도서관 시스템

3조

차례

주요 클래스
구성

Q&A
기타

1
주요 클래스 구성

주요 클래스 관계도


APP
Library
Service
Impl
Services
싱글톤
Library
Storage
싱글톤
Book
Manage
Impl
Book
State
Impl
Book
Transaction
Impl
User
Manage
Impl
User
Viewer
Impl

VO 클래스 구성

UserVO

- 아이디
- 비밀번호
- 이름
- 이메일

BookVO

- ISBN13
- 도서명
- 글쓴이
- 출판사
- 출시일
- 재고
- 분야명

Book Manage VO

- ISBN13
- 대여자 아이디
- 대여일
- 반납일

주요 클래스 구성

- 최상단 클래스로 앱의 실행을 담당

주요 클래스 구성

Library Service Impl

- 사용자 인터페이스(UI) 역할 수행
- 사용자/관리자에게 별도의 메뉴를 출력
- 각각 분할하여 작성한 클래스(5개) 객체의 메서드를 호출

주요 클래스 구성

싱글톤

- 작성한 클래스(5개)의 객체를 각각 1개씩 참조 (필드변수)
- 사용자 등록정보, 대여-반납 정보, 책 정보 3개의 컬렉션 보관

주요 클래스 구성

- 도서 추가
- 도서 삭제
- 도서 검색 (도서명 / ISBN13)

주요 클래스 구성

- 도서 대여 기록 조회
- 대여 중인 도서 조회 (미반납)
- 반납한 도서 목록 조회

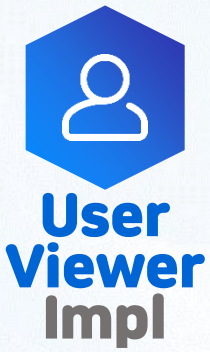
주요 클래스 구성

- 도서 대여 작업
- 도서 반납 작업

주요 클래스 구성

- 회원가입
- 로그인/로그아웃
- 사용자 정보수정
- 탈퇴

주요 클래스 구성



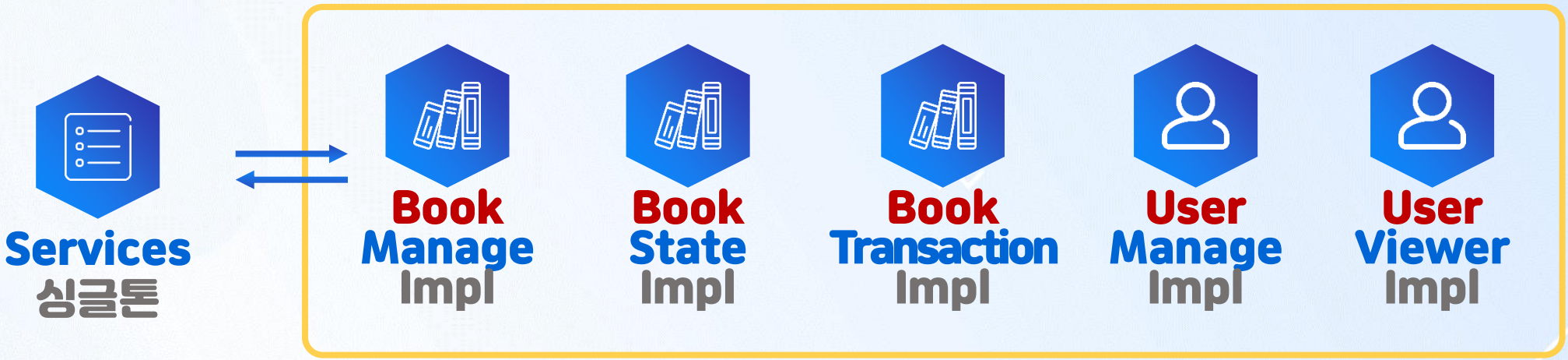
- 회원 목록 조회
- 아이디 검색

주요 클래스 구성

- 사용자 목록, 도서 목록, 대여/반납 목록 컬렉션 보유
- 로그인한 사용자의 세션 정보 보관 (UserVO객체)
- 사용자 ID로 검색하여 UserVO 객체로 반환
- ISBN13, ID로 검색하여 미반납 도서 목록 반환
- 대여/반납 객체로 도서 정보 반환

주요 클래스 구성

- 각 기능 클래스의 객체를 **1개씩** 보관



- LibraryService클래스나 각 기능 클래스에서 필요한 객체를 불러와 사용할 수 있음.

