

# 통합구현

## 수행평가

---

날짜 : 2022/11/18

이름 : 공민혁

# 목차

---

1. 프로젝트 생성 및 구성화면

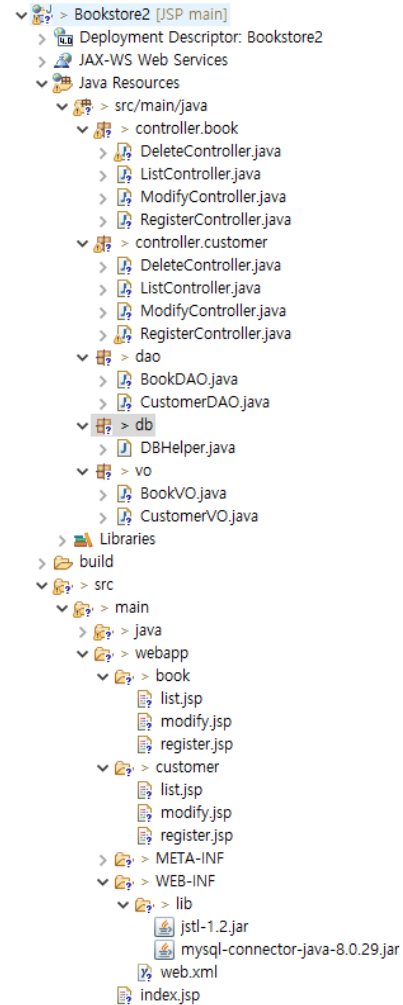
2. 화면구현

3.기능 구현

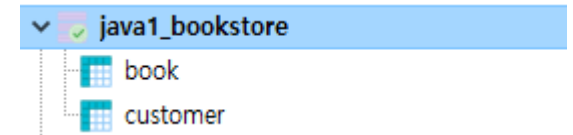
4.실행

# 1. 프로젝트 생성

- 프로젝트명 : Bookstore2
- WAS : Tomcat 9
- DB : java1\_bookstore
- 개발도구 : Eclipse, Workbench, HeidiSQL



프로젝트



DB

## 2. 화면 구현

### BookStore

[도서목록](#) [고객목록](#)

#### 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
2	축구이해함	미디어대한	50020	<a href="#">수정</a> <a href="#">삭제</a>

#### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
9	공고공	부산시 서울구	010-1234-2231	<a href="#">수정</a> <a href="#">삭제</a>

#### 도서등록

[처음으로 도서목록](#)

도서번호	<input type="text"/>
도서명	<input type="text"/>
출판사	<input type="text"/>
가격	<input type="text"/>
<input type="button" value="등록하기"/>	

#### 도서수정

[처음으로 도서목록](#)

도서번호	<input type="text" value="1"/>
도서명	<input type="text" value="축구의 역사"/>
출판사	<input type="text" value="굿스포츠"/>
가격	<input type="text" value="7000"/>
<input type="button" value="수정"/>	

#### 고객등록

[처음으로 고객목록](#)

고객명	<input type="text"/>
주소	<input type="text"/>
휴대폰	<input type="text"/>
<input type="button" value="등록하기"/>	

#### 고객수정

[처음으로 고객목록](#)

고객번호	<input type="text" value="9"/>
고객명	<input type="text" value="공고공"/>
주소	<input type="text" value="부산시 서울구"/>
휴대폰	<input type="text" value="010-1234-2231"/>
<input type="button" value="수정"/>	

### 3. 기능구현 - VO

```
package vo;

public class BookVO {

    private int bookId;
    private String bookName;
    private String pubName;
    private int price;

    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public void setBookId(String bookId) {
        this.bookId = Integer.parseInt(bookId);
    }
    public String getBookName() {
        return bookName;
    }
    public void setBookName(String bookName) {
        this.bookName = bookName;
    }
    public String getPubName() {
        return pubName;
    }
    public void setPubName(String pubName) {
        this.pubName = pubName;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
    public void setPrice(String price) {
        this.price = Integer.parseInt(price);
    }
}
```

```
package vo;

public class CustomerVO {

    private int cusId;
    private String cusName;
    private String cusAddr;
    private String cusHp;

    public int getCusId() {
        return cusId;
    }
    public void setCusId(int cusId) {
        this.cusId = cusId;
    }
    public void setCusId(String cusId) {
        this.cusId = Integer.parseInt(cusId);
    }
    public String getCusName() {
        return cusName;
    }
    public void setCusName(String cusName) {
        this.cusName = cusName;
    }
    public String getCusAddr() {
        return cusAddr;
    }
    public void setCusAddr(String cusAddr) {
        this.cusAddr = cusAddr;
    }
    public String getCusHp() {
        return cusHp;
    }
    public void setCusHp(String cusHp) {
        this.cusHp = cusHp;
    }
}
```

