

통합 구현

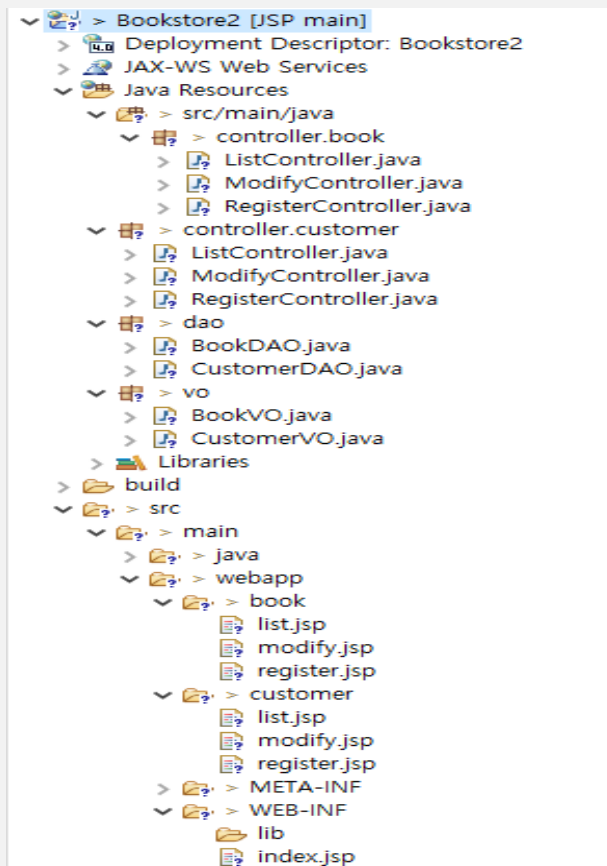
2022/11/18

임민지

목 차

- 1. 프로젝트 생성 및 구성
- 2. 화면 구현
- 3. 기능 구현
- 4. 실행

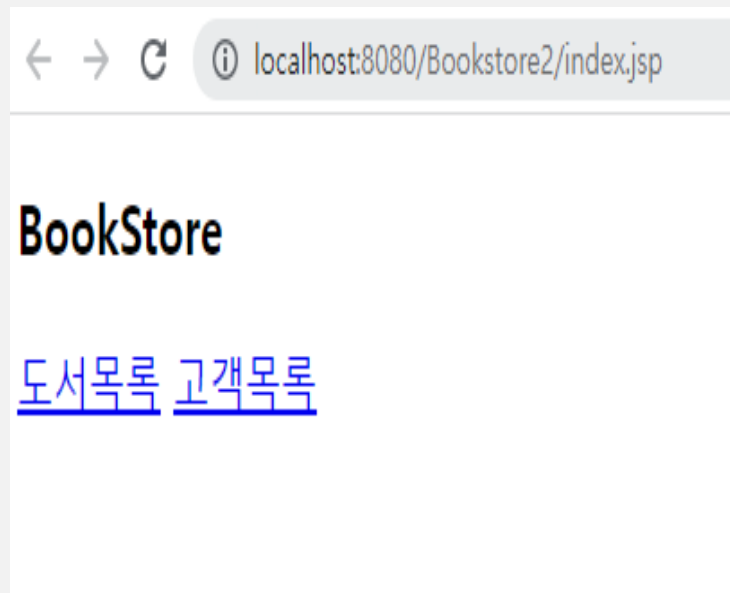
문제 1. 프로젝트 생성 및 구성



문제 2. 화면 구현

- 시작 페이지

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
  </head>
  <body>
    <h3>BookStore</h3>
    <a href="#">도서목록</a>
    <a href="#">고객목록</a>
  </body>
</html>
```



문제 2-1. 화면 구현

- 도서 목록

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h3>도서목록</h3>
<a href="#"> 처음으로</a>
<a href="#"> 도서등록</a>

<table border="1">
<tr>
<th>도서번호</th>
<th>도서명</th>
<th>출판사</th>
<th>가격</th>
<th>관리</th>
</tr>
<tr>
<td>1</td>
<td>축구의 역사</td>
<td>굿스포츠</td>
<td>7,000</td>
<td>
<a href="#">수정</a>
<a href="#">삭제</a>
</td>
</tr>
<tr>
<td>2</td>
<td>축구의 이해</td>
<td>대한미디어</td>
<td>22,000</td>
<td>
<a href="#">수정</a>
<a href="#">삭제</a>
</td>
</tr>
<tr>
<td>3</td>
<td>골프바이블</td>
<td>대한미디어</td>
<td>35,000</td>
<td>
<a href="#">수정</a>
<a href="#">삭제</a>
</td>
</tr>
</table>
</body>
</html>
```

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7,000	수정 삭제
2	축구의 이해	대한미디어	22,000	수정 삭제
3	골프바이블	대한미디어	35,000	수정 삭제