LibraryService 메서드 구성

```
public interface LibraryService {

    public void entrance();  로그인 상태에 따라 메뉴별로 호출 (무한 루프)

    public boolean showDefaultMenu();   로그인/회원가입 화면

    public boolean showAdminMenu();     관리자 로그인 후

    public boolean showUserMenu();      일반 사용자 로그인 후

}
```

BookManage 메서드 구성

```
public interface BookManage {
    public void insertBook();     도서 등록
    public void updateBook();     도서 수정
    public void deleteBook();     도서 삭제
    public void findByISBN();     ISBN으로 도서 검색
    public void findByTitle();    제목으로 도서 검색
    public void listBook();       등록된 도서 목록 출력
 }
```

BookState 메서드 구성

```
public interface BookState {
    public void allList();              모든 대여 목록 조회
    public void borrowList();           대여 중인 목록 조회
    public void returnList();           반납한 목록 조회
    public void findId();               아이디로 이력 조회 (Scanner 입력)
    public void findId(String id);      아이디로 이력 조회 (오버로딩)
}
```

BookTransaction 메서드 구성

```
public interface BookTransaction {
    public BookManageVO rentalBook();  도서 대여 요청
    public BookManageVO returnBook(); 도서 반환 요청 (Scanner 입력)
    public BookManageVO returnBook(BookManageVO vo);          도서 반환 요청 (매개변수 전달)
    public BookManageVO returnBook(String code, Date endDate);  도서 반환 요청 (실제 처리 구문)
}
```

UserManage 메서드 구성

```
public interface UserManage {
    public static final String ADMIN_ID = "admin";    관리자용 아이디 상수 설정
    public static final String ADMIN_PW = "1234";     관리자용 암호 상수 설정

    public void join();                               회원가입
    public void login();                              로그인
    public void logout();                             로그아웃 => 로그아웃(true)
    public void logout(boolean isSilent);             로그아웃 (메시지 출력여부) / 탈퇴 시 조용히 로그아웃 처리
    public void update();                             정보수정
    public void out();                                회원탈퇴
}
```

UserViewer 메서드 구성

```
public interface UserViewer {

    public void printUsers();          회원 목록 조회

    public void findUserById();        아이디로 회원정보 조회

}
```

