#2022.11.18 최아영

수행평가

통합 구현

목차

개발환경,공통 모듈,
서버,배치 구현하기

#1, 프로젝트 생성 및 구성

#2, 화면 구현

#3, 기능 구현

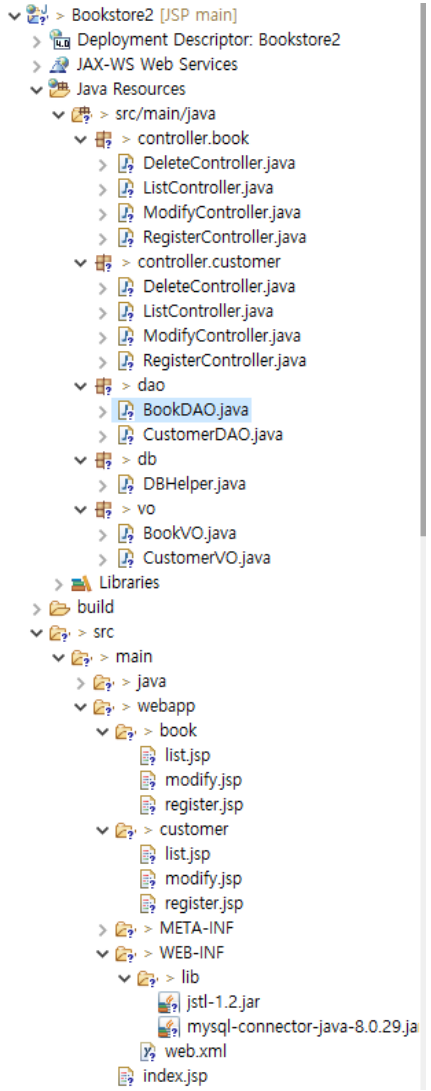
#4, 실행

Part 1,

프로젝트 생성 및 구성

Part 1

프로젝트 생성 및 구성



프로젝트 구조

프로젝트명 : BookStore2

WAS : Tomcat 9

DB : java1_bookstore

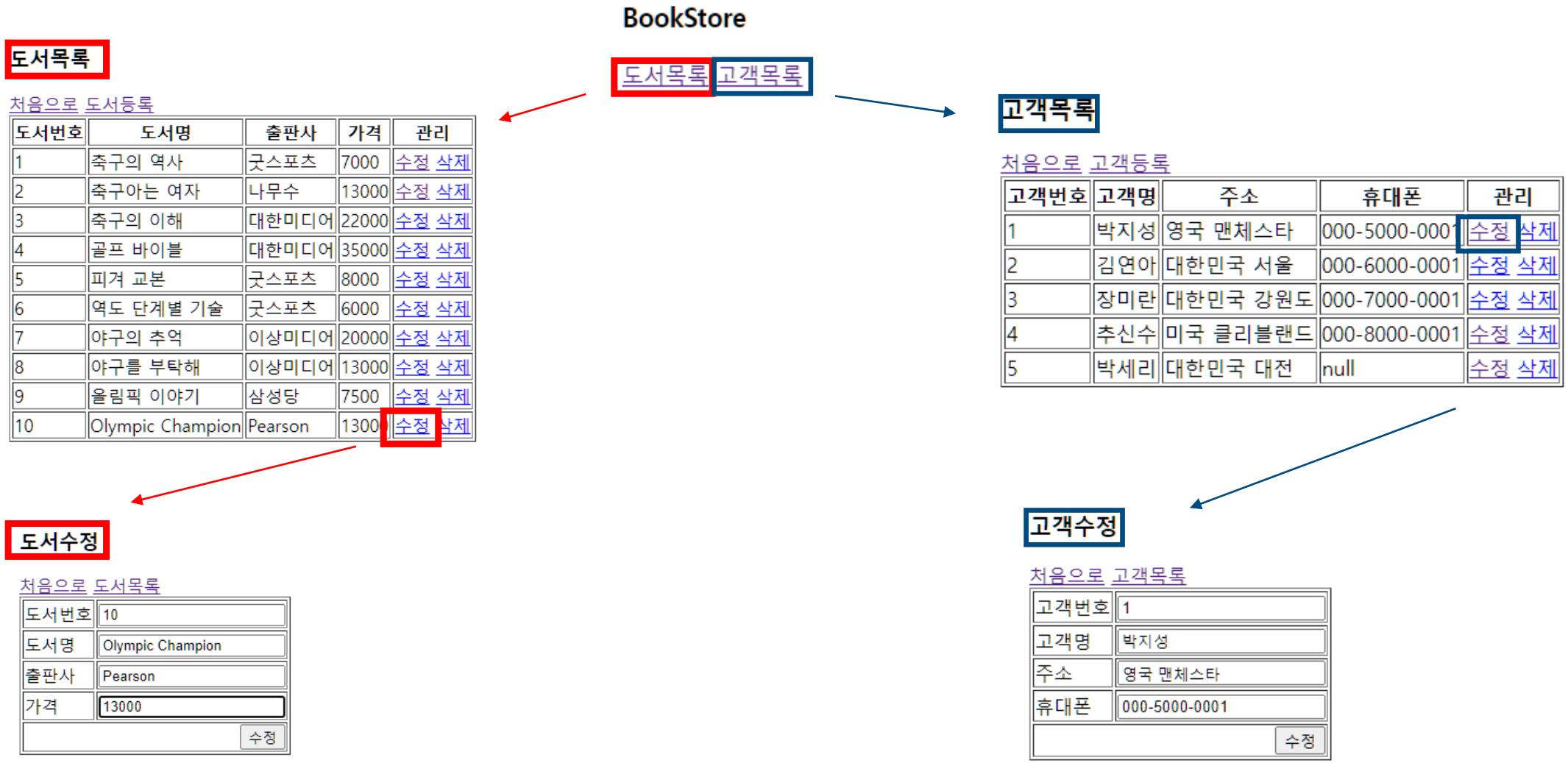
개발도구 : Eclipse, Workbench

Part 2,

화면 구현

Part 2

화면 구현


BookStore
도서목록 고객목록
도서목록
처음으로 도서등록
도서번호 도서명 출판사 가격 관리
1 축구의 역사 굿스포츠 7000 수정 삭제
2 축구아는 여자 나무수 13000 수정 삭제
3 축구의 이해 대한미디어 22000 수정 삭제
4 골프 바이블 대한미디어 35000 수정 삭제
5 피겨 교본 굿스포츠 8000 수정 삭제
6 역도 단계별 기술 굿스포츠 6000 수정 삭제
7 야구의 추억 이상미디어 20000 수정 삭제
8 야구를 부탁해 이상미디어 13000 수정 삭제
9 올림픽 이야기 삼성당 7500 수정 삭제
10 Olympic Champion Pearson 1300 수정 삭제
고객목록
처음으로 고객등록
고객번호 고객명 주소 휴대폰 관리
1 박지성 영국 맨체스타 000-5000-0001 수정 삭제
2 김연아 대한민국 서울 000-6000-0001 수정 삭제
3 장미란 대한민국 강원도 000-7000-0001 수정 삭제
4 추신수 미국 클리블랜드 000-8000-0001 수정 삭제
5 박세리 대한민국 대전 null 수정 삭제
도서수정
처음으로 도서목록
도서번호 10
도서명 Olympic Champion
출판사 Pearson
가격 13000
수정
고객수정
처음으로 고객목록
고객번호 1
고객명 박지성
주소 영국 맨체스타
휴대폰 000-5000-0001
수정

Part 3,

기능 구현

Part 3

기능 구현

[index]

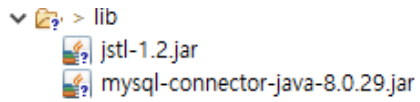
```
<body>
    <h3>BookStore</h3>
    <a href="./book/list.do">도서목록</a>
    <a href="./customer/list.do">고객목록</a>
</body>
```