도서 목록

ListController

```
package controller.book;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

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

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

← → ↻ ⓘ localhost:8080/Bookstore2/book/list.do

도서목록

[처음으로](#) [도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7,000	수정 삭제
2	축구의 이해	대한미디어	22,000	수정 삭제
3	골프바이블	대한미디어	35,000	수정 삭제

문제 2-2. 화면 구현

- 도서 등록

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Insert title here</title>
  </head>
  <body>
    <h3>도서등록</h3>
    <a href="#">처음으로</a>
    <a href="#">도서목록</a>

    <form action="./registerProc.jsp">
      <table border="1">
        <tr>
          <th>도서번호</th>
          <td><input type="text" name="bookId" /></td>
        </tr>
        <tr>
          <th>도서명</th>
          <td><input type="text" name="bookName" /></td>
        </tr>
        <tr>
          <th>출판사</th>
          <td><input type="text" name="publisher" /></td>
        </tr>
        <tr>
          <th>가격</th>
          <td><input type="text" name="price" /></td>
        </tr>
        <tr>
          <td colspan="2" align="right"><input type="submit" value="등록"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

도서등록

처음으로 도서목록

도서번호	<input type="text"/>
도서명	<input type="text"/>
출판사	<input type="text"/>
가격	<input type="text"/>
<input type="submit" value="등록"/>	

도서 등록

RegisterController

```
package controller.book;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@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 {
    }
}
```

← → ↻ ⓘ localhost:8080/Bookstore2/book/register.do

도서등록

[처음으로 도서목록](#)

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

```
// 데이터 수신
String bookId = req.getParameter("bookId");
String bookName = req.getParameter("bookName");
String publisher = req.getParameter("publisher");
String price = req.getParameter("price");

BookVO vo = new BookVO();
vo.setBookId(bookId);
vo.setBookName(bookName);
vo.setPublisher(publisher);
vo.setPrice(price);

// DAO - insertBook() 호출
BookDAO.getInstance().insertBook(vo);

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


도서 등록

도서등록

처음으로

도서번호	123
도서명	sdf
출판사	sgetryhty
가격	777777
<input type="button" value="등록"/>	

bookId	bookName	publisher	price
1	축구의 역사	굿스포츠	7,000
2	축구하는 여자	나무수	13,000
3	축구의 이해	대한미디어	22,000
4	골프 바이블	대한미디어	35,000
5	피겨교본	굿스포츠	8,000
6	역도 단계별기술	굿스포츠	6,000
7	야구의 추억	이상미디어	20,000
8	야구를 부탁해	이상미디어	13,000
9	올림픽 이야기	삼성당	7,500
10	Olympic Champions	Pearson	13,000
11	jjj	jjjjj	1,200
23	i-ê³¼e°ëë	i-ê³¼	8,000
34	sdfgdfsg	dsfgs	3,333
123	sdf	sgetryhty	777,777
555	i-ê³¼	i-ê³¼i§	5,000
777	sdfs	sdf	1,000
999	etrtwr	wertrt	888,999

문제 2-3. 화면 구현

• 도서 수정

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h3>도서수정</h3>
<a href="#">처음으로</a>
<a href="#">도서등록</a>

<table border="1">
<tr>
<th>도서번호</th>
<td><input type="text" name="bookId" /></td>
</tr>
<tr>
<th>도서명</th>
<td><input type="text" name="bookName" /></td>
</tr>
<tr>
<th>출판사</th>
<td><input type="text" name="publisher" /></td>
</tr>
<tr>
<th>가격</th>
<td><input type="text" name="price" /></td>
</tr>
<tr>
<td colspan="2" align="right"><input type="submit" value="수정"/></td>
</tr>
</table>
</form>
</body>
</html>
```

도서수정

[처음으로 도서등록](#)

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

ModifyController

```
package controller.book;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

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

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

    }

}
```

localhost:8080/Bookstore2/book/modify.do

도서수정

[처음으로 도서등록](#)

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

BOOKDAO & BOOKVO

```
package dao;
import java.util.ArrayList;
import java.util.List;

import db.DBHelper;
import vo.BookVO;

public class BookDAO extends DBHelper {

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

    // DB CRUD
    public void insertBook(BookVO vo) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("insert into 'book' values (?, ?, ?, ?)");
            pstmt.setInt(1, vo.getBookId());
            pstmt.setString(2, vo.getBookName());
            pstmt.setString(3, vo.getPublisher());
            pstmt.setInt(4, vo.getPrice());
            pstmt.executeUpdate();
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public BookVO selectBook(String bookId) {
        BookVO vo = null;
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("select * from 'book' where 'bookId'=?");
            pstmt.setString(1, bookId);
            rs = pstmt.executeQuery();

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

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

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

```
        public List<BookVO> selectBooks() {
            List<BookVO> books = new ArrayList<>();
            try {
                conn = getConnection();
                pstmt = conn.prepareStatement("select * from 'book'");
                rs = pstmt.executeQuery();

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

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

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

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

