회사원</OPTION>
22        <OPTION>학생</OPTION>
23        <OPTION>기타</OPTION>
24      </SELECT><BR><BR>
25      <INPUT TYPE=RESET VALUE="취소">
26      <INPUT TYPE=SUBMIT VALUE="확인">
27    </FORM>
28  </BODY>
29 </HTML>
```

```
1 import javax.servlet.http.*;
2 import javax.servlet.*;
3 import java.io.*;
4 public class PersonalInfoServlet extends HttpServlet {
5     public void doPost(HttpServletRequest request,
6                         HttpServletResponse response)
7         throws IOException, ServletException {
8         request.setCharacterEncoding("euc-kr");
9         String name = request.getParameter("NAME");
10        String id = request.getParameter("ID");
11        String password = request.getParameter("PASSWORD");
12        String gender = request.getParameter("GENDER");
13        if (gender.equals("MALE"))
14            gender = "남";
15        else
16            gender = "여";
17        String iNotice = request.getParameter("INOTICE");
18        String clNotice = request.getParameter("CNOTICE");
19        String dNotice = request.getParameter("DNOTICE");
20        String job = request.getParameter("JOB");
21        response.setContentType("text/html; charset=euc-kr");
22        PrintWriter out = response.getWriter();
23        out.println("<HTML>");
24        out.println("<HEAD><TITLE>개인 정보 입력 - 결과 화면</TITLE></HEAD>");
25        out.println("<BODY>");
26        out.println("<H2>개인 정보 입력</H2>");
27        out.printf("이름: %s <BR>", name);
28        out.printf("아이디: %s <BR>", id);
29        out.printf("암호: %s <BR>", password);
30        out.printf("성별: %s <BR>", gender);
31        out.printf("공지 메일: %s <BR>", noticeToHangul(iNotice));
32        out.printf("광고 메일: %s <BR>", noticeToHangul(clNotice));
33        out.printf("배송 확인 메일: %s <BR>", noticeToHangul(dNotice));
34        out.printf("직업: %s <BR>", job);
35        out.println("</BODY>");
36        out.println("</HTML>");
37    }
38    private String noticeToHangul(String notice) {
39        if (notice == null)
40            return "받지 않음";
41        else if (notice.equals("on"))
42            return "받음";
43        else
44            return notice;
45    }
46 }
```

연습문제 3-1 JSP 페이지 작성 연습

[예제 3-6]과 [예제 3-7]을 각각 다음과 같이 수정하면 됩니다.

[예제 3-6] 웹 브라우저로부터 이름을 입력받는 HTML 문서 - brain03/YourName.html

```
1 <HTML>
2     <HEAD>
3         <META http-equiv="Content-Type" content="text/html; charset=euc-kr">
4         <TITLE>이름 입력</TITLE>
5     </HEAD>
6     <BODY>
7         당신의 이름을 입력하십시오.
8         <FORM ACTION=/brain03/Hi.jsp METHOD=POST>
9             이름: <INPUT TYPE=TEXT NAME=YOURNAME>
10            <INPUT TYPE=SUBMIT VALUE='확인'>
11        </FORM>
12    </BODY>
13 </HTML>
```

[예제 3-7] 입력된 이름을 가지고 인사말을 출력하는 JSP 페이지 - brain03/Hi.jsp

```
1 <%@page contentType="text/html; charset=euc-kr"%>
2 <% request.setCharacterEncoding("euc-kr"); %>
3 <HTML>
4     <HEAD><TITLE>인사하기</TITLE></HEAD>
5     <BODY>
6         안녕하세요, <%= request.getParameter("YOURNAME") %>님
7     </BODY>
8 </HTML>
```

연습문제 3-2 파일 읽기 연습

다음과 같이 작성하면 됩니다.

노래 파일 목록을 생성하는 JSP 페이지 - brain03/SongList.jsp

```
1 <%@page contentType="text/html; charset=euc-kr"%>
2 <%@page import="java.io.File" %>
3 <%
4     String dirPath = application.getRealPath("/WEB-INF/songs");
5     File dir = new File(dirPath);
```

```

6     String filenames[] = dir.list();
7     %>
8     <HTML>
9         <HEAD>
10            <HEAD><TITLE>노래 목록</TITLE></HEAD>
11        </HEAD>
12        <BODY>
13            <H4>노래 파일 목록</H4>
14            <% for (String filename : filenames) { %>
15                <A HREF='SongReader.jsp?FILE_NAME=<%= filename %>'>
16                    <%= filename %>
17                </A><BR>
18            <% } %>
19        </BODY>
20    </HTML>

```

노래 가사를 읽어서 출력하는 JSP 페이지 - brain03/SongReader.jsp

```

1     <%@page contentType="text/html; charset=euc-kr" %>
2     <%@page import="java.io.*"%>
3     <% String filename = request.getParameter("FILE_NAME"); %>
4     <HTML>
5         <HEAD><TITLE>노래 가사</TITLE></HEAD>
6         <BODY>
7             <H4>파일 이름: <%= filename %></H4>
8             <%
9                 BufferedReader reader = null;
10                try {
11                    String filePath = application.getRealPath(
12                        "/WEB-INF/songs/" + filename);
13                    reader = new BufferedReader(new FileReader(filePath));
14                    while (true) {
15                        String str = reader.readLine();
16                        if (str == null)
17                            break;
18                        out.println(str + "<BR>");
19                    }
20                } catch (FileNotFoundException fnfe) {
21                    out.println("파일이 존재하지 않습니다.");
22                } catch (IOException ioe) {
23                    out.println("파일을 읽을 수 없습니다.");
24                }
25            <%
26            finally {
27                try {

```

```

28             reader.close();
29         }
30         catch (Exception e) {
31         }
32     }
33     %>
34 </BODY>
35 </HTML>

```

** 이 예제는 붉은 글씨로 표시된 부분을 제외하고는 [예제 3-14]와 동일합니다.

연습문제 4-1 쿠키 데이터 읽기 연습

다음과 같이 작성하면 됩니다.

웹 브라우저로부터 받은 모든 쿠키 데이터를 읽어서 출력하는 JSP 페이지 - brain04/AllCookies.jsp

```

1  <%@page contentType="text/html; charset=euc-kr"%>
2  <% Cookie[] cookies = request.getCookies(); %>
3  <HTML>
4      <HEAD><TITLE>쿠키 목록</TITLE></HEAD>
5      <BODY>
6          <%
7              if (cookies == null) {
8                  out.println("쿠키가 없습니다.");
9              } else {
10                 for (Cookie cookie : cookies)
11                     out.println(cookie.getName() + ": " + cookie.getValue()
12 + "<BR>");
13             }
14         %>
15 </BODY>
16 </HTML>

```

연습문제 4-2 세션 사용하기 연습

[예제 4-12]를 다음과 같이 수정하세요.

```
1 <HTML>
2   <HEAD>
3     <META http-equiv="Content-Type" content="text/html; charset=euc-kr">
4     <TITLE>회원 가입</TITLE>
5   </HEAD>
6   <BODY>
7     개인 정보를 입력하십시오.
8     <FORM ACTION=Addresses.jsp METHOD=POST>
9       아이디: <INPUT TYPE=TEXT NAME=ID><BR>
10      비밀번호: <INPUT TYPE=PASSWORD NAME=PASSWORD><BR>
11      이름: <INPUT TYPE=TEXT NAME=NAME><BR>
12      <INPUT TYPE=SUBMIT VALUE='확인'>
13    </FORM>
14  </BODY>
15 </HTML>
```

그리고 다음과 같은 JSP 페이지를 추가하세요.

```
1 <%@page contentType="text/html; charset=euc-kr"%>
2 <%
3   request.setCharacterEncoding("euc-kr");
4   String id = request.getParameter("ID");
5   String password = request.getParameter("PASSWORD");
6   String name = request.getParameter("NAME");
7   session.setAttribute("ID", id);
8   session.setAttribute("PASSWORD", password);
9   session.setAttribute("NAME", name);
10 %>
11 <HTML>
12   <HEAD><TITLE>회원 가입</TITLE></HEAD>
13   <BODY>
14     연락처를 입력하십시오.
15     <FORM ACTION=Agreement.jsp METHOD=POST>
16       주소: <INPUT TYPE=TEXT NAME=ADDRESS><BR>
17       전화번호: <INPUT TYPE=PASSWORD NAME=TELNO><BR>
18       이메일: <INPUT TYPE=TEXT NAME=EMAIL><BR>
19       <INPUT TYPE=SUBMIT VALUE='확인'>
20     </FORM>
21   </BODY>
22 </HTML>
```

그리고 [예제 4-13]과 [예제 4-14]를 다음과 같이 수정하세요.

[예제 4-13] 약관 동의 화면을 제공하는 JSP 페이지 - brain04/subscribe/Agreement.jsp

```
1 <%@page contentType="text/html; charset=euc-kr"%>
2 <%
3     request.setCharacterEncoding("euc-kr");
4     String address = request.getParameter("ADDRESS");
5     String telNo = request.getParameter("TELNO");
6     String email = request.getParameter("EMAIL");
7     session.setAttribute("ADDRESS", address);
8     session.setAttribute("TELNO", telNo);
9     session.setAttribute("EMAIL", email);
10 %>
11 <HTML>
12     <HEAD><TITLE>회원 가입</TITLE></HEAD>
13     <BODY>
14         <H3>약관</H3>
15         ----- <BR>
16         1. 회원 정보는 웹 사이트의 운영을 위해서만 사용됩니다. <BR>
17         2. 웹 사이트의 정상 운영을 방해하는 회원은 탈퇴 처리합니다. <BR>
18         ----- <BR>
19         <FORM ACTION=Subscribe.jsp METHOD=POST>
20             위의 약관에 동의하십니까?
21             <INPUT TYPE=RADIO NAME=AGREE VALUE=YES>동의함
22             <INPUT TYPE=RADIO NAME=AGREE VALUE=NO>동의하지 않음<BR><BR>
23             <INPUT TYPE=SUBMIT VALUE='확인'>
24         </FORM>
25     </BODY>
26 </HTML>
```

[예제 4-14] 회원 정보를 저장하는 JSP 페이지 - brain04/subscribe/Subscribe.jsp

```
1 <%@page contentType="text/html; charset=euc-kr"%>
2 <%@page import="java.io.*"%>
3 <%
4     String agree = request.getParameter("AGREE");
5     String result = null;
6     if (agree.equals("YES")) {
7         String id = (String) session.getAttribute("ID");
8         String password = (String) session.getAttribute("PASSWORD");
9         String name = (String) session.getAttribute("NAME");
10        String address = (String) session.getAttribute("ADDRESS");
11        String telNo = (String) session.getAttribute("TELNO");
12        String email = (String) session.getAttribute("EMAIL");
```

