

서버 프로그램 구현 수행평가 실습보고서

2023/01/10 박가영

목차

table of contents

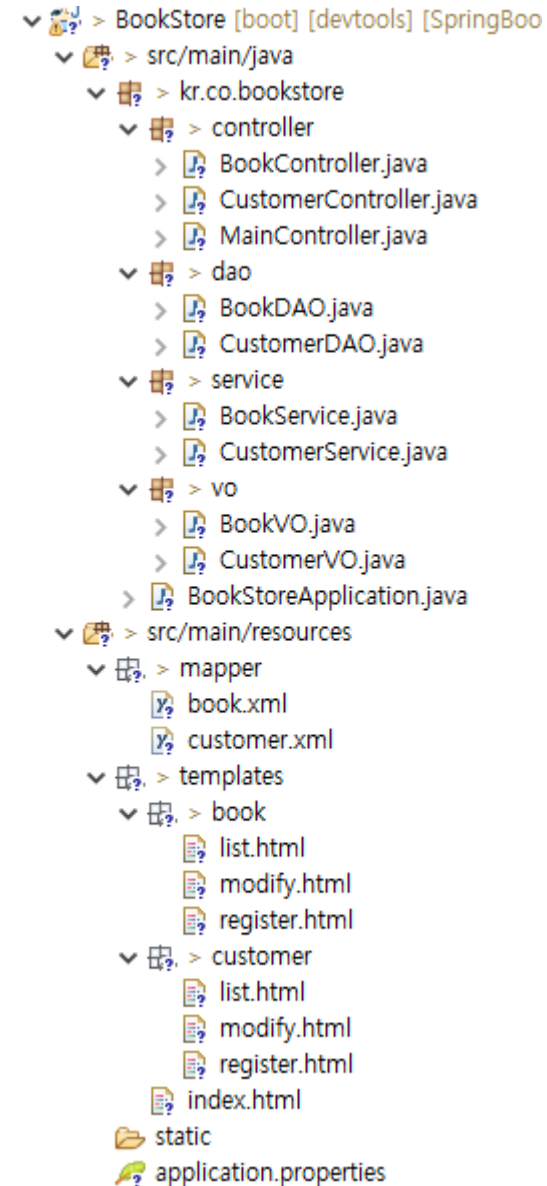
- 1 문제1. 프로젝트 생성
및 구성
- 2 문제2. 화면 구현
- 3 문제3. 기능 구현
- 4 문제4. 실행



문제1-프로젝트 생성 및 구성

문제1

- ProjectName : BookStore
- WAS : Tomcat9
- DB : java2_bookstore
- Framework : SpringBoot, Mybatis
- Tools : Eclipse, Workbench



도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	그냥사는남자	매미스포츠	13000	수정 삭제
2	축구아는 남자	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 단계별기술	굿스포츠	6000	수정 삭제
7	축구알아?	레버너	8000	수정 삭제
9	올림픽 이야기	그린그림	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
20	축구사랑	나는축구러버	15000	수정 삭제
22	테니스왕자	테니스공주	6000	수정 삭제

도서등록

처음으로 도서목록

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

도서수정

처음으로 도서목록

도서번호	<input type="text" value="1"/>
도서명	<input type="text" value="그냥사는남자"/>
출판사	<input type="text" value="매미스포츠"/>
가격	<input type="text" value="13000"/>
<input type="button" value="수정"/>	

고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박민성	영국 맨체스타	100-1000-1111	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
6	홍길동	부산시 동래구	010-1234-1234	수정 삭제
10	이미자	경기도 성남시	010-0000-8888	수정 삭제

고객등록

[처음으로 고객목록](#)

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

고객수정

[처음으로 고객목록](#)