2
Q&A
기타

Q&A

Library S···

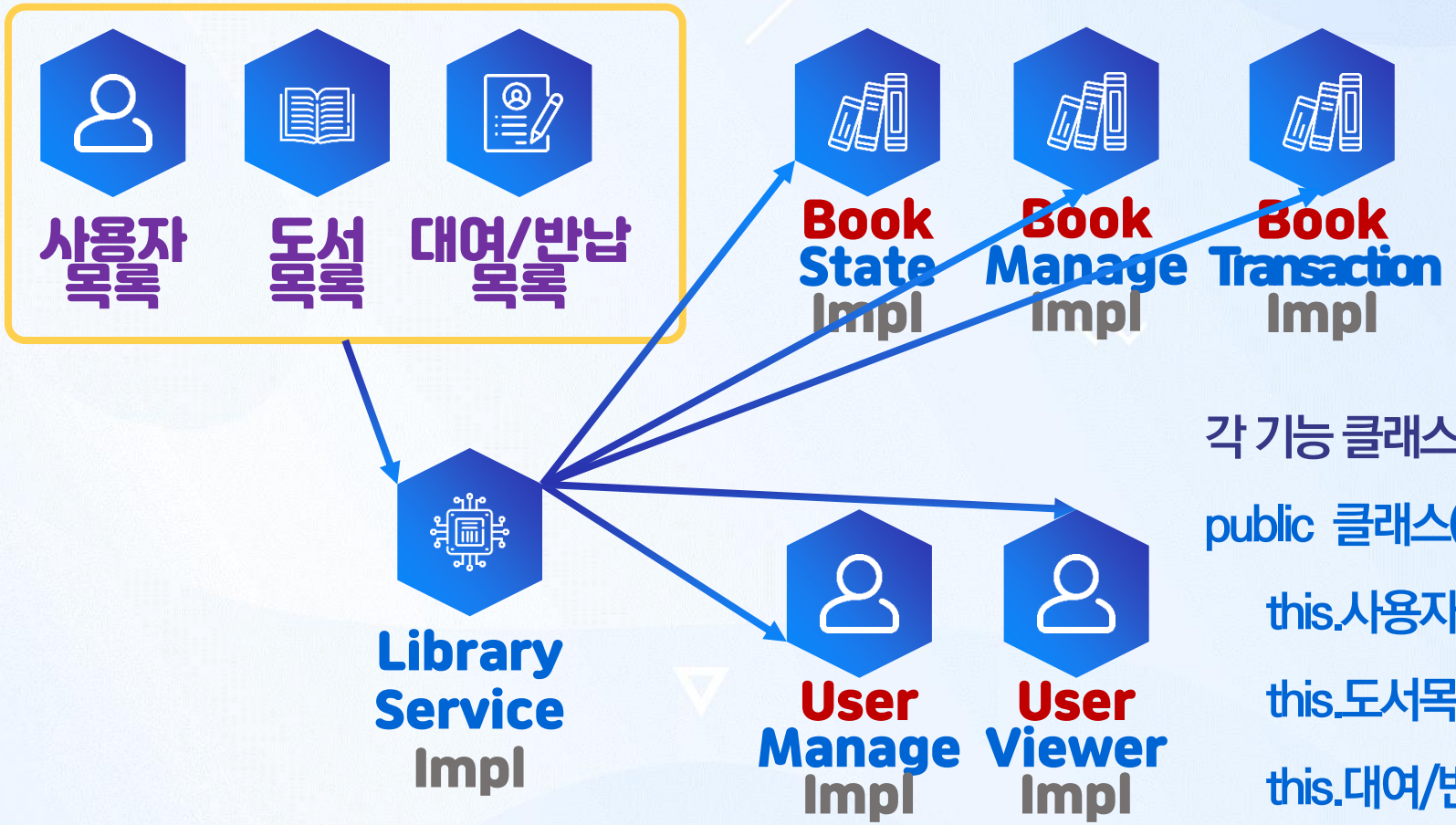
Services

싱글톤

사용한 이유?

싱글톤을 사용하지 않았다면? (생성자로 전달)