```

13     PrintWriter writer = null;
14     try {
15         String filePath = application.getRealPath(
16             "/WEB-INF/" + id + ".txt");
17         writer = new PrintWriter(filePath);
18         writer.println("아이디: " + id);
19         writer.println("패스워드: " + password);
20         writer.println("이름: " + name);
21         writer.println("주소: " + address);
22         writer.println("전화번호: " + telNo);
23         writer.println("이메일: " + email);
24         result = "SUCCESS";
25     }
26     catch (IOException ioe) {
27         result = "FAIL";
28     }
29     finally {
30         try {
31             writer.close();
32         }
33         catch (Exception e) {
34             }
35     }
36     else {
37         result = "FAIL";
38     }
39     session.invalidate();
40     response.sendRedirect("Result.jsp?RESULT=" + result);
41     %>

```

[예제 4-15]는 수정할 필요가 없습니다.

연습문제 5-1 에러 페이지 작성 연습

[예제 3-14]를 다음과 같이 수정하세요.

[예제 3-14] 텍스트 파일의 내용을 읽어서 웹 브라우저로 출력하는 JSP 페이지 - brain03/FileReader.jsp

```

1     <%@page contentType="text/html; charset=euc-kr" errorPage="FileError.jsp"
2     %>
3     <%@page import="java.io.*" %>
4     <HTML>
5         <HEAD><TITLE>파일 읽기</TITLE></HEAD>

```

```

5     <BODY>
6     <%
7         BufferedReader reader = null;
8         try {
9             String filePath = application.getRealPath(
10                "/WEB-INF/input.txt");
11             reader = new BufferedReader(new FileReader(filePath));
12             while (true) {
13                 String str = reader.readLine();
14                 if (str == null)
15                     break;
16                 out.println(str + "<BR>");
17             }
18         } finally {
19             try {
20                 reader.close();
21             }
22             catch (Exception e) {
23             }
24         }
25     %>
26 </BODY>
27 </HTML>

```

** [예제 3-14]에 있던 두 개의 catch 블록을 삭제했습니다. 하지만 익셉션이 발생했을 경우에도 JSP 페이지를 끝내기 전에 파일을 닫아야 하므로 finally 블록은 그대로 두었습니다.

그리고 위의 JSP 페이지가 발생하는 익셉션을 처리하는 에러 페이지를 다음과 같이 작성하세요.

위의 JSP 페이지가 발생하는 익셉션을 처리하는 에러 페이지 - [brain03/FileError.jsp](#)

```

1  <%@page contentType="text/html; charset=euc-kr" isErrorPage="true" %>
2  <%@page import="java.io.*" %>
3  <% response.setStatus(200); %>
4  <HTML>
5      <HEAD><TITLE>파일 에러</TITLE></HEAD>
6      <BODY>
7          <%
8              if (exception instanceof FileNotFoundException)
9                  out.println("파일이 존재하지 않습니다.");
10             else
11                 out.println("파일을 읽을 수 없습니다.");
12         %>
13     </BODY>
14 </HTML>

```

연습문제 6-1 init 메서드 작성 연습

다음과 같은 서블릿 클래스를 작성하면 됩니다.

약관 파일을 읽어서 출력하는 서블릿 클래스 - brain06/agreement

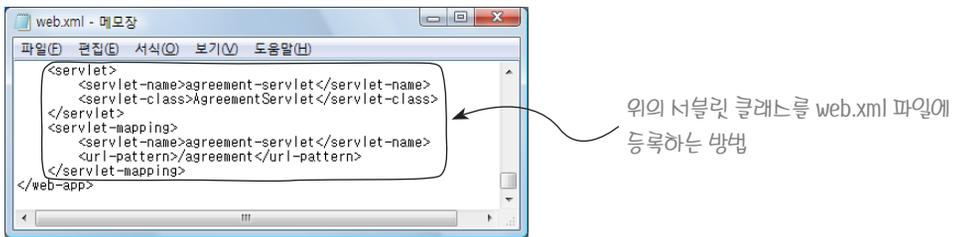
```
1 import javax.servlet.http.*;
2 import javax.servlet.*;
3 import java.io.*;
4 public class AgreementServlet extends HttpServlet {
5     private StringBuilder agreement = new StringBuilder();
6     public void init() {
7         BufferedReader reader = null;
8         try {
9             ServletContext context = getServletContext();
10            String filePath = context.getRealPath("/WEB-INF/agreement.txt");
11            reader = new BufferedReader(new FileReader(filePath));
12            while (true) {
13                String str = reader.readLine();
14                if (str == null)
15                    break;
16                agreement.append(str);
17                agreement.append("<BR>");
18            }
19        }
20        catch (FileNotFoundException fnfe) {
21            System.out.println("약관 파일을 찾을 수 없습니다.");
22        }
23        catch (IOException ioe) {
24            System.out.println("약관 파일을 읽을 수 없습니다.");
25        }
26        finally {
27            try {
28                reader.close();
29            }
30            catch (Exception e) {
31            }
32        }
33    }
34    public void doGet(HttpServletRequest request,
35                       HttpServletResponse response)
36        throws IOException, ServletException {
37        response.setContentType("text/html;charset=euc-kr");
38        PrintWriter out = response.getWriter();
39        out.println("<HTML>");
40        out.println("<HEAD><TITLE>약관</TITLE></HEAD>");
41        out.println("<BODY>");
```

```

41     out.println("<H3>약관</H3>");
42     out.println(agreement.toString());
43     out.println("</BODY>");
44     out.println("</HTML>");
45 }
46 }

```

위의 서블릿 클래스를 web.xml 파일에 등록하는 방법은 다음과 같습니다.



그리고 나서 이 서블릿 클래스를 실행하기 전에 이 클래스가 읽을 약관 파일을 다음과 같이 만들어서 WEB-INF 디렉터리에 저장해 두면 됩니다.

약관 파일을 읽어서 출력하는 서블릿 클래스 - WEB-INF/agreement.txt

```

1 -----
2 1. 회원 정보는 웹 사이트의 운영을 위해서만 사용됩니다.
3 2. 웹 사이트의 정상 운영을 방해하는 회원은 탈퇴 처리합니다.
4 3. 유료 서비스는 사용료를 지불한 회원만 사용할 수 있습니다.
5 -----

```

연습문제 6-2 서블릿의 초기화 파라미터 사용하기

[예제 6-1]을 다음과 같이 수정하면 됩니다.

[예제 6-1] 피보나치 수열을 출력하는 서블릿 클래스 - brain06/fibonacci

```

1 import javax.servlet.http.*;
2 import javax.servlet.*;
3 import java.io.*;
4 import java.math.BigInteger;
5 public class FibonacciServlet extends HttpServlet {
6     private BigInteger arr[];

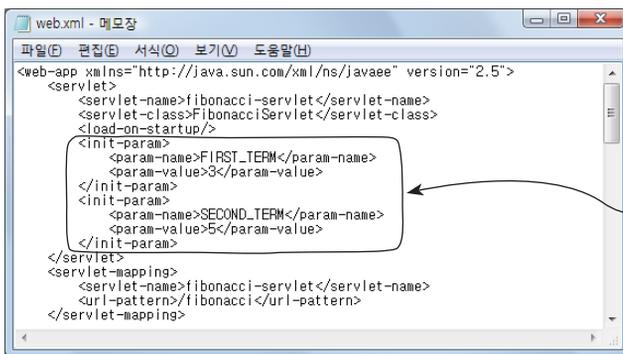
```

```

7     public void init() {
8         String firstTerm = getInitParameter("FIRST_TERM");
9         String secondTerm = getInitParameter("SECOND_TERM");
10        arr = new BigInteger[100];
11        arr[0] = new BigInteger(firstTerm);
12        arr[1] = new BigInteger(secondTerm);
13        for (int cnt = 2; cnt < arr.length; cnt++)
14            arr[cnt] = arr[cnt-2].add(arr[cnt-1]);
15    }
16    public void doGet(HttpServletRequest request,
17                       HttpServletResponse response)
18        throws IOException, ServletException {
19        String str = request.getParameter("NUM");
20        int num = Integer.parseInt(str);
21        if (num > 100)
22            num = 100;
23        response.setContentType("text/html;charset=euc-kr");
24        PrintWriter out = response.getWriter();
25        out.println("<HTML>");
26        out.println("<HEAD><TITLE>피보나치 수열</TITLE></HEAD>");
27        out.println("<BODY>");
28        for (int cnt = 0; cnt < num; cnt++)
29            out.println(arr[cnt] + " ");
30        out.println("</BODY>");
31        out.println("</HTML>");
32    }

```

그리고 나서 web.xml 파일에 위의 서블릿 클래스를 등록할 때 다음과 같은 방법으로 초기화 파라미터를 지정하면 됩니다.



피보나치 수열의 첫 번째와 두 번째 항의 값을 지정하는 초기화 파라미터

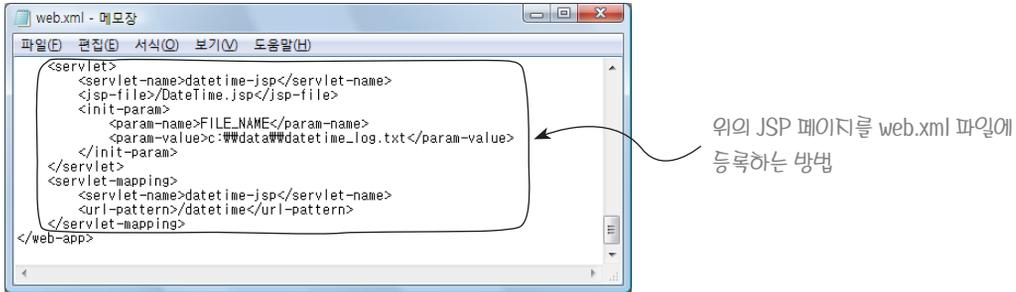
연습문제 6-3 JSP 페이지의 초기화 파라미터 사용 연습

[예제 6-5]를 다음과 같이 수정하면 됩니다.

[예제 6-5] jspInit, jspDestroy 메서드의 사용 예를 보여주는 JSP 페이지 - brain06/DateTime.jsp