### 3. 기능구현 - DB, 커넥션풀

```
package db;

import java.sql.Connection;

public class DBHelper {

    protected Connection conn = null;
    protected PreparedStatement pstmt = null;
    protected Statement stmt = null;
    protected ResultSet rs = null;

    public Connection getConnection() {

        try {
            DataSource ds = (DataSource) new InitialContext().lookup("java:comp/env/dbcp_java1_bookstore");
            conn = ds.getConnection();

        } catch (Exception e) {
            e.printStackTrace();
        }
        return conn;
    }

    public void close() {
        try {
            if(rs != null) {
                rs.close();
            }
            if(stmt != null) {
                stmt.close();
            }
            if(pstmt != null) {
                pstmt.close();
            }
            if(conn != null) {
                conn.close();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

<Resource

```
name="dbcp_java1_bookstore"
auth="Container"
type="javax.sql.DataSource"
driverClassName="com.mysql.cj.jdbc.Driver"
url="jdbc:mysql://127.0.0.1:3306/java1_bookstore"
username="root"
password="1234"
maxTotal="8"
maxIdle="8"
maxWaitMillis="3000"
```

/>

### 3. 기능구현 - dao (book)

```
public class BookDAO extends DBHelper{

    private static BookDAO instance = new BookDAO();
    public static BookDAO getInstance() {
        return instance;
    }

    private BookDAO() {}

    public BookVO selectBook (String bookName) {
        BookVO vo = null;

        try {
            conn = getConnection();
            psmt = conn.prepareStatement("select * from `book` where `bookName`=?"");
            psmt.setString(1, bookName);
            rs = psmt.executeQuery();

            if(rs.next()) {
                vo = new BookVO();
                vo.setBookId(rs.getInt(1));
                vo.setBookName(rs.getString(2));
                vo.setPubName(rs.getString(3));
                vo.setPrice(rs.getInt(4));
            }

            close();

        }catch (Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public List<BookVO> selectBooks() {

        List<BookVO> books = new ArrayList<>();

        try {
            conn = getConnection();
            stmt = conn.createStatement();
            rs = stmt.executeQuery("select * from `book`");
```

```
            rs = stmt.executeQuery("select * from `book`");

            while(rs.next()) {
                BookVO vo = new BookVO();
                vo.setBookId(rs.getInt(1));
                vo.setBookName(rs.getString(2));
                vo.setPubName(rs.getString(3));
                vo.setPrice(rs.getInt(4));
                books.add(vo);
            }

            close();

        }catch (Exception e) {
            e.printStackTrace();
        }
        return books;
    }

    public void insertBook (BookVO vo) {

        try {
            conn = getConnection();
            psmt = conn.prepareStatement("insert into `book` values (?, ?, ?, ?)");
            psmt.setInt(1, vo.getBookId());
            psmt.setString(2, vo.getBookName());
            psmt.setString(3, vo.getPubName());
            psmt.setInt(4, vo.getPrice());
            psmt.executeUpdate();
            close();
        }catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void modifyBook (BookVO vo) {

        try {
            conn = getConnection();
            psmt = conn.prepareStatement("update `book` set `bookName`=?, `pubName`=?, `price`=? where `bookId`=?");
            psmt.setString(1, vo.getBookName());
            psmt.setString(2, vo.getPubName());
            psmt.setInt(3, vo.getPrice());
            psmt.setInt(4, vo.getBookId());
            psmt.executeUpdate();
            close();
```

```
        }

    }catch (Exception e) {
        e.printStackTrace();
    }
}

public void deleteBook (String BookName) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement("delete from `book` where `bookName`=?"");
        psmt.setString(1, BookName);
        psmt.executeUpdate();
        close();
    }catch (Exception e) {
        e.printStackTrace();
    }
}
```

### 3. 기능구현 - dao (customer)