고객번호	<input type="text" value="1"/>
고객명	<input type="text" value="박민성"/>
주소	<input type="text" value="영국 맨체스타"/>
휴대폰	<input type="text" value="100-1000-1111"/>
<input type="button" value="수정"/>	

```
@Controller
public class MainController

    @GetMapping("/index")
    public String index() {
        return "/index";
    }
}
```

```
@Controller
public class BookController {

    @Autowired
    private BookService service;

    @GetMapping("/book/list")
    public String list(Model model) {
        List<BookVO> books = service.selectBooks();
        model.addAttribute("books", books);
        return "/book/list";
    }

    @GetMapping("/book/register")
    public String register() {
        return "/book/register";
    }

    @PostMapping("/book/register")
    public String register(BookVO vo) {
        service.insertBook(vo);
        return "redirect:/book/list";
    }

    @GetMapping("/book/modify")
    public String modify(int bookId, Model model) {
        BookVO book = service.selectBook(bookId);
        model.addAttribute("book", book);
        return "/book/modify";
    }

    @PostMapping("/book/modify")
    public String modify(BookVO vo) {
        service.updateBook(vo);
        return "redirect:/book/list";
    }

    @GetMapping("/book/delete")
    public String delete(int bookId) {
        service.deleteBook(bookId);
        return "redirect:/book/list";
    }
}
```

```
@Controller
public class CustomerController {

    @Autowired
    private CustomerService service;

    @GetMapping("/customer/list")
    public String list(Model model) {
        List<CustomerVO> customers = service.selectCustomers();
        model.addAttribute("customers", customers);
        return "/customer/list";
    }

    @GetMapping("/customer/register")
    public String register() {
        return "/customer/register";
    }

    @PostMapping("/customer/register")
    public String register(CustomerVO vo) {
        service.insertCustomer(vo);
        return "redirect:/customer/list";
    }

    @GetMapping("/customer/modify")
    public String modify(int custId, Model model) {
        CustomerVO customer = service.selectCustomer(custId);
        model.addAttribute("customer", customer);
        return "/customer/modify";
    }

    @PostMapping("/customer/modify")
    public String modify(CustomerVO vo) {
        service.updateCustomer(vo);
        return "redirect:/customer/list";
    }

    @GetMapping("/customer/delete")
    public String delete(int custId) {
        service.deleteCustomer(custId);
        return "redirect:/customer/list";
    }
}
```

```
@Service
public class BookService {

    @Autowired
    private BookDAO dao;

    public void insertBook(BookVO vo) {
        dao.insertBook(vo);
    }
    public BookVO selectBook(int bookId) {
        return dao.selectBook(bookId);
    }
    public List<BookVO> selectBooks() {
        return dao.selectBooks();
    }
    public void updateBook(BookVO vo) {
        dao.updateBook(vo);
    }
    public void deleteBook(int bookId) {
        dao.deleteBook(bookId);
    }
}
```

```
@Service
public class CustomerService {

    @Autowired
    private CustomerDAO dao;

    public void insertCustomer(CustomerVO vo) {
        dao.insertCustomer(vo);
    }
    public CustomerVO selectCustomer(int custId) {
        return dao.selectCustomer(custId);
    }
    public List<CustomerVO> selectCustomers() {
        return dao.selectCustomers();
    }
    public void updateCustomer(CustomerVO vo) {
        dao.updateCustomer(vo);
    }
    public void deleteCustomer(int custId) {
        dao.deleteCustomer(custId);
    }
}
```



```
@Mapper
@Repository
public class BookDAO {

    @Autowired
    private SqlSessionTemplate mybatis;

    public void insertBook(BookVO vo) {
        mybatis.insert("insertBook", vo);
    }
    public BookVO selectBook(int bookId) {
        return mybatis.selectOne("selectBook", bookId);
    }
    public List<BookVO> selectBooks() {
        return mybatis.selectList("selectBooks");
    }
    public void updateBook(BookVO vo) {
        mybatis.update("updateBook", vo);
    }
    public void deleteBook(int bookId) {
        mybatis.delete("deleteBook", bookId);
    }
}
```

```
@Mapper
@Repository
public class CustomerDAO {

    @Autowired
    private SqlSessionTemplate mybatis;

    public void insertCustomer(CustomerVO vo) {
        mybatis.insert("insertCustomer", vo);
    }
    public CustomerVO selectCustomer(int custId) {
        return mybatis.selectOne("selectCustomer", custId);
    }
    public List<CustomerVO> selectCustomers() {
        return mybatis.selectList("selectCustomers");
    }
    public void updateCustomer(CustomerVO vo) {
        mybatis.update("updateCustomer", vo);
    }
    public void deleteCustomer(int custId) {
        mybatis.delete("deleteCustomer", custId);
    }
}
```