```
1  <%@page contentType="text/html; charset=euc-kr"
                                import="java.io.*, java.util.*" %>
2  <%!
3      private PrintWriter logFile;
4      public void jspInit() {
5          String filename = getInitParameter("FILE_NAME");
6          try {
7              logFile = new PrintWriter(new FileWriter(filename, true));
8          }
9          catch (IOException e) {
10             System.out.printf("%TT - %s 파일을 열 수 없습니다. %n",
                                new GregorianCalendar(), filename);
11         }
12     }
13 %>
14 <HTML>
15     <HEAD><TITLE>현재의 날짜와 시각</TITLE></HEAD>
16     <BODY>
17     <%
18         GregorianCalendar now = new GregorianCalendar();
19         String date = String.format("현재 날짜: %TY년 %Tm월 %Te일",
                                     now, now, now);
20         String time = String.format("현재 시각: %TI시 %Tm분 %TS초",
                                     now, now, now);
21         out.println(date + "<BR>");
22         out.println(time + "<BR>");
23         if (logFile != null)
24             logFile.printf("%TF %TT에 호출되었습니다.%n", now, now);
25     %>
26     </BODY>
27 </HTML>
28 <%!
29     public void jspDestroy() {
30         if (logFile != null)
31             logFile.close();
32     }
33 %>
```

그리고 나서 위의 JSP 페이지를 다음과 같이 web.xml 파일에 등록해야 합니다.



그리고 이 예제를 실행할 때는 `http://localhost:8080/brain06/datetime`이라는 URL을 사용해야 합니다.

연습문제 6-4 getRealPath 메서드의 사용 연습

[예제 6-3]을 다음과 같이 수정하면 됩니다.

[예제 6-3] 인사말을 출력하는 서블릿 클래스 - brain06/greeting

```
1 import javax.servlet.http.*;
2 import javax.servlet.*;
3 import java.io.*;
4 import java.util.*;
5 public class GreetingServlet extends HttpServlet {
6     private PrintWriter logFile;
7     public void init() throws ServletException {
8         ServletContext context = getServletContext();
9         String filePath = context.getRealPath("WEB-INF\\log.txt");
10        try {
11            logFile = new PrintWriter(new FileWriter(filePath, true));
12        }
13        catch (IOException e) {
14            throw new ServletException(e);
15        }
16    }
17    public void doGet(HttpServletRequest request,
18                       HttpServletResponse response)
19                       throws IOException, ServletException {
20        String name = request.getParameter("NAME");
21        String greeting = "안녕하세요, " + name + "님.";
22        if (logFile != null) {
23            GregorianCalendar now = new GregorianCalendar();
24            logFile.printf("%TF %TT - %s %n", now, now, name);
25        }
26    }
27 }
```

```

23     }
24     response.setContentType("text/html;charset=euc-kr");
25     PrintWriter out = response.getWriter();
26     out.println("<HTML>");
27     out.println("<HEAD><TITLE>인사하기</TITLE></HEAD>");
28     out.println("<BODY>");
29     out.println(greeting);
30     out.println("</BODY>");
31     out.println("</HTML>");
32 }
33 public void destroy() {
34     if (logFile != null)
35         logFile.close();
36 }
37 }

```

연습문제 7-1 익스프레션 언어의 내장 객체 사용 연습

[예제 4-2]를 다음과 같이 수정하면 됩니다.

[예제 4-2] 쿠키 데이터를 읽는 JSP 페이지 - brain04/ReadCookies.jsp

```

1 <%@page contentType="text/html; charset=euc-kr"%>
2 <HTML>
3     <HEAD><TITLE>쿠키 데이터 읽기</TITLE></HEAD>
4     <BODY>
5         이름: ${cookie.NAME.value} <BR>
6         성별: ${cookie.GENDER.value} <BR>
7         나이: ${cookie.AGE.value}
8     </BODY>
9 </HTML>

```

연습문제 7-2 EL 함수 만들기

날짜와 시각을 출력하는 정적 메서드는 다음과 같이 작성하면 됩니다.

날짜와 시각을 출력하는 정적 메서드를 포함하는 자바 클래스 - MyUtils.java

```

1 package util;
2 import java.util.GregorianCalendar;
3 public class MyUtils {

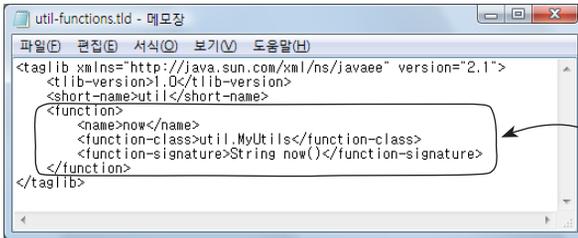
```

```

4     public static String now() {
5         GregorianCalendar now = new GregorianCalendar();
6         String str = String.format("%TF %TT", now, now);
7         return str;
8     }
9 }

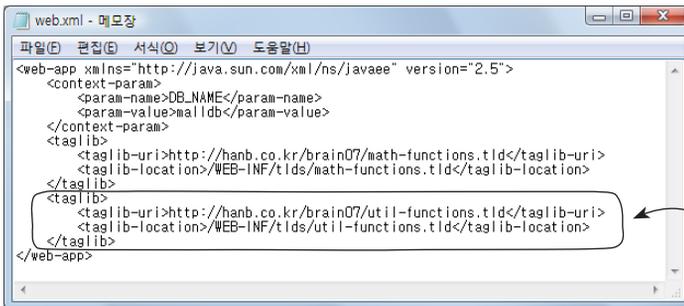
```

위 클래스의 정적 메서드 now를 EL 함수로 등록하려면 책의 본문에서 만들었던 brain07/WEB-INF/tlds/math-functions.tld 파일에 등록을 해도 되고, 다음과 같이 새로운 TLD 파일을 만들어서 등록해도 됩니다.



위 클래스의 now 메서드를 TLD 파일에 EL 함수로 등록하는 방법

그런데 이렇게 새로운 TLD 파일을 만들었을 경우에는 이 파일을 web.xml 파일에 등록해야 합니다. 위의 TLD 파일을 brain07/WEB-INF/tlds/util-functions.tld이라는 이름으로 저장했을 경우에는 brain07/WEB-INF/web.xml 파일에 이 TLD 파일을 다음과 같이 등록하면 됩니다.



위의 TLD 파일을 web.xml 파일에 등록하는 방법

연습문제 8-1 자바빈을 이용하는 표준 액션 연습

다음과 같이 작성하면 됩니다.

세션 데이터 영역에 자바빈을 저장하는 JSP 페이지 – brain08/SessionBeanWriter.jsp

```

1 <%@page contentType="text/html; charset=euc-kr"%>
2 <jsp:useBean class="mall.PersonalInfo" id="pinfo" scope="session" />

```

```

3 <jsp:setProperty name="pinfo" property="name" value="김미경" />
4 <jsp:setProperty name="pinfo" property="gender" value="여" />
5 <jsp:setProperty name="pinfo" property="age" value="20" />
6 <HTML>
7     <HEAD><TITLE>세션 데이터 영역에 자바빈 저장하기</TITLE></HEAD>
8     <BODY>
9         세션 데이터 영역에 자바빈이 저장되었습니다.
10    </BODY>
11 </HTML>

```

세션 데이터 영역에 있는 자바빈의 프로퍼티 값을 출력하는 JSP 페이지 - brain08/SessionBeanReader.jsp

```

1 <%@page contentType="text/html; charset=euc-kr"%>
2 <HTML>
3     <HEAD><TITLE>세션 데이터 영역의 자바빈 읽어오기</TITLE></HEAD>
4     <BODY>
5         <jsp:useBean class="mall.PersonalInfo" id="pinfo" scope="session" />
6         이름: <jsp:getProperty name="pinfo" property="name" /> <BR>
7         성별: <jsp:getProperty name="pinfo" property="gender" /> <BR>
8         나이: <jsp:getProperty name="pinfo" property="age" />
9     </BODY>
10 </HTML>

```

연습문제 10-1 변수를 지원하는 태그 클래스 작성 연습

다음과 같이 작성하면 됩니다.

<tool:compute> 커스텀 액션을 구현하는 태그 클래스 - ComputeTag.java

```

1 package tool;
2 import java.io.*;
3 import javax.servlet.jsp.*;
4 import javax.servlet.jsp.tagext.*;
5 public class ComputeTag extends SimpleTagSupport {
6     private String var;
7     private int start, end;
8     public void setVar(String var) {
9         this.var = var;
10    }
11    public void setStart(Integer start) {
12        this.start = start;
13    }

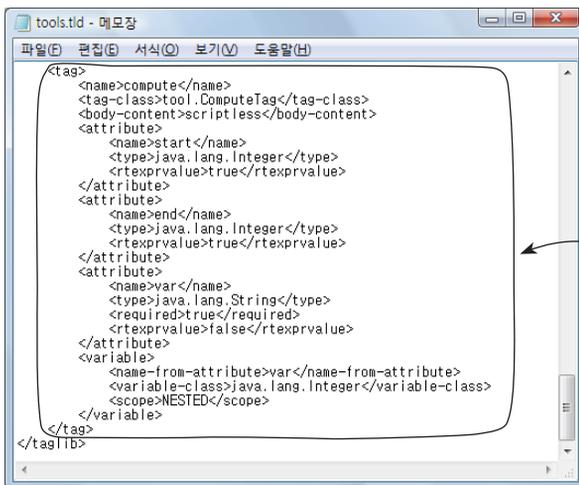
```

```

14     public void setEnd(Integer end) {
15         this.end = end;
16     }
17     public void doTag() throws JspException, IOException {
18         JspContext context = getJspContext();
19         JspWriter out = context.getOut();
20         JspFragment body = getJspBody();
21         for (int cnt = start; cnt <= end; cnt++) {
22             context.setAttribute(var, cnt);
23             body.invoke(out);
24         }
25         return;
26     }
27 }

```

위의 태그 클래스를 TLD 파일에 등록하기 위해서는 10장에서 만들었던 brain10/WEB-INF/tlds/tools.tld 파일에 다음과 같은 <tag> 엘리먼트를 추가해야 합니다.



위의 태그 클래스를 TLD 파일에 등록하는 방법

연습문제 11-1 필터 만들기 연습

[예제 11-8]을 다음과 같이 수정하면 됩니다.

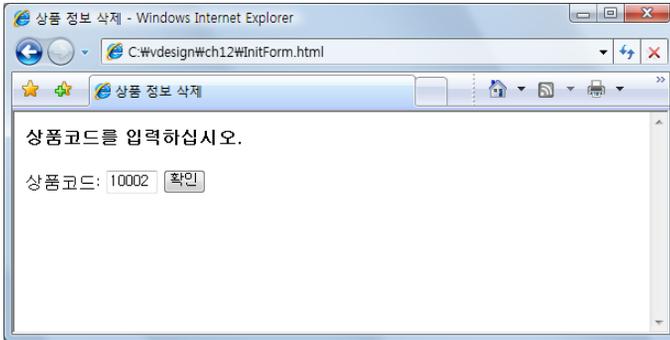
[예제 11-8] 유용한 정보를 포함한 로그 메시지를 기록하는 필터 클래스 - NewLogMessageFilter.java