```
public class CustomerDAO extends DBHelper{

    private static CustomerDAO instance = new CustomerDAO();
    public static CustomerDAO getInstance() {
        return instance;
    }

    private CustomerDAO() {}

    public CustomerVO selectBook (String CusName) {
        CustomerVO vo = null;

        try {
            conn = getConnection();
            psmt = conn.prepareStatement("select * from `customer` where `CusName`=?"");
            psmt.setString(1, CusName);
            rs = psmt.executeQuery();

            if(rs.next()) {
                vo = new CustomerVO();
                vo.setCusId(rs.getInt(1));
                vo.setCusName(rs.getString(2));
                vo.setCusAddr(rs.getString(3));
                vo.setCusHp(rs.getString(4));
            }

            close();

        }catch (Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public List<CustomerVO> selectCustomers() {

        List<CustomerVO> customers = new ArrayList<>();

        try {
            conn = getConnection();
            stmt = conn.createStatement();
            rs = stmt.executeQuery("select * from `customer`");
```

```
            while(rs.next()) {
                CustomerVO vo = new CustomerVO();
                vo.setCusId(rs.getInt(1));
                vo.setCusName(rs.getString(2));
                vo.setCusAddr(rs.getString(3));
                vo.setCusHp(rs.getString(4));
                customers.add(vo);
            }

            close();

        }catch (Exception e) {
            e.printStackTrace();
        }
        return customers;
    }

    public void insertCustomer (CustomerVO vo) {
        try {
            conn = getConnection();
            psmt = conn.prepareStatement("insert into `customer` values (?, ?, ?, ?)");
            psmt.setInt(1, vo.getCusId());
            psmt.setString(2, vo.getCusName());
            psmt.setString(3, vo.getCusAddr());
            psmt.setString(4, vo.getCusHp());
            psmt.executeUpdate();
            close();
        }catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void modifyCustomer (CustomerVO vo) {
        try {
            conn = getConnection();
            psmt = conn.prepareStatement("update `customer` set `CusName`=?, `CusAddr`=?, `CusHp`=? where `CusId`=?"");
            psmt.setString(1, vo.getCusName());
            psmt.setString(2, vo.getCusAddr());
            psmt.setString(3, vo.getCusHp());
            psmt.setInt(4, vo.getCusId());
            psmt.executeUpdate();
            close();
```

```
        }catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void deleteCustomer (String CusName) {
        try {
            conn = getConnection();
            psmt = conn.prepareStatement("delete from `customer` where `CusName`=?"");
            psmt.setString(1, CusName);
            psmt.executeUpdate();
            close();
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



### 3. 기능구현 - controller(book)

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/list.do")
public class ListController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        List<BookVO> books = BookDAO.getInstance().selectBooks();

        req.setAttribute("books", books);

        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/list.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

List(목록)

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/register.do")
public class RegisterController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/register.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String bookId = req.getParameter("bookId");
        String bookName = req.getParameter("bookName");
        String pubName = req.getParameter("pubName");
        String price = req.getParameter("price");

        BookVO vo = new BookVO();
        vo.setBookId(bookId);
        vo.setBookName(bookName);
        vo.setPubName(pubName);
        vo.setPrice(price);

        BookDAO.getInstance().insertBook(vo);

        resp.sendRedirect("/Bookstore2/book/list.do");
    }
}
```

Register(등록)

### 3. 기능구현 - controller(book)

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/modify.do")
public class ModifyController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String bookName = req.getParameter("bookName");
        BookVO vo = BookDAO.getInstance().selectBook(bookName);

        req.setAttribute("vo", vo);

        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/modify.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String bookId = req.getParameter("bookId");
        String bookName = req.getParameter("bookName");
        String pubName = req.getParameter("pubName");
        String price = req.getParameter("price");

        BookVO vo = new BookVO();
        vo.setBookId(bookId);
        vo.setBookName(bookName);
        vo.setPubName(pubName);
        vo.setPrice(price);

        BookDAO.getInstance().modifyBook(vo);

        resp.sendRedirect("/Bookstore2/book/list.do");
    }
}
```

modify(수정)