<코드 GitHub참조>

click

[SQL연결]

```
public class DBHelper {

    protected Connection conn = null;
    protected PreparedStatement psmt = 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(psmt != null) {
                psmt.close();
            }

            if(conn != null) {
                conn.close();
            }

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




```
<!-- 커넥션 풀 설정 -->
<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"
/>
```

Part 3

기능 구현

```jsp
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>book::list</title>
    </head>
    <body>
        <h3>도서목록</h3>
        <a href="/Bookstore2/">처음으로</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.getPublisher()}</td>
                    <td>${book.getPrice()}</td>
                    <td>
                        <a href="/Bookstore2/book/modify.do?bookId=${book.bookId}">수정</a>
                        <a href="/Bookstore2/book/delete.do?bookId=${book.bookId}">삭제</a>
                    </td>
                </tr>
            </c:forEach>
        </table>
    </body>
</html>
```

[book-list]

```jsp
<body>
    <h3>도서수정</h3>
    <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="${requestScope.bv.bookId}"/></td>
            </tr>
            <tr>
                <td>도서명</td>
                <td><input type="text" name="bookName" value="${bv.bookName}"/></td>
            </tr>
            <tr>
                <td>출판사</td>
                <td><input type="text" name="publisher" value="${bv.publisher}"/></td>
            </tr>
            <tr>
                <td>가격</td>
                <td><input type="number" name="price" value="${bv.price}"/></td>
            </tr>
            <tr>
                <td colspan="2" align="right">
                    <input type="submit" value="수정"/>
                </td>
            </tr>
        </table>
    </form>
</body>
```

[book-modify]

Part 3

기능 구현

```
<body>
    <h3>도서등록</h3>
    <a href="/Bookstore2/">처음으로</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" placeholder="도서번호 입력"/></td>
            </tr>
            <tr>
                <td>도서명</td>
                <td><input type="text" name="bookName" placeholder="도서명 입력"/></td>
            </tr>
            <tr>
                <td>출판사</td>
                <td><input type="text" name="publisher" placeholder="출판사 입력"/></td>
            </tr>
            <tr>
                <td>가격</td>
                <td><input type="number" name="price" placeholder="가격 입력"/></td>
            </tr>
            <tr>
                <td colspan="2" align="right">
                    <input type="submit" value="등록"/>
                </td>
            </tr>
        </table>
    </form>
</body>
```

[book-register]

```
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>customer::list</title>
    </head>
    <body>
        <h3>고객목록</h3>
        <a href="/Bookstore2/index.jsp">처음으로</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="${custs}">
                <tr>
                    <td>${customer.getCustId()}</td>
                    <td>${customer.getName()}</td>
                    <td>${customer.getAddress()}</td>
                    <td>${customer.getPhone()}</td>
                    <td>
                        <a href="/Bookstore2/customer/modify.do?custId=${customer.custId}">수정</a>
                        <a href="/Bookstore2/customer/delete.do?custId=${customer.custId}">삭제</a>
                    </td>
                </tr>
            </c:forEach>
        </table>
    </body>
</html>
```

[customer-list]

Part 3

기능 구현

```
<body>
    <h3>고객수정</h3>
    <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="custId" readonly value="${requestScope.cv.custId}"/></td>
            </tr>
            <tr>
                <td>고객명</td>
                <td><input type="text" name="name" value="${cv.name}"/></td>
            </tr>
            <tr>
                <td>주소</td>
                <td><input type="text" name="address" value="${cv.address}"/></td>
            </tr>
            <tr>
                <td>휴대폰</td>
                <td><input type="text" name="phone" value="${cv.phone}"/></td>
            </tr>
            <tr>
                <td colspan="2" align="right">
                    <input type="submit" value="수정"/>
                </td>
            </tr>
        </table>
    </form>
</body>
```

[customer-modify]