```
1  package myfilter;
2  import javax.servlet.*;
3  import javax.servlet.http.*;
4  import java.io.*;
5  import java.util.*;
6  public class NewLogMessageFilter implements Filter {
7      PrintWriter writer;
8      public void init(FilterConfig config) throws ServletException {
9          String filename = config.getInitParameter("FILE_NAME") ;
10         if (filename == null)
11             throw new ServletException("로그 파일의 이름을 찾을 수 없습니다.");
12         try {
13             writer = new PrintWriter(new FileWriter(filename, true), true);
14         }
15         catch (IOException e) {
16             throw new ServletException("로그 파일을 열 수 없습니다.");
17         }
18     }
19     public void doFilter(ServletRequest request,
20                         ServletResponse response,
21                         FilterChain chain)
22         throws IOException, ServletException {
23         GregorianCalendar now = new GregorianCalendar();
24         writer.printf("현재일시: %TF %TT %n", now, now);
25         String clientAddr = request.getRemoteAddr();
26         writer.printf("클라이언트 주소: %s %n", clientAddr);
27         chain.doFilter(request, response);
28         String contentType = response.getContentType();
29         writer.printf("문서의 콘텐츠 타입: %s %n", contentType);
30         String browserType =
31             ((HttpServletRequest) request).getHeader("User-Agent");
32         writer.printf("웹 브라우저의 종류: %s %n", browserType);
33         writer.println("-----");
34     }
35     public void destroy() {
36         writer.close();
37     }
38 }
```

연습문제 12-1 데이터베이스에서 데이터 삭제하기

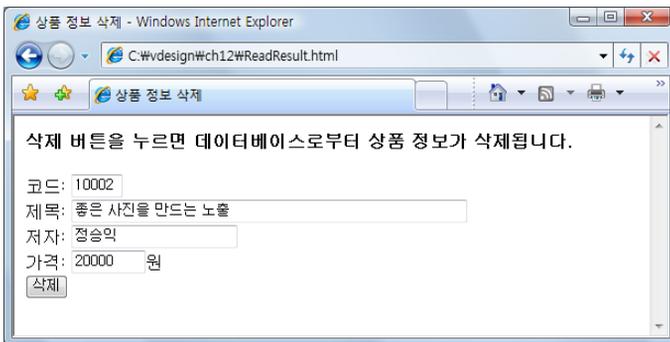
상품 정보 테이블에서 데이터를 삭제하는 웹 애플리케이션의 화면 설계는 다음과 같이 할 수 있습니다.

[초기 화면]



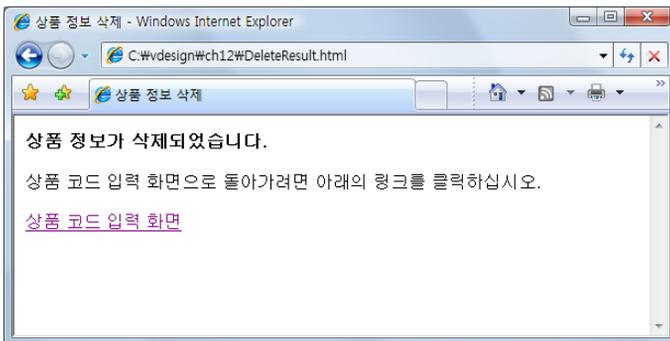
① 상품코드를 입력하고 “확인” 버튼을 누르면 현재 DB에 저장되어 있는 상품 정보가 화면에 나타납니다.

[상품 정보 조회 화면]

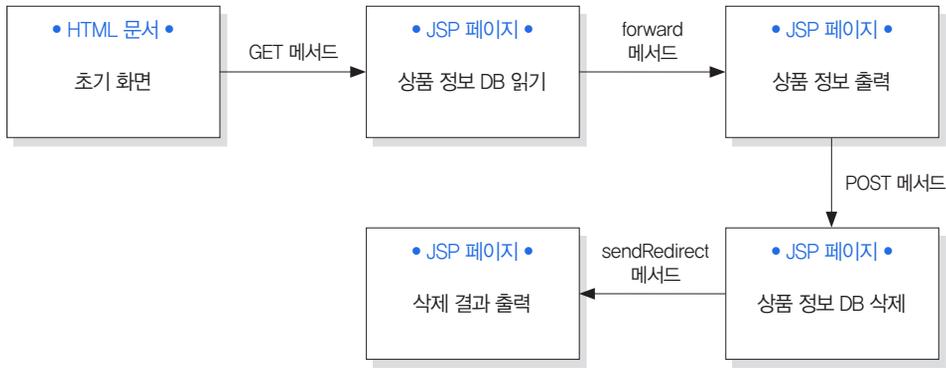


② “삭제” 버튼을 누르면 그 정보가 DB에서 삭제되고, 다음과 같은 결과 화면이 나타납니다.

[상품 정보 삭제 결과 화면]



위 그림의 첫 번째 화면은 HTML 문서로 만들 수 있지만, 두 번째와 세 번째 화면은 JSP 페이지로 구현해야 합니다. 그리고 두 번째 화면을 출력하기 위해서는 데이터베이스로부터 상품 정보를 읽어 와야 하고, 세 번째 화면을 출력하기 전에 데이터베이스로부터 상품 정보를 삭제해야 하는데, 이 두 기능은 각각에 해당하는 화면을 출력하는 JSP 페이지와 별도의 모듈로 구현하는 것이 좋습니다. 그러므로 이 애플리케이션은 다음과 같은 5개의 모듈로 구성하겠습니다.



위 그림에서 볼 수 있는 것처럼 이 애플리케이션을 구성하는 모듈은 다섯 개나 되기 때문에 brain12 웹 애플리케이션 디렉터리 아래에 GID라는 서브디렉터리를 만들고, 그 안에 이 다섯 개의 모듈을 모두 저장하겠습니다. 그리고 그 모듈 각각의 URL을 다음과 같이 정하겠습니다.

- `http://localhost:8080/brain12/GID/InitForm.html` ← 상품 코드 입력 화면 HTML 문서
- `http://localhost:8080/brain12/GID/Reader.jsp` ← 상품 정보 DB 읽기 JSP 페이지
- `http://localhost:8080/brain12/GID/ReadResult.jsp` ← 상품 정보 출력 JSP 페이지
- `http://localhost:8080/brain12/GID/Delete.jsp` ← 상품 정보 DB 삭제 JSP 페이지
- `http://localhost:8080/brain12/GID/DeleteResult.jsp` ← 상품 정보 삭제 결과 JSP 페이지

위에 있는 다섯 개의 URL에 해당하는 모듈은 다음과 같이 작성할 수 있습니다.

상품코드 입력 화면을 제공하는 HTML 문서 - brain12/GID/InitForm.html

```

1 <HTML>
2   <HEAD>
3     <META http-equiv="Content-Type" content="text/html; charset=euc-kr">
4     <TITLE>상품 정보 삭제</TITLE>
5   </HEAD>
6   <BODY>
7     <H4>상품코드를 입력하십시오.</H4>
8     <FORM ACTION=Reader.jsp METHOD=GET>
9       상품코드: <INPUT TYPE=TEXT NAME=code SIZE=5>
  
```

```

10         <INPUT TYPE=SUBMIT VALUE='확인'>
11     </FORM>
12 </BODY>
13 </HTML>

```

** 이 코드는 [예제 12-8]과 동일합니다.

DB로부터 상품 정보를 읽는 JSP 페이지 - brain12/GID/Reader.jsp

```

1  <%@page contentType="text/html; charset=euc-kr" errorPage="../DBError.jsp" %>
2  <%@page import="java.sql.*"%>
3  <%
4      String code = request.getParameter("code");
5      if (code == null)
6          throw new Exception("상품코드를 입력하십시오.");
7      Connection conn = null;
8      Statement stmt = null;
9      try {
10         Class.forName("com.mysql.jdbc.Driver");
11         conn = DriverManager.getConnection(
12             "jdbc:mysql://localhost:3306/webdb", "root", "1234");
13         if (conn == null)
14             throw new Exception("데이터베이스에 연결할 수 없습니다.");
15         stmt = conn.createStatement();
16         ResultSet rs = stmt.executeQuery(
17             "select * from goodsinfo where code = '" + code + "'");
18         if (!rs.next())
19             throw new Exception("상품코드(" + code + ")에 해당하는 데이터가 없습니다.");
20         String title = rs.getString("title");
21         String writer = rs.getString("writer");
22         int price = rs.getInt("price");
23         request.setAttribute("CODE", code);
24         request.setAttribute("TITLE", toUnicode(title));
25         request.setAttribute("WRITER", toUnicode(writer));
26         request.setAttribute("PRICE", new Integer(price));
27     }
28     finally {
29         try {
30             stmt.close();
31         }
32         catch (Exception ignored) {
33             try {
34                 conn.close();
35             }
36             catch (Exception ignored) {

```

```

36     }
37 }
38   RequestDispatcher dispatcher =
           request.getRequestDispatcher("ReadResult.jsp");
39   dispatcher.forward(request, response);
40 %>
41 <%!
42   private String toUnicode(String str) {
           // ISO-8859-1 문자열을 Unicode 문자열로 바꾸는 메서드
43       try {
44           byte[] b = str.getBytes("ISO-8859-1");
45           return new String(b);
46       }
47       catch (java.io.UnsupportedEncodingException uee) {
48           System.out.println(uee.getMessage());
49           return null;
50       }
51   }
52 %>

```

** 11행: “1234”라고 쓴 위치에는 여러분이 MySQL을 설치할 때 입력한 root 패스워드를 대신 써넣어야 합니다.

** 이 코드는 붉은 글씨로 표시한 부분을 제외하고는 [예제 12-9]와 동일합니다.

상품 정보를 출력하는 화면을 제공하는 JSP 페이지 – brain12/Gid/ReadResult.jsp

```

1  <%@page contentType="text/html; charset=euc-kr" %>
2  <HTML>
3      <HEAD>
4          <TITLE>상품 정보 삭제</TITLE>
5      </HEAD>
6      <BODY>
7          <H4>삭제 버튼을 누르면 데이터베이스로부터 상품 정보가 삭제됩니다.</H4>
8          <FORM ACTION=Deleter.jsp METHOD=POST>
9              코드: <INPUT TYPE=TEXT NAME=code SIZE=5 VALUE='${CODE}'
                                READONLY=TRUE> <BR>
10             제목: <INPUT TYPE=TEXT NAME=title SIZE=50 VALUE='${TITLE}'
                                READONLY=TRUE> <BR>
11             저자: <INPUT TYPE=TEXT NAME=writer SIZE=20 VALUE='${WRITER}'
                                READONLY=TRUE> <BR>
12             가격: <INPUT TYPE=TEXT NAME=price SIZE=8 VALUE='${PRICE}'
                                READONLY=TRUE>원 <BR>
13             <INPUT TYPE=SUBMIT VALUE='삭제'>
14         </FORM>
15     </BODY>
16 </HTML>

```

```

1  <%@page contentType="text/html; charset=euc-kr" errorPage="../DBError.jsp" %>
2  <%@page import="java.sql.*"%>
3  <%
4      String code = request.getParameter("code");
5      if (code == null)
6          throw new Exception("상품코드를 입력하십시오.");
7      Connection conn = null;
8      Statement stmt = null;
9      try {
10         Class.forName("com.mysql.jdbc.Driver");
11         conn = DriverManager.getConnection(
12             "jdbc:mysql://localhost:3306/webdb", "root", "1234");
13         if (conn == null)
14             throw new Exception("데이터베이스에 연결할 수 없습니다.");
15         stmt = conn.createStatement();
16         String command = String.format(
17             "delete from goodsinfo where code = '%s'", code);
18         int rowNum = stmt.executeUpdate(command);
19         if (rowNum < 1)
20             throw new Exception("데이터를 DB에서 삭제할 수 없습니다.");
21     }
22     finally {
23         try {
24             stmt.close();
25         }
26         catch (Exception ignored) {
27             }
28         try {
29             conn.close();
30         }
31         catch (Exception ignored) {
32             }
33     }
34     response.sendRedirect("DeleteResult.jsp?code=" + code);
35 %>

```

** 11행. "1234"라고 쓴 위치에는 여러분이 MySQL을 설치할 때 입력한 root 패스워드를 대신 써넣어야 합니다.