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/delete.do")
public class DeleteController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String bookName = req.getParameter("bookName");
        BookDAO.getInstance().deleteBook(bookName);

        resp.sendRedirect("/Bookstore2/book/list.do");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

delete(삭제)

### 3. 기능구현 - controller(customer)

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/list.do")
public class ListController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        List<CustomerVO> customers = CustomerDAO.getInstance().selectCustomers();

        req.setAttribute("customers", customers);

        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/list.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

List(목록)

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/register.do")
public class RegisterController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/register.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String cusId = req.getParameter("cusId");
        String cusName = req.getParameter("cusName");
        String cusAddr = req.getParameter("cusAddr");
        String cusHp = req.getParameter("cusHp");

        CustomerVO vo = new CustomerVO();
        vo.setCusId(0);
        vo.setCusName(cusName);
        vo.setCusAddr(cusAddr);
        vo.setCusHp(cusHp);

        CustomerDAO.getInstance().insertCustomer(vo);

        resp.sendRedirect("/Bookstore2/customer/list.do");
    }
}
```

Register(등록)

### 3. 기능구현 - controller(customer)

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/modify.do")
public class ModifyController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String cusName = req.getParameter("cusName");
        CustomerVO vo = CustomerDAO.getInstance().selectBook(cusName);

        req.setAttribute("vo", vo);

        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/modify.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String cusId = req.getParameter("cusId");
        String cusName = req.getParameter("cusName");
        String cusAddr = req.getParameter("cusAddr");
        String cusHp = req.getParameter("cusHp");

        CustomerVO vo = new CustomerVO();
        vo.setCusId(cusId);
        vo.setCusName(cusName);
        vo.setCusAddr(cusAddr);
        vo.setCusHp(cusHp);

        CustomerDAO.getInstance().modifyCustomer(vo);

        resp.sendRedirect("/Bookstore2/customer/list.do");
    }
}
```

modify(수정)

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/delete.do")
public class DeleteController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String cusName = req.getParameter("cusName");
        CustomerDAO.getInstance().deleteCustomer(cusName);

        resp.sendRedirect("/Bookstore2/customer/list.do");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

delete(삭제)

# 3. 기능구현 - 코드(book)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>book::list</title>
  </head>
  <body>
    <h3>도서목록</h3>

    <a href="/Bookstore2/">처음오르</a>
    <a href="/Bookstore2/book/register.do">도서등록</a>

    <table border="1">
      <tr>
        <th>도서번호</th>
        <th>도서명</th>
        <th>출판사</th>
        <th>가격</th>
        <th>관리</th>
      </tr>
      <c:forEach var="book" items="${books}">
        <tr>
          <td>${book.getBookId()}</td>
          <td>${book.getBookName()}</td>
          <td>${book.getPubName()}</td>
          <td>${book.getPrice()}</td>
          <td>
            <a href="/Bookstore2/book/modify.do?bookName=${book.bookName}">수정</a>
            <a href="/Bookstore2/book/delete.do?bookName=${book.bookName}">삭제</a>
          </td>
        </tr>
      </c:forEach>
    </table>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>book::register</title>
  </head>
  <body>
    <h3>도서등록</h3>

    <a href="/Bookstore2/">처음오르</a>
    <a href="/Bookstore2/book/List.do">도서목록</a>

    <form action="/Bookstore2/book/register.do" method="post">
      <table border="1">
        <tr>
          <td>도서번호</td>
          <td><input type="text" name="bookId"/></td>
        </tr>
        <tr>
          <td>도서명</td>
          <td><input type="text" name="bookName"/></td>
        </tr>
        <tr>
          <td>출판사</td>
          <td><input type="text" name="pubName"/></td>
        </tr>
        <tr>
          <td>가격</td>
          <td><input type="number" name="price"/></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="등록하기"/>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>book::modify</title>
  </head>
  <body>
    <h3>도서수정</h3>

    <a href="/Bookstore2/">처음오르</a>
    <a href="/Bookstore2/book/List.do">도서목록</a>
    <form action="/Bookstore2/book/modify.do" method="post">
      <table border="1">
        <tr>
          <td>도서번호</td>
          <td><input type="text" name="bookId" readonly value="${requestScope.vo.bookId}"/></td>
        </tr>
        <tr>
          <td>도서명</td>
          <td><input type="text" name="bookName" value="${vo.bookName}"/></td>
        </tr>
        <tr>
          <td>출판사</td>
          <td><input type="text" name="pubName" value="${vo.pubName}"/></td>
        </tr>
        <tr>
          <td>가격</td>
          <td><input type="number" name="price" value="${vo.price}"/></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="수정"/>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