```
<body>
    <h3>고객등록</h3>
    <a href="/Bookstore2/">처음으로</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="name" placeholder="고객명 입력"/></td>
            </tr>
            <tr>
                <td>주소</td>
                <td><input type="text" name="address" placeholder="주소 입력"/></td>
            </tr>
            <tr>
                <td>휴대폰</td>
                <td><input type="text" name="phone" placeholder="휴대폰 입력"/></td>
            </tr>
            <tr>
                <td colspan="2" align="right">
                    <input type="submit" value="등록"/>
                </td>
            </tr>
        </table>
    </form>
</body>
```

[customer-register]

Part 3

기능 구현 [VO]

```java
package vo;

public class BookVO {
    private int bookId;
    private String bookName;
    private String publisher;
    private String price;


    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public String getBookName() {
        return bookName;
    }
    public void setBookName(String bookName) {
        this.bookName = bookName;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }


}
```

```java
package vo;

public class CustomerVO {
    private int custId;
    private String name;
    private String address;
    private String phone;

    public int getCustId() {
        return custId;
    }
    public void setCustId(int custId) {
        this.custId = custId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }


}
```

Part 3

기능 구현 [BookDAO]

```
public class BookDAO extends DBHelper {

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

    private BookDAO() {}

    public void insertUser(BookVO bv) {

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

    public BookVO selectUser(String bookId) {

        BookVO bv = null;

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

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

            close();

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

        return bv;
    }
```

```
    public List<BookVO> selectUsers() {

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

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

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

            close();

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

        return books;
    }

    public void updateUser(BookVO bv) {
        try {
            conn = getConnection();
            psmt = conn.prepareStatement("update `book` set `bookName`=?, `publisher`=?, `price`=? where `bookId`=?" );
            psmt.setString(1, bv.getBookName());
            psmt.setString(2, bv.getPublisher());
            psmt.setString(3, bv.getPrice());
            psmt.setInt(4, bv.getBookId());
            psmt.executeUpdate();
            close();
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
```

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

Part 3

기능 구현 [CustomerDAO]

```
private CustomerDAO() {}

public void insertUser(CustomerVO bv) {

    try {
        conn = getConnection();
        psmt = conn.prepareStatement("insert into `customer` values (?,?,?,?)");
        psmt.setInt(1, bv.getCustId());
        psmt.setString(2, bv.getName());
        psmt.setString(3, bv.getAddress());
        psmt.setString(4, bv.getPhone());
        psmt.executeUpdate();
        close();
    }catch (Exception e) {
        e.printStackTrace();
    }
}

public CustomerVO selectUser(String custId) {

    CustomerVO cv = null;

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

        if(rs.next()) {
            cv = new CustomerVO();
            cv.setCustId(rs.getInt(1));
            cv.setName(rs.getString(2));
            cv.setAddress(rs.getString(3));
            cv.setPhone(rs.getString(4));
        }

        close();

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

    return cv;
}
```

```
public List<CustomerVO> selectUsers() {

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

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

        while(rs.next()) {
            CustomerVO cv = new CustomerVO();
            cv.setCustId(rs.getInt(1));
            cv.setName(rs.getString(2));
            cv.setAddress(rs.getString(3));
            cv.setPhone(rs.getString(4));
            custs.add(cv);
        }

        close();

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

    return custs;
}

public void updateUser(CustomerVO cv) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement("update `customer` set `name`=?, `address`=?, `phone`=? where `custId`=?");
        psmt.setString(1, cv.getName());
        psmt.setString(2, cv.getAddress());
        psmt.setString(3, cv.getPhone());
        psmt.setInt(4, cv.getCustId());
        psmt.executeUpdate();
        close();
    }catch (Exception e) {
        e.printStackTrace();
    }
}

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

Part 3

기능 구현 [controller.book]

```java
@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().selectUsers();

        // View에서 데이터 출력을 위한 request Scope 데이터 설정
        req.setAttribute("books", books);

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

    }
}
```