```

1  <%@page contentType="text/html; charset=euc-kr"%>
2  <HTML>
3      <HEAD><TITLE>상품 정보 삭제</TITLE></HEAD>
4      <BODY>

```

```

5         <H4>상품 정보가 삭제되었습니다.</H4>
6         상품 코드 입력 화면으로 돌아가려면 아래의 링크를 클릭하십시오.<BR><BR>
7         <A HREF=InitForm.html>상품 코드 입력 화면</A>
8     </BODY>
9 </HTML>

```

연습문제 12-2 데이터베이스 커넥션 풀 사용하기

[예제 12-5]와 [예제 12-7]은 수정할 필요가 없으며, [예제 12-6]만 다음과 같이 수정하면 됩니다.

[예제 12-6] 회원 정보를 데이터베이스에 입력하는 JSP 페이지 - brain12/Subscription.jsp

```

1 <%@page contentType="text/html; charset=euc-kr" errorPage="DBError.jsp" %>
2 <%@page import="java.sql.*"%>
3 <%
4     String name = request.getParameter("name");
5     String id = request.getParameter("id");
6     String password = request.getParameter("password");
7     if (name == null || id == null || password == null)
8         throw new Exception("데이터를 입력하십시오.");
9     Connection conn = null;
10    Statement stmt = null;
11    try {
12        Class.forName("org.apache.commons.dbcp.PoolingDriver");
13        conn = DriverManager.getConnection(
14            "jdbc:apache:commons:dbcp:/webdb_pool");
15        if (conn == null)
16            throw new Exception("데이터베이스에 연결할 수 없습니다.");
17        stmt = conn.createStatement();
18        String command = String.format("insert into userinfo " +
19            "(name, id, password) values ('%s', '%s', '%s');",
20            name, id, password);
21        int rowNum = stmt.executeUpdate(command);
22        if (rowNum < 1)
23            throw new Exception("데이터를 DB에 입력할 수 없습니다.");
24    }
25    finally {
26        try {
27            stmt.close();
28        }
29        catch (Exception ignored) {
30        }
31        try {
32            conn.close();
33        }
34    }
35 }

```

```

31         catch (Exception ignored) {
32             }
33     }
34     response.sendRedirect("SubscriptionResult.jsp");
35 %>

```

연습문제 13-1 모델 2 프로그래밍 연습

[예제 13-5], [예제 13-6], [예제 13-7]을 다음과 같이 수정하면 됩니다.

[예제 13-5] 게시물 목록 데이터를 표현하는 자바빈 클래스 - BBSList.java

```

1  package web;
2  import java.io.*;
3  import java.sql.*;
4  import java.util.ArrayList;
5  public class BBSList {
6      private ArrayList<Integer> seqNoList = new ArrayList<Integer>(); // 순번
7      private ArrayList<String> titleList = new ArrayList<String>(); // 제목
8      private ArrayList<String> writerList = new ArrayList<String>(); // 작성자
9      private ArrayList<Date> dateList = new ArrayList<Date>(); // 저장일자
10     private ArrayList<Time> timeList = new ArrayList<Time>(); // 저장시각
11     private boolean firstPage = false; // 게시물 목록의 첫번째 페이지인지 여부
12     private boolean lastPage = false; // 게시물 목록의 마지막 페이지인지 여부
13     private int pageNum; // 게시물 목록 페이지의 수
14     public BBSList() {
15     }
16     public void setSeqNo(int index, Integer seqNo) {
17         this.seqNoList.add(index, seqNo);
18     }
19     public void setTitle(int index, String title) {
20         this.titleList.add(index, title);
21     }
22     public void setWriter(int index, String writer) {
23         this.writerList.add(index, writer);
24     }
25     public void setDate(int index, Date date) {
26         this.dateList.add(index, date);
27     }
28     public void setTime(int index, Time time) {
29         this.timeList.add(index, time);
30     }
31     public void setFirstPage(boolean firstPage) {
32         this.firstPage = firstPage;
33     }

```

```

34     public void setLastPage(boolean lastPage) {
35         this.lastPage = lastPage;
36     }
37     public Integer[] getSeqNo() {
38         return seqNoList.toArray(new Integer[seqNoList.size()]);
39     }
40     public String[] getTitle() {
41         return titleList.toArray(new String[titleList.size()]);
42     }
43     public String[] getWriter() {
44         return writerList.toArray(new String[writerList.size()]);
45     }
46     public Date[] getDate() {
47         return dateList.toArray(new Date[dateList.size()]);
48     }
49     public Time[] getTime() {
50         return timeList.toArray(new Time[timeList.size()]);
51     }
52     public boolean isFirstPage() {
53         return firstPage;
54     }
55     public boolean isLastPage() {
56         return lastPage;
57     }
58     public int getListSize() {           // 게시글의 수를 리턴하는 메서드
59         return seqNoList.size();
60     }
61     public void setPageNum(int pageNum) { // 페이지의 수를 설정하는 메서드
62         this.pageNum = pageNum;
63     }
64     public int getPageNum() {           // 페이지의 수를 리턴하는 메서드
65         return pageNum;
66     }
67 }

```

[예제 13-6] 데이터베이스로부터 게시글 정보를 읽어서 게시글 목록을 만드는 서블릿 클래스 - brain13/bbs-list

```

1     package web;
2     import javax.servlet.http.*;
3     import javax.servlet.*;
4     import java.io.*;
5     import java.sql.*;
6     public class BBSListServlet extends HttpServlet {
7         public void doGet(HttpServletRequest request,
8                             HttpServletResponse response)
9             throws IOException, ServletException {

```

```

8      String strFirstSeqNo = request.getParameter("FIRST_SEQ_NO");
9      String strLastSeqNo = request.getParameter("LAST_SEQ_NO");
10     String strPageNo = request.getParameter("PAGE_NO");
11     BBSList list;
12     if (strFirstSeqNo != null) {
13         list = readPrevPage(Integer.parseInt(strFirstSeqNo));
14     }
15     else if (strLastSeqNo != null) {
16         list = readNextPage(Integer.parseInt(strLastSeqNo));
17     }
18     else if (strPageNo != null) {
19         list = readPage(Integer.parseInt(strPageNo));
20     }
21     else {
22         list = readNextPage(Integer.MAX_VALUE);
23         list.setFirstPage(true);
24     }
25     list.setPageNum(readPageNum());
26     request.setAttribute("BBS_LIST", list);
27     RequestDispatcher dispatcher = request.getRequestDispatcher(
28         "BBSListView.jsp");
29     dispatcher.forward(request, response);
30 }
31 private BBSList readPrevPage(int lowerSeqNo) throws ServletException {
32     // DB로부터 이전 페이지를 읽는 메서드
33     BBSList list = new BBSList();
34     Connection conn = null;
35     Statement stmt = null;
36     try {
37         Class.forName("org.apache.commons.dbcp.PoolingDriver");
38         conn = DriverManager.getConnection(
39             "jdbc:apache:commons:dbcp:/wdbpool");
40         if (conn == null)
41             throw new Exception("데이터베이스에 연결할 수 없습니다.");
42         stmt = conn.createStatement();
43         ResultSet rs = stmt.executeQuery(
44             "select * from bbs where seqNo > " + lowerSeqNo +
45             " order by seqno asc;");
46         for (int cnt = 0; cnt < 5; cnt++) {
47             if (!rs.next())
48                 break;
49             list.setSeqNo(0, rs.getInt("seqNo"));
50             list.setTitle(0, toUnicode(rs.getString("title")));
51             list.setWriter(0, toUnicode(rs.getString("writer")));
52             list.setDate(0, rs.getDate("wdate"));
53             list.setTime(0, rs.getTime("wtime"));
54         }
55     } catch (Exception e) {
56         e.printStackTrace();
57     } finally {
58         if (stmt != null) stmt.close();
59         if (conn != null) conn.close();
60     }
61     return list;
62 }

```

```

51         list.setFirstPage(true);
52     }
53     catch (Exception e) {
54         throw new ServletException(e);
55     }
56     finally {
57         try {
58             stmt.close();
59         }
60         catch (Exception ignored) {
61         }
62         try {
63             conn.close();
64         }
65         catch (Exception ignored) {
66         }
67     }
68     return list;
69 }
70 private BBSList readNextPage(int upperSeqNo) throws ServletException {
71     // DB로부터 다음 페이지를 읽는 메서드
72     BBSList list = new BBSList();
73     Connection conn = null;
74     Statement stmt = null;
75     try {
76         Class.forName("org.apache.commons.dbcp.PoolingDriver");
77         conn = DriverManager.getConnection(
78             "jdbc:apache:commons:dbcp:/wdbpool");
79         if (conn == null)
80             throw new Exception("데이터베이스에 연결할 수 없습니다.");
81         stmt = conn.createStatement();
82         ResultSet rs = stmt.executeQuery(
83             "select * from bbs where seqNo < " + upperSeqNo +
84             " order by seqno desc;");
85         for (int cnt = 0; cnt < 5; cnt++) {
86             if (!rs.next())
87                 break;
88             list.setSeqNo(cnt, rs.getInt("seqNo"));
89             list.setTitle(cnt, toUnicode(rs.getString("title")));
90             list.setWriter(cnt, toUnicode(rs.getString("writer")));
91             list.setDate(cnt, rs.getDate("wdate"));
92             list.setTime(cnt, rs.getTime("wtime"));
93         }
94         if (!rs.next())
95             list.setLastPage(true);
96     }
97     catch (Exception e) {
98         throw new ServletException(e);
99     }

```