- LibraryService에 인스턴스 변수로 선언 후 생성자를 통해 각 클래스 매개변수로 전달



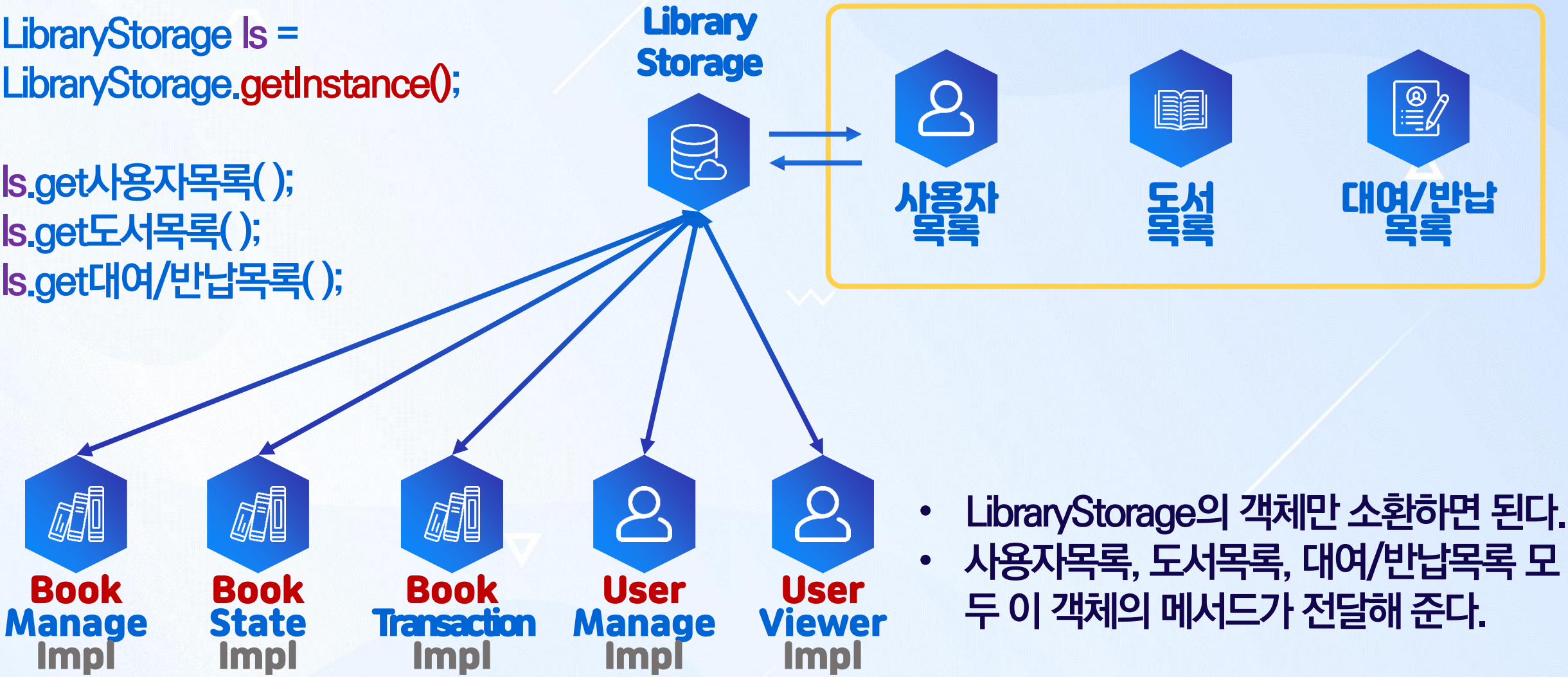
각 기능 클래스에서 모두 선언해야…

```
public  클래스(사용자목록,도서목록,대여/반납목록){
    this.사용자목록 = 사용자목록;
    this.도서목록 = 도서목록;
    this.대여/반납목록 = 대여/반납목록;
}
```

싱글톤을 사용했을 때의 LibraryStorage (목록 저장소 ≒ DB)

```
LibraryStorage ls =
LibraryStorage.getInstance();

ls.get사용자목록( );
ls.get도서목록( );
ls.get대여/반납목록( );
```



- LibraryStorage의 객체만 소환하면 된다.
- 사용자목록, 도서목록, 대여/반납목록 모두 이 객체의 메서드가 전달해 준다.

싱글톤을 사용했을 때의 Services (각 기능을 수행하는 클래스를 저장)

- Services 객체만 소환하면 된다.
- 각 기능 클래스끼리도 책 대여/반납진행 객체에서 책 대여목록 조회 기능을 편리하게 호출
- 일일이 중심부에서 생성자로 전달할 필요 없음


```
Services ss = Services.getInstance();
책 대여목록 조회 a = ss.get책 대여목록 조회( );
책 대여/반납 진행 b = ss.get책 대여/반납진행( );
```

감사합니다

https://github.com/ineggapps/java-2020-academy-sslibrary