문제3-기능 구현

VO (book, customer)

```

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class BookVO {

    private int bookId;
    private String bookName;
    private String publisher;
    private int price;
}

```

```

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
public class CustomerVO {

    private int custId;
    private String name;
    private String address;
    private String phone;
}

```

BookStoreApplication, application.properties

```

@MapperScan("kr.co.bookstore.persistence")
@SpringBootApplication
public class BookStoreApplication {

    public static void main(String[] args) {
        SpringApplication.run(BookStoreApplication.class, args);
    }

}

```

```
server.servlet.context-path=/BookStore
```

```
server.port=8080
```

```
spring.thymeleaf.cache=false
```

```
# Mybatis 설정
```

```
mybatis.mapper-locations=classpath:mapper/**/*.xml
```

```
# 데이터베이스 정보
```

```
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/java2_bookstore
```

```
spring.datasource.username=root
```

```
spring.datasource.password=1234
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="book">
  <insert id="insertBook">
    insert into `book`(`bookName`, `publisher`, `price`) values(#{bookName}, #{publisher}, #{price});
  </insert>
  <select id="selectBook" resultType="kr.co.bookstore.vo.BookVO">
    select * from `book` where `bookId`=#{bookId};
  </select>
  <select id="selectBooks" resultType="kr.co.bookstore.vo.BookVO">
    select * from `book`;
  </select>
  <update id="updateBook">
    update `book` set
      `bookName`=#{bookName},
      `publisher`=#{publisher},
      `price`=#{price}
    where
      `bookId`=#{bookId};
  </update>
  <delete id="deleteBook">
    delete from `book` where `bookId`=#{bookId};
  </delete>
</mapper>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="customer">
  <insert id="insertCustomer">
    insert into `customer`(`name`, `address`, `phone`) values(#{name}, #{address}, #{phone});
  </insert>
  <select id="selectCustomer" resultType="kr.co.bookstore.vo.CustomerVO">
    select * from `customer` where `custId`=#{custId};
  </select>
  <select id="selectCustomers" resultType="kr.co.bookstore.vo.CustomerVO">
    select * from `customer`;
  </select>
  <update id="updateCustomer">
    update `customer` set
      `name`=#{name},
      `address`=#{address},
      `phone`=#{phone}
    where
      `custId`=#{custId};
  </update>
  <delete id="deleteCustomer">
    delete from `customer` where `custId`=#{custId};
  </delete>
</mapper>
```

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	그냥사는남자	매미스포츠	13000	수정 삭제
2	축구아는 남자	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 단계별기술	굿스포츠	6000	수정 삭제
7	축구알아?	레버너	8000	수정 삭제
9	올림픽 이야기	그린그림	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
20	축구사랑	나는축구러버	15000	수정 삭제
22	테니스왕자	테니스공주	6000	수정 삭제

도서등록

처음으로 도서목록

도서명	배구의 역사
출판사	하이큐
가격	28000
<input type="button" value="등록"/>	

도서수정

처음으로 도서목록

도서번호	1
도서명	그냥사는스포츠맨
출판사	매미스포츠
가격	13000
<input type="button" value="수정"/>	

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	그냥사는스포츠맨	매미스포츠	13000	수정 삭제
2	축구아는 남자	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 단계별기술	굿스포츠	6000	수정 삭제
7	축구알아?	레버너	8000	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
20	축구사랑	나는축구러버	15000	수정 삭제
22	테니스왕자	테니스공주	6000	수정 삭제
23	배구의 역사	하이큐	28000	수정 삭제

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박민성	영국 맨체스타	100-1000-1111	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
6	홍길동	부산시 동래구	010-1234-1234	수정 삭제
10	이미자	경기도 성남시	010-0000-8888	수정 삭제

고객등록

처음으로 고객목록

고객명	강감찬
주소	서울시 동작구
휴대폰	010-8888-7777
등록	

고객수정

처음으로 고객목록

고객번호	1
고객명	박지성
주소	영국 맨체스타
휴대폰	010-1818-9999
수정	

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	010-1818-9999	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
10	이미자	경기도 성남시	010-0000-8888	수정 삭제
11	강감찬	서울시 동작구	010-8888-7777	수정 삭제

이상으로 보고서를 마치겠습니다. 감사합니다.