```

95     }
96     finally {
97         try {
98             stmt.close();
99         }
100        catch (Exception ignored) {
101        }
102        try {
103            conn.close();
104        }
105        catch (Exception ignored) {
106        }
107    }
108    return list;
109 }
110 private BBSList readPage(int pageNo) throws ServletException {
111     // DB로부터 특정 페이지를 읽는 메서드
112     BBSList list = new BBSList();
113     Connection conn = null;
114     Statement stmt = null;
115     try {
116         Class.forName("org.apache.commons.dbcp.PoolingDriver");
117         conn = DriverManager.getConnection(
118             "jdbc:apache:commons:dbcp:/wdbpool");
119         if (conn == null)
120             throw new Exception("데이터베이스에 연결할 수 없습니다.");
121         stmt = conn.createStatement();
122         ResultSet rs = stmt.executeQuery(
123             "select * from bbs order by seqno desc;");
124         if (pageNo > 1)
125             rs.absolute((pageNo - 1) * 5);
126         for (int cnt = 0; cnt < 5; cnt++) {
127             if (!rs.next())
128                 break;
129             list.setSeqNo(cnt, rs.getInt("seqNo"));
130             list.setTitle(cnt, toUnicode(rs.getString("title")));
131             list.setWriter(cnt, toUnicode(rs.getString("writer")));
132             list.setDate(cnt, rs.getDate("wdate"));
133             list.setTime(cnt, rs.getTime("wtime"));
134         }
135         if (pageNo == 1)
136             list.setFirstPage(true);
137         if (!rs.next())
138             list.setLastPage(true);
139     }
140     catch (Exception e) {
141         throw new ServletException(e);
142     }

```

```

140     finally {
141         try {
142             stmt.close();
143         }
144         catch (Exception ignored) {
145         }
146         try {
147             conn.close();
148         }
149         catch (Exception ignored) {
150         }
151     }
152     return list;
153 }
154 private int readPageNum() throws ServletException {
155     // DB로부터 페이지 수를 읽는 메서드
156     int pageNum;
157     Connection conn = null;
158     Statement stmt = null;
159     try {
160         Class.forName("org.apache.commons.dbcp.PoolingDriver");
161         conn = DriverManager.getConnection(
162             "jdbc:apache:commons:dbcp:/wdbpool");
163         if (conn == null)
164             throw new Exception("데이터베이스에 연결할 수 없습니다.");
165         stmt = conn.createStatement();
166         ResultSet rs = stmt.executeQuery(
167             "select count(*) as NUM from bbs");
168         if (!rs.next())
169             return 0;
170         pageNum = rs.getInt("NUM");
171     }
172     catch (Exception e) {
173         throw new ServletException(e);
174     }
175     finally {
176         try {
177             stmt.close();
178         }
179         catch (Exception ignored) {
180         }
181         try {
182             conn.close();
183         }
184         catch (Exception ignored) {
185         }
186     }
187     return (pageNum + 4) / 5;

```

```

185     }
186     private String toUnicode(String str) {
187         // ISO-8859-1 문자열을 Unicode 문자열로 바꾸는 메서드
188         if (str == null)
189             return null;
190         try {
191             byte[] b = str.getBytes("ISO-8859-1");
192             return new String(b);
193         } catch (java.io.UnsupportedEncodingException uee) {
194             System.out.println(uee.getMessage());
195             return null;
196         }
197     }
198 }

```

**** 122행:** absolute 메서드는 DB로부터 읽은 데이터 등에서 파라미터로 두어진 인덱스에 해당하는 행으로 바로 이동하는 기능을 합니다. 그리고 이 인덱스 값은 0부터 시작하므로 rs.absolute(5);라는 메서드 호출문은 데이터베이스로부터 읽은 데이터 중에서 6번째 행으로 이동하는 기능을 합니다.

[예제 13-7] 게시물 목록을 출력하는 JSP 페이지 - brain11/BBSListView.jsp

```

1  <%@page contentType="text/html; charset=euc-kr"%>
2  <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3  <HTML>
4      <HEAD><TITLE>게시판</TITLE></HEAD>
5      <BODY>
6          <H4>게시판 목록 보기</H4>
7          <TABLE border=1>
8              <TR>
9                  <TD width=40>순번</TD>
10                 <TD width=300>제목</TD>
11                 <TD width=80>작성자</TD>
12                 <TD width=90>작성일자</TD>
13                 <TD width=70>작성시각</TD>
14             </TR>
15             <c:forEach var="cnt" begin="0" end="{BBS_LIST.listSize - 1}">
16                 <TR>
17                     <TD>${BBS_LIST.seqNo[cnt]}</TD>
18                     <TD>${BBS_LIST.title[cnt]}</TD>
19                     <TD>${BBS_LIST.writer[cnt]}</TD>
20                     <TD>${BBS_LIST.date[cnt]}</TD>
21                     <TD>${BBS_LIST.time[cnt]}</TD>
22                 </TR>
23             </c:forEach>
24         </TABLE>

```

```

25         <c:if test="${!BBS_LIST.firstPage}">
26             <A href='bbs-list?FIRST_SEQ_NO=${BBS_LIST.seqNo[0]}'>이전 페이지</A>
27         </c:if>
28         <c:forEach var="cnt" begin="1" end="${BBS_LIST.pageNum}">
29             <A href='bbs-list?PAGE_NO=${cnt}'>${cnt}</A>
30         </c:forEach>
31         <c:if test="${!BBS_LIST.lastPage}">
32             <A href='bbs-list?LAST_SEQ_NO=${BBS_LIST.seqNo[
33                 BBS_LIST.listSize - 1]}'>다음 페이지</A>
34         </c:if>
35     </BODY>
36 </HTML>

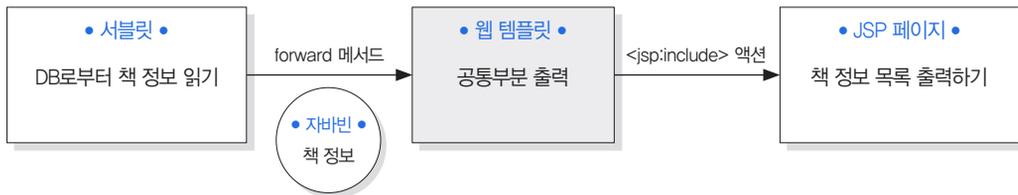
```

연습문제 13-2 웹 템플릿 활용 연습(1)

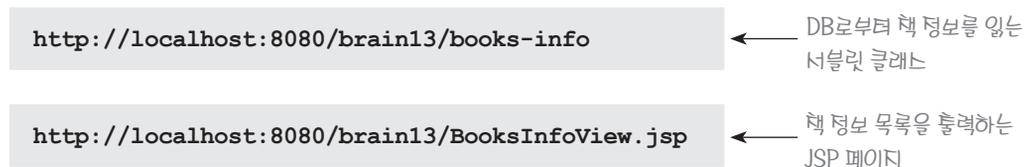
모델 2를 이용할 경우에는 이 애플리케이션을 다음과 같은 세 개의 모듈로 구성할 수 있습니다.

- DB로부터 책 정보를 읽는 서블릿 클래스
- 책 정보를 표현하는 자바빈 클래스
- 책 정보를 출력하는 JSP 페이지

그리고 이 세 모듈이 웹 템플릿과 함께 작동하도록 만들기 위해서는 웹 템플릿을 다음과 같이 서블릿 클래스와 JSP 페이지의 사이에 두어야 합니다.



이 세 모듈 중 자바빈 클래스의 이름은 BooksInfo라고 정하고, 서블릿 클래스와 JSP 페이지의 URL은 다음과 같이 정하겠습니다.



지금 언급한 세 개의 모듈은 다음과 같이 작성할 수 있습니다.

책 정보 목록을 표현하는 자바빈 클래스 - BooksInfo.java

```
1 package web;
2 import java.io.*;
3 import java.sql.*;
4 import java.util.ArrayList;
5 public class BooksInfo {
6     private ArrayList<String> codeList = new ArrayList<String>(); // 상품코드
7     private ArrayList<String> titleList = new ArrayList<String>(); // 제목
8     private ArrayList<String> writerList = new ArrayList<String>(); // 저자
9     private ArrayList<Integer> priceList = new ArrayList<Integer>(); // 가격
10    private boolean firstPage = false; // 책 정보 목록의 첫번째 페이지인지 여부
11    private boolean lastPage = false; // 책 정보 목록의 마지막 페이지인지 여부
12    public BooksInfo() {
13    }
14    public void setCode(int index, String code) {
15        this.codeList.add(index, code);
16    }
17    public void setTitle(int index, String title) {
18        this.titleList.add(index, title);
19    }
20    public void setWriter(int index, String writer) {
21        this.writerList.add(index, writer);
22    }
23    public void setPrice(int index, Integer price) {
24        this.priceList.add(index, price);
25    }
26    public void setFirstPage(boolean firstPage) {
27        this.firstPage = firstPage;
28    }
29    public void setLastPage(boolean lastPage) {
30        this.lastPage = lastPage;
31    }
32    public String[] getCode() {
33        return codeList.toArray(new String[codeList.size()]);
34    }
35    public String[] getTitle() {
36        return titleList.toArray(new String[titleList.size()]);
37    }
38    public String[] getWriter() {
39        return writerList.toArray(new String[writerList.size()]);
40    }
41    public Integer[] getPrice() {
42        return priceList.toArray(new Integer[priceList.size()]);
43    }
}
```

```

44     public boolean isFirstPage() {
45         return firstPage;
46     }
47     public boolean isLastPage() {
48         return lastPage;
49     }
50     public int getSize() {          // 책 정보의 수를 리턴하는 메서드
51         return codeList.size();
52     }
53 }

```

데이터베이스로부터 책 정보를 읽는 서블릿 클래스 - brain13/books-info