### 3. 기능구현 - 코드(customer)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>customer::list</title>
  </head>
  <body>
    <h3>고객목록</h3>

    <a href="/Bookstore2/">처음오르</a>
    <a href="/Bookstore2/customer/register.do">고객등록</a>

    <table border="1">
      <tr>
        <th>고객번호</th>
        <th>고객명</th>
        <th>주소</th>
        <th>휴대폰</th>
        <th>관리</th>
      </tr>
      <c:forEach var="customer" items="{customers}">
        <tr>
          <td>${customer.getCusId()}</td>
          <td>${customer.getCusName()}</td>
          <td>${customer.getCusAddr()}</td>
          <td>${customer.getCusHp()}</td>
          <td>
            <a href="/Bookstore2/customer/modify.do?cusName=${customer.cusName}">수정</a>
            <a href="/Bookstore2/customer/delete.do?cusName=${customer.cusName}">삭제</a>
          </td>
        </tr>
      </c:forEach>
    </table>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>customer::modify</title>
  </head>
  <body>
    <h3>고객수정</h3>

    <a href="/Bookstore2/">처음오르</a>
    <a href="/Bookstore2/customer/List.do">고객목록</a>
    <form action="/Bookstore2/customer/modify.do" method="post">
      <table border="1">
        <tr>
          <td>고객번호</td>
          <td><input type="text" name="cusId" readonly value="${requestScope.vo.cusId}"/></td>
        </tr>
        <tr>
          <td>고객명</td>
          <td><input type="text" name="cusName" value="${vo.cusName}"/></td>
        </tr>
        <tr>
          <td>주소</td>
          <td><input type="text" name="cusAddr" value="${vo.cusAddr}"/></td>
        </tr>
        <tr>
          <td>휴대폰</td>
          <td><input type="text" name="cusHp" value="${vo.cusHp}"/></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="수정"/>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>customer::register</title>
  </head>
  <body>
    <h3>고객등록</h3>

    <a href="/Bookstore2/">처음오르</a>
    <a href="/Bookstore2/customer/List.do">고객목록</a>

    <form action="/Bookstore2/customer/register.do" method="post">
      <table border="1">
        <tr>
          <td>고객명</td>
          <td><input type="text" name="cusName"/></td>
        </tr>
        <tr>
          <td>주소</td>
          <td><input type="text" name="cusAddr"/></td>
        </tr>
        <tr>
          <td>휴대폰</td>
          <td><input type="text" name="cusHp"/></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="등록하기"/>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

# 4. 실행 (도서)

## 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
2	축구이해함	미디어대한	50020	<a href="#">수정</a> <a href="#">삭제</a>

## 도서등록

[처음으로 도서목록](#)

도서번호	3
도서명	민혁이의 일대기
출판사	내머리
가격	9999
<input type="button" value="등록하기"/>	

## 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
2	축구이해함	미디어대한	50020	<a href="#">수정</a> <a href="#">삭제</a>
3	민혁이의 일대기	내머리	9999	<a href="#">수정</a> <a href="#">삭제</a>

등록 후 삭제하는 화면

## 도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
3	민혁이의 일대기	내머리	9999	<a href="#">수정</a> <a href="#">삭제</a>

## 4. 실행 (고객)

### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
9	공고공	부산시 서울구	010-1234-2231	<a href="#">수정</a> <a href="#">삭제</a>

### 고객등록

[처음으로 고객목록](#)

고객명	유신킴
주소	부산시 마포구
휴대폰	010-9999-9911
<input type="button" value="등록하기"/>	

### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
9	공고공	부산시 서울구	010-1234-2231	<a href="#">수정</a> <a href="#">삭제</a>
10	유신킴	부산시 마포구	010-9999-9911	<a href="#">수정</a> <a href="#">삭제</a>

등록 후 삭제하는 화면

### 고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
10	유신킴	부산시 마포구	010-9999-9911	<a href="#">수정</a> <a href="#">삭제</a>



끝