```
package vo;

public class BookVO {

    private int bookId;
    private String bookName;
    private String publisher;
    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 getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
    public void setprice(String price) {
        this.price = Integer.parseInt(price);
    }
}
```

문제 2-4. 화면 구현

고객 목록

```
<%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Insert title here</title>
</head>
<body>
<h3>고객목록</h3>
<a href="#">처음인원</a>
<a href="#">고객목록</a>
<table border="1">
<tr>
<th>고객번호</th>
<th>고객명</th>
<th>주소</th>
<th>휴대폰</th>
<th>관리</th>
</tr>
<tr>
<td>1</td>
<td>박지성</td>
<td>영국 맨체스터</td>
<td>010-1234-1001</td>
<td>
<a href="#">수정</a>
<a href="#">삭제</a>
</td>
</tr>
<tr>
<td>2</td>
<td>김연아</td>
<td>대한민국 서울</td>
<td>010-1234-1002</td>
<td>
<a href="#">수정</a>
<a href="#">삭제</a>
</td>
</tr>
<tr>
<td>3</td>
<td>장미란</td>
<td>대한민국 강원도</td>
<td>010-1234-1003</td>
<td>
<a href="#">수정</a>
<a href="#">삭제</a>
</td>
</tr>
</table>
</body>
</html>
```

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	010-1234-1001	수정 삭제
2	김연아	대한민국 서울	010-1234-1002	수정 삭제
3	장미란	대한민국 강원도	010-1234-1003	수정 삭제

ListController

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

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

문제 2-5. 화면 구현

• 고객 등록

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>고객등록</title>
  </head>
  <body>
    <h3>고객등록</h3>

    <a href="#">처음으로</a>
    <a href="#">고객목록</a>

    <table border="1">
      <tr>
        <td>고객명</td>
        <td><input type="text" name="custId" /></td>
      </tr>
      <tr>
        <td>주소</td>
        <td><input type="text" name="name" /></td>
      </tr>
      <tr>
        <td>휴대폰</td>
        <td><input type="text" name="address" /></td>
      </tr>
      <tr>
        <td colspan="2" align="right"><input type="submit" value="등록"/></td>
      </tr>
    </table>
  </form>
</body>
</html>
</html>
```

RegisterController

```
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 doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/register.jsp");
        dispatcher.forward(req, resp);
    }

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

```
// 데이터 수신
String custId = req.getParameter("custId");
String name = req.getParameter("name");
String address = req.getParameter("address");
String phone = req.getParameter("phone");

CustomerVO vo = new CustomerVO();
vo.setCustId(custId);
vo.setName(name);
vo.setAddress(address);
vo.setPhone(phone);

// DAO - insertBook() 호출
CustomerDAO.getInstance().insertCustomer(vo);

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

문제 2-6. 화면 구현

- 고객 수정

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h3>고객수정</h3>
<a href="">처음으로</a>
<a href="">고객등록</a>

<table border="1">
<tr>
<th>고객번호</th>
<td><input type="text" name="custId" /></td>
</tr>
<tr>
<th>고객명</th>
<td><input type="text" name="name" /></td>
</tr>
<tr>
<th>주소</th>
<td><input type="text" name="address" /></td>
</tr>
<tr>
<th>휴대폰</th>
<td><input type="text" name="phone" /></td>
</tr>
<tr>
<td colspan="2" align="right"><input type="submit" value="수정"/></td>
</tr>
</table>
</body>
</html>
```

고객수정

처음으로 고객등록

고객번호	<input type="text"/>
고객명	<input type="text"/>
주소	<input type="text"/>
휴대폰	<input type="text"/>
<input type="submit" value="수정"/>	

ModifyController

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

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

    }

}
```

CUSTOMERDAO&CUSTOMERVO

```
package vo;
import java.util.ArrayList;

public class CustomerDAO extends DBHelper {

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

    // @@@ CRM
    public void insertCustomer(CustomerVO vo) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("insert into 'customer' values (?, ?, ?, ?)");
            pstmt.setInt(1, vo.getCustId());
            pstmt.setString(2, vo.getName());
            pstmt.setString(3, vo.getAddress());
            pstmt.setString(4, vo.getPhone());
            pstmt.executeUpdate();
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public CustomerVO selectCustomer(String custId) {
        CustomerVO vo = null;
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("select * from 'customer' where 'custId'=?");
            pstmt.setString(1, custId);
            rs = pstmt.executeQuery();

            if(rs.next()) {
                vo = new CustomerVO();
                vo.setCustId(rs.getInt(1));
                vo.setName(rs.getString(2));
                vo.setAddress(rs.getString(3));
                vo.setPhone(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.setName(rs.getString(2));
vo.setAddress(rs.getString(3));
vo.setPhone(rs.getString(4));
customers.add(vo);
}
close();
} catch (Exception e) {
    e.printStackTrace();
}
return customers;
}

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

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

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

```
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 void setCustId(String custId) {
        this.custId = Integer.parseInt(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;
    }
}
```