```

1  package web;
2  import javax.servlet.http.*;
3  import javax.servlet.*;
4  import java.io.*;
5  import java.sql.*;
6  public class BooksInfoServlet extends HttpServlet {
7      public void doGet(HttpServletRequest request,
8                          HttpServletResponse response)
9                          throws IOException, ServletException {
10         String firstCode = request.getParameter("FIRST_CODE");
11         String lastCode = request.getParameter("LAST_CODE");
12         BooksInfo booksInfo;
13         if (firstCode != null) {
14             booksInfo = readPrevPage(firstCode);
15         }
16         else if (lastCode != null) {
17             booksInfo = readNextPage(lastCode);
18         }
19         else {
20             booksInfo = readNextPage("00000");
21             booksInfo.setFirstPage(true);
22         }
23         request.setAttribute("BOOKS_INFO", booksInfo);
24         RequestDispatcher dispatcher = request.getRequestDispatcher(
25             "WebTemplate.jsp?BODY_PATH=BooksInfoView.jsp");
26         dispatcher.forward(request, response);
27     }
28     private BooksInfo readPrevPage(String firstCode) throws ServletException {
29         // DB로부터 이전 페이지를 읽는 메서드
30
31         BooksInfo booksInfo = new BooksInfo();
32         Connection conn = null;
33         Statement stmt = null;
34         try {

```

```

30         Class.forName("org.apache.commons.dbcp.PoolingDriver");
31         conn = DriverManager.getConnection(
32             "jdbc:apache:commons:dbcp:/wdbpool");
33         if (conn == null)
34             throw new Exception("데이터베이스에 연결할 수 없습니다.");
35         stmt = conn.createStatement();
36         ResultSet rs = stmt.executeQuery("select * " +
37             "from goodsinfo where code < '" + firstCode +
38             "' order by code desc;");
39         for (int cnt = 0; cnt < 5; cnt++) {
40             if (!rs.next())
41                 break;
42             booksInfo.setCode(0, rs.getString("code"));
43             booksInfo.setTitle(0, toUnicode(rs.getString("title")));
44             booksInfo.setWriter(0, toUnicode(rs.getString("writer")));
45             booksInfo.setPrice(0, rs.getInt("price"));
46         }
47         if (!rs.next())
48             booksInfo.setFirstPage(true);
49     }
50     catch (Exception e) {
51         throw new ServletException(e);
52     }
53     finally {
54         try {
55             stmt.close();
56         }
57         catch (Exception ignored) {
58             }
59         try {
60             conn.close();
61         }
62         catch (Exception ignored) {
63             }
64     }
65     return booksInfo;
66 }
67 private BooksInfo readNextPage(String lastCode) throws ServletException {
68     // DB로부터 다음 페이지를 읽는 메서드
69     BooksInfo booksInfo = new BooksInfo();
70     Connection conn = null;
71     Statement stmt = null;
72     try {
73         Class.forName("org.apache.commons.dbcp.PoolingDriver");
74         conn = DriverManager.getConnection(
75             "jdbc:apache:commons:dbcp:/wdbpool");
76         if (conn == null)
77             throw new Exception("데이터베이스에 연결할 수 없습니다.");

```

```

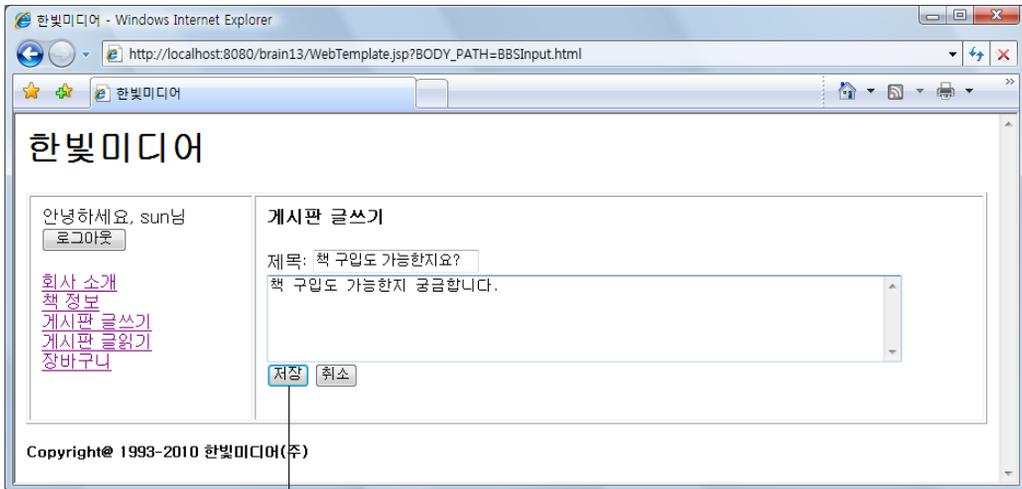
73         stmt = conn.createStatement();
74         ResultSet rs = stmt.executeQuery("select * " +
            "from goodsinfo where code > '" + lastCode +
            "'" + " order by code asc;");
75         for (int cnt = 0; cnt < 5; cnt++) {
76             if (!rs.next())
77                 break;
78             booksInfo.setCode(cnt, rs.getString("code"));
79             booksInfo.setTitle(cnt, toUnicode(rs.getString("title")));
80             booksInfo.setWriter(cnt, toUnicode(rs.getString("writer")));
81             booksInfo.setPrice(cnt, rs.getInt("price"));
82         }
83         if (!rs.next())
84             booksInfo.setLastPage(true);
85     }
86     catch (Exception e) {
87         throw new ServletException(e);
88     }
89     finally {
90         try {
91             stmt.close();
92         }
93         catch (Exception ignored) {
94         }
95         try {
96             conn.close();
97         }
98         catch (Exception ignored) {
99         }
100    }
101    return booksInfo;
102 }
103 private String toUnicode(String str) {
            // ISO-8859-1 문자열을 Unicode 문자열로 바꾸는 메서드
104     if (str == null)
105         return null;
106     try {
107         byte[] b = str.getBytes("ISO-8859-1");
108         return new String(b);
109     }
110     catch (java.io.UnsupportedEncodingException uee) {
111         System.out.println(uee.getMessage());
112         return null;
113     }
114 }
115 }

```

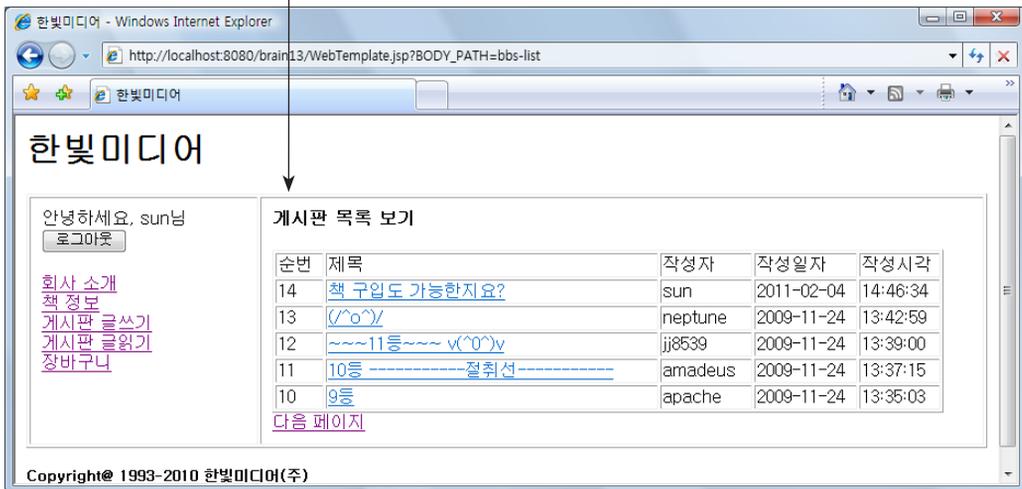
```
1 <%@page contentType="text/html; charset=euc-kr"%>
2 <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <H4>책 정보</H4>
4 <TABLE border=1>
5     <TR>
6         <TD width=70>상품코드</TD>
7         <TD width=250>제목</TD>
8         <TD width=80>저자</TD>
9         <TD width=80>가격</TD>
10    </TR>
11    <c:forEach var="cnt" begin="0" end="{BOOKS_INFO.size - 1}">
12        <TR>
13            <TD>${BOOKS_INFO.code[cnt]}</TD>
14            <TD>${BOOKS_INFO.title[cnt]}</TD>
15            <TD>${BOOKS_INFO.writer[cnt]}</TD>
16            <TD>${BOOKS_INFO.price[cnt]}원</TD>
17        </TR>
18    </c:forEach>
19 </TABLE>
20 <c:if test="{!BOOKS_INFO.firstPage}">
21     <A href='books-info?FIRST_CODE=${BOOKS_INFO.code[0]}'>이전 페이지</A>
22 </c:if>
23 <c:if test="{!BOOKS_INFO.lastPage}">
24     <A href='books-info?LAST_CODE=${
25         BOOKS_INFO.code[BOOKS_INFO.size - 1]}'>다음 페이지</A>
26 </c:if>
```

연습문제 13-3 웹 템플릿 활용 연습(2)

이 애플리케이션은 다음과 같이 작동하도록 구현할 수 있습니다.



게시글의 제목과 내용을 입력하고 “저장” 버튼을 누르면 해당 내용이 DB에 저장된 후에 게시판 목록 보기 화면이 나타납니다.



이런 애플리케이션은 다음과 같은 3개의 모듈로 구성할 수 있습니다.

- 게시글 입력 화면을 제공하는 HTML 문서
- 게시글을 DB에 저장하는 서블릿 클래스
- 게시글 목록을 제공하는 JSP 페이지

이 중 첫 번째와 세 번째 모듈에 해당하는 코드는 책의 본문에서 이미 작성했으므로 다시 작성할 필요가 없고, 두 번째 모듈인 서블릿 클래스만 다음과 같이 새로 작성하면 됩니다.

게시글을 데이터베이스에 입력하는 서블릿 클래스 - brain13/bbs-post

```
1 package web;
2 import javax.servlet.http.*;
3 import javax.servlet.*;
4 import java.io.*;
5 import java.sql.*;
6 import java.util.*;
7 public class BBSPostServlet extends HttpServlet {
8     public void doPost(HttpServletRequest request,
9                         HttpServletResponse response)
10                        throws IOException, ServletException {
11         HttpSession session = request.getSession();
12         String id = (String) session.getAttribute("LOGIN_ID");
13         if (id == null)
14             throw new ServletException("게시글을 등록하려면 먼저 로그인을 해야 합니다.");
15         String title = request.getParameter("TITLE");
16         String content = request.getParameter("CONTENT");
17         if (title == null || content == null)
18             throw new ServletException("데이터를 입력하십시오.");
19         int seqNo = 1;
20         GregorianCalendar now = new GregorianCalendar();
21         Connection conn = null;
22         Statement stmt = null;
23         try {
24             Class.forName("org.apache.commons.dbcp.PoolingDriver");
25             conn = DriverManager.getConnection(
26                 "jdbc:apache:commons:dbcp:/wdbpool");
27             if (conn == null)
28                 throw new ServletException("데이터베이스에 연결할 수 없습니다.");
29             stmt = conn.createStatement();
30             ResultSet rs = stmt.executeQuery(
31                 "select max(seqno) as max_seqno from bbs");
32             if (rs.next())
33                 seqNo = rs.getInt("max_seqno") + 1;
34             String command = String.format("insert into bbs " +
35                 "(seqNo, title, content, writer, wdate, wtime) values " +
36                 "(%d, '%s', '%s', '%s', '%TF', '%TT');",
37                 seqNo, title, content, id, now, now);
38             int rowNum = stmt.executeUpdate(command);
39             if (rowNum < 1)
40                 throw new ServletException("데이터를 DB에 입력할 수 없습니다.");
41         }
42     }
43 }
```