[ListController]

```java
@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 {
        // View forward
        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/register.jsp");
        dispatcher.forward(req, resp);
    }

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

        BookVO bv = new BookVO();
        bv.setBookId(Integer.parseInt(bookId));
        bv.setBookName(bookName);
        bv.setPublisher(publisher);
        bv.setPrice(price);

        BookDAO.getInstance().insertUser(bv);

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

[RegisterController]

Part 3

기능 구현 [controller.book]

```java
@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 bookId = req.getParameter("bookId");
        BookVO bv = BookDAO.getInstance().selectUser(bookId);

        req.setAttribute("bv", bv);

        // 포워드
        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 publisher = req.getParameter("publisher");
        String price     = req.getParameter("price");

        BookVO bv = new BookVO();
        bv.setBookId(Integer.parseInt(bookId));
        bv.setBookName(bookName);
        bv.setPublisher(publisher);
        bv.setPrice(price);

        BookDAO.getInstance().updateUser(bv);

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

[ModifyController]

```java
@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 bookId = req.getParameter("bookId");
        BookDAO.getInstance().deleteUser(bookId);

        resp.sendRedirect("/book/list.do");
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
```

[DeleteController]

Part 3

기능 구현 [controller.customer]

```java
@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> custs = CustomerDAO.getInstance().selectUsers();

        // View에서 데이터 출력을 위한 request Scope 데이터 설정
        req.setAttribute("custs", custs);

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

    }
```

[ListController]

```java
@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 {
        // View forward
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/register.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        // POST 요청
        String name = req.getParameter("name");
        String address   = req.getParameter("address");
        String phone  = req.getParameter("phone");

        CustomerVO cv = new CustomerVO();
        cv.setName(name);
        cv.setAddress(address);
        cv.setPhone(phone);

        CustomerDAO.getInstance().insertUser(cv);

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

[RegisterController]

Part 3

기능 구현 [controller.customer]

```java
@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 custId = req.getParameter("custId");
        CustomerVO cv = CustomerDAO.getInstance().selectUser(custId);

        req.setAttribute("cv", cv);

        // 포워드
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/modify.jsp");
        dispatcher.forward(req, resp);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String custId    = req.getParameter("custId");
        String name  = req.getParameter("name");
        String address = req.getParameter("address");
        String phone     = req.getParameter("phone");

        CustomerVO cv = new CustomerVO();
        cv.setCustId(Integer.parseInt(custId));
        cv.setName(name);
        cv.setAddress(address);
        cv.setPhone(phone);

        CustomerDAO.getInstance().updateUser(cv);

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

[ModifyController]

```java
@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 custId = req.getParameter("custId");
        CustomerDAO.getInstance().deleteUser(custId);

        resp.sendRedirect("/customer/list.do");
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

[DeleteController]

Part 4,

실행

Part 4

실행 [book]

BookStore

도서목록 고객목록

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별 기술	굿스포츠	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champion	Pearson	13000	수정 삭제

도서수정

처음으로 도서목록

도서번호	1
도서명	축구의 역사
출판사	굿스포츠
가격	7000
	수정

도서수정

처음으로 도서목록

도서번호	1
도서명	축구의 역사
출판사	굿스포츠1
가격	7100
	수정

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠1	7100	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별 기술	굿스포츠	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champion	Pearson	13000	수정 삭제

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
2	축구아는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별 기술	굿스포츠	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champion	Pearson	13000	수정 삭제

Part 4

실행 [customer]

BookStore

도서목록 고객목록

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	000-5000-0001	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	박세리	대한민국 대전	null	수정 삭제

고객수정

처음으로 고객목록

고객번호	1
고객명	박지성
주소	영국 맨체스타
휴대폰	000-5000-0001
	수정

고객수정

처음으로 고객목록

고객번호	1
고객명	박지성
주소	영국 런던
휴대폰	000-5000-0002
	수정

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 런던	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	박세리	대한민국 대전	null	수정 삭제

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	박세리	대한민국 대전	null	수정 삭제

감사합니다.