```

35         catch (ClassNotFoundException cnfe) {
36             throw new ServletException(cnfe);
37         }
38         catch (SQLException se) {
39             throw new ServletException(se);
40         }
41         finally {
42             try {
43                 stmt.close();
44             }
45             catch (Exception ignored) {
46             }
47             try {
48                 conn.close();
49             }
50             catch (Exception ignored) {
51             }
52         }
53         response.sendRedirect("bbs-list");
54     }
55 }

```

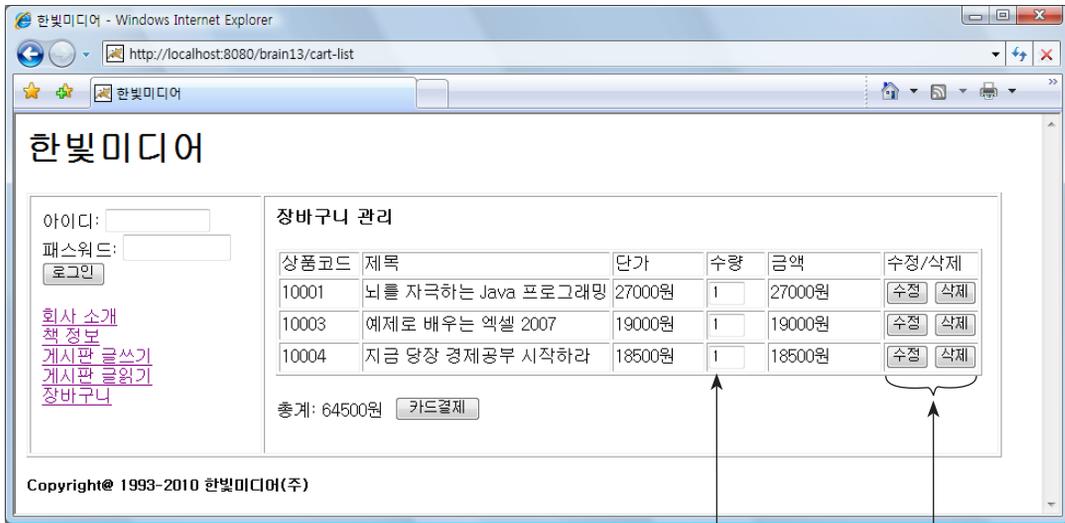
위의 서블릿 클래스를 컴파일한 다음에 그 결과를 brain13/WEB-INF/classes/web 디렉터리에 저장하세요. 그리고 나서 brain13/WEB-INF/web.xml 파일에 이 서블릿 클래스를 다음과 같이 등록하면 앞에서 보았던 그림과 같이 작동하는 게시판 글쓰기 기능이 구현됩니다.



위의 서블릿 클래스를 web.xml 파일에 등록하는 방법

연습문제 13-4 장바구니 관리 애플리케이션에 기능 추가하기

장바구니 관리 애플리케이션에 장바구니 항목의 수정 기능과 삭제 기능을 추가하기 위해서는 장바구니 관리 화면을 다음과 같이 수정해야 할 필요가 있습니다.



수량을 변경할 수 있도록
텍스트 상자 안에 표시합니다.

수정 버튼과 삭제 버튼을
추가합니다.

위 그림에서 장바구니 항목의 수를 변경하고 “수정” 버튼을 누르거나, 삭제를 원하는 항목의 “삭제” 버튼을 누르면 해당 항목이 수정/삭제된 다음에 이 화면이 다시 나타나야 합니다. 장바구니 관리 애플리케이션이 이런 식으로 작동하도록 만들려면 장바구니 목록을 출력하는 JSP 페이지를 수정하고, 장바구니 항목을 수정/삭제하는 서블릿 클래스를 새로 작성해야 합니다. 그리고 장바구니 데이터를 표현하는 클래스도 수정해야 할 필요가 있습니다.

장바구니 목록을 출력하는 JSP 페이지인 [예제 13-27]은 다음과 같이 수정하면 됩니다.

[예제 13-32] 장바구니 목록을 출력하는 JSP 페이지 - brain13/CartListView.jsp

```

1  <%@page contentType="text/html; charset=euc-kr"%>
2  <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3  <H4>장바구니 관리</H4>
4  <c:choose>
5      <c:when test="\${CART_LIST == null || CART_LIST.size <= 0}">
6          장바구니가 비어 있습니다.
7      </c:when>
8      <c:otherwise>
9          <TABLE border=1>
10             <TR>
11                 <TD width=70>상품코드</TD>
12                 <TD width=200>제목</TD>
13                 <TD width=80>단가</TD>
14                 <TD width=50>수량</TD>
15                 <TD width=100>금액</TD>

```

```

16         <TD width=70>수정/삭제</TD>
17     </TR>
18     <c:forEach var="cnt" begin="0" end="{CART_LIST.size - 1}">
19         <FORM ACTION=/brain13/modify-cart METHOD=POST>
20             <INPUT TYPE=HIDDEN NAME=CODE
21                 VALUE="{CART_LIST.code[cnt]}">
22             <TR>
23                 <TD>{CART_LIST.code[cnt]}</TD>
24                 <TD>{CART_LIST.title[cnt]}</TD>
25                 <TD>{CART_LIST.price[cnt]}원</TD>
26                 <TD><INPUT TYPE=TEXT SIZE=3 NAME=NUMBER
27                     VALUE="{CART_LIST.number[cnt]}"></TD>
28                 <TD>{CART_LIST.price[cnt] *
29                     CART_LIST.number[cnt]}원</TD>
30                 <TD><INPUT TYPE=SUBMIT NAME=MOD_DEL VALUE='수정'>
31                     <INPUT TYPE=SUBMIT NAME=MOD_DEL VALUE='삭제'></TD>
32             </TR>
33         </FORM>
34     </c:forEach>
35 </TABLE>
36 <FORM ACTION=/brain13/pay METHOD=POST>
37     총계: {CART_LIST.totalAmount}원
38     <INPUT TYPE=HIDDEN NAME=TOTAL_AMOUNT
39         VALUE="{CART_LIST.totalAmount}">
40     <INPUT TYPE=SUBMIT VALUE='카드결제'>
41 </FORM>
42 </c:otherwise>
43 </c:choose>

```

장바구니 데이터를 표현하는 클래스인 [예제 13-27]에는 장바구니 항목을 추가하는 기능은 있지만 장바구니 항목을 수정하고 삭제하는 기능은 없습니다. 그러므로 이 두 기능을 추가하기 위해 다음과 같은 두 개의 메서드를 추가하겠습니다.

[예제 13-27] 장바구니 데이터를 표현하는 클래스 - Cart.java

```

1 package web;
2 import java.util.LinkedList;
3 public class Cart {
4     private LinkedList<String> codeList = new LinkedList<String>(); // 상품코드
5     private LinkedList<Integer> numberList = new LinkedList<Integer>(); // 수량
6     public void addItem(String code, int num) { // 장바구니에 책 정보를 추가하는 메서드
7         for (int cnt = 0; cnt < codeList.size(); cnt++) {
8             if (codeList.get(cnt).equals(code)) {
9                 numberList.set(cnt, numberList.get(cnt) + 1);
10            }
11        }
12    }
13 }

```

```

11         }
12     }
13     codeList.add(code);
14     numberList.add(num);
15 }
16 public void modifyItem(String code, int num) {
17     // 장바구니의 책 정보를 수정하는 메서드
18     for (int cnt = 0; cnt < codeList.size(); cnt++) {
19         if (codeList.get(cnt).equals(code)) {
20             numberList.set(cnt, num);
21             return;
22         }
23     }
24 public void deleteItem(String code) { // 장바구니의 책 정보를 삭제하는 메서드
25     for (int cnt = 0; cnt < codeList.size(); cnt++) {
26         if (codeList.get(cnt).equals(code)) {
27             codeList.remove(cnt);
28             numberList.remove(cnt);
29             return;
30         }
31     }
32 }
33 public String getCode(int index) {
34     return codeList.get(index);
35 }
36 public int getNumber(int index) {
37     return numberList.get(index);
38 }
39 public int getSize() { // 장바구니에 있는 항목의 수를 리턴하는 메서드
40     return codeList.size();
41 }
42 }

```

마지막으로 장바구니 항목을 수정하고 삭제하는 기능의 서블릿 클래스는 다음과 같이 작성하면 됩니다.

장바구니의 항목을 수정/삭제하는 서블릿 클래스 - brain13/modify-cart

```

1 package web;
2 import javax.servlet.http.*;
3 import javax.servlet.*;
4 import java.io.*;
5 public class ModifyCartServlet extends HttpServlet {
6     public void doPost(HttpServletRequest request,
7                         HttpServletResponse response)
8         throws IOException, ServletException {

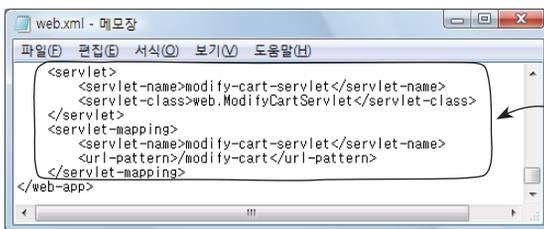
```

```

7         request.setCharacterEncoding("euc-kr");
8         String code = request.getParameter("CODE");
9         int number = Integer.parseInt(request.getParameter("NUMBER"));
10        String buttonName = request.getParameter("MOD_DEL");
11        HttpSession session = request.getSession();
12        Cart cart = (Cart) session.getAttribute("CART");
13        if (buttonName.equals("수정"))
14            cart.modifyItem(code, number);
15        else if (buttonName.equals("삭제"))
16            cart.deleteItem(code);
17        response.sendRedirect("cart-list");
18    }
19 }

```

위의 서블릿 클래스를 brain13/WEB-INF/web.xml 파일에 다음과 같은 방법으로 등록하면 앞에서 보았던 장바구니 관리 화면처럼 작동하는 장바구니 항목 수정/삭제 기능의 구현이 완성됩니다.



위의 서블릿 클래스를 web.xml 파일에 등록하는 방법