

<기능 명세 - 도서관리시스템(BookManagementSystemProject)>

1) Main

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo	
클래스 이름	BookManagementSystemMain	
메소드	public static void main(String[] args)	

2) Model

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.model	
클래스 이름	User	
상수	private String id	
	private String password	
	private String name	
	private String birth	//TODO::Date로 변경해보기
	private String email	
	private int phoneNumber	

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.model	
클래스 이름	Manager	
상위클래스	User	

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.model	
클래스 이름	Book	
상수	private String isbn	
	private String title	
	private String author	
	private String publisher	
	private boolean isBorrow	

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.model	
클래스 이름	Borrow	
상수	private String isbn	
	private String title	
	private String userId	
	private Date borrowDate	
	private Date returnDate	

### 3) Controller

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.controller	
인터페이스 이름	Controller	
상수	ScannerIOUtil io	

분류	식별자	<부연설명>											
패키지 이름	kr.ac.kopo.controller												
클래스 이름	AccountController												
상위인터페이스	Controller												
메소드	public void signup()	1) 관리자 여부를 입력받음 <table><tr><td>&lt;Console 창&gt;</td></tr><tr><td>관리자인가요?(y/n) :</td></tr></table> 2-1) 관리자일 경우 Manager 인스턴스의 멤버변수를 채워 UserServie.signup(User User)을 호출 2-2) 회원일 경우 User 인스턴스의 멤버변수를 채워 UserServie.signup(User User)을 호출 <table><tr><td>&lt;Console 창&gt;</td></tr><tr><td>2-1), 2-2) 동일</td></tr><tr><td>아이디 :</td></tr><tr><td>비밀번호 :</td></tr><tr><td>비밀번호 재입력 :</td></tr><tr><td>이름 :</td></tr><tr><td>생년월일(ex.1996.03.08.) :</td></tr><tr><td>이메일 :</td></tr><tr><td>폰번호 :</td></tr></table> 3) 결과 출력 3-1) 비밀번호 2개의 입력값이 같지 않을 경우 에러 출력 3-2) 아이디가 공백일 경우 에러 출력 3-3) 회원가입 성공 여부 출력	<Console 창>	관리자인가요?(y/n) :	<Console 창>	2-1), 2-2) 동일	아이디 :	비밀번호 :	비밀번호 재입력 :	이름 :	생년월일(ex.1996.03.08.) :	이메일 :	폰번호 :
	<Console 창>												
관리자인가요?(y/n) :													
<Console 창>													
2-1), 2-2) 동일													
아이디 :													
비밀번호 :													
비밀번호 재입력 :													
이름 :													
생년월일(ex.1996.03.08.) :													
이메일 :													
폰번호 :													
	public User login()	1) 아이디와 비밀번호를 입력받아 UserServie.getUser(...)을 호출  2-1) 해당 계정을 가지고 있다면 로그인 성공 2-2) 없다면 예외처리 후 로그인 종료											

		<div> <div>&lt;Console 창&gt;</div> <div>아이디 :</div> <div>비밀번호 :</div> <div>로그인 되었습니다.</div> <div>입력하신 정보가 맞지 않습니다.</div> </div> <div>3) 반환된 User가 Manager 여부에 따라 메뉴 진행</div> <div>3-1) 관리자 : ManagerMenu.process()</div> <div>3-2) User : UserMenu.process()</div>
	public boolean resign	<div>1) UserService에서 회원 삭제</div> <div>2) 결과 출력</div> <div>2-1) 해당 회원이 존재하지 않을 경우 예러 출력</div> <div>2-2) 대여 중인 도서가 있을 경우 예러 출력</div> <div>2-3) 성공여부 출력</div>

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.controller	
인터페이스 이름	MenuController	
상위인터페이스	Controller	
추상메소드	public void process()	

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.controller	
클래스 이름	AccountMenuController	사용자로부터 회원가입, 로그인, 종료를 선택받고 그에 따른 일을 수행함.
상위인터페이스	MenuController	
메소드		1) 사용자로부터 1~3 정수를 입력받는다.
		<div style="border: 1px solid black; padding: 5px;">           &lt;Console 창&gt;            안녕하세요.            도서관리스시스템입니다.            항목을 선택하세요(1.회원가입            2.로그인 3.종료) :         </div>
		2-1) 1을 입력할 경우 회원가입 진행 2-2) 2를 입력할 경우 로그인 진행 2-3) 3을 입력할 경우 프로그램 종료
	public void process()	
	private void signup()	AccountController.signup() 실행
	private void login()	AccountController.login() 실행

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.controller	
클래스 이름	UserMenuController	사용자로부터 마이페이지, 도서검색, 대여, 반납, 로그아웃을 선택받고 그에 따른 일을 수행함.
상위인터페이스	MenuController	
상수	User user	현재 로그인 중인 User
메소드		1) 사용자에게 관련된 화면을 출력한다.
		<div style="border: 1px solid black; padding: 5px; margin: 5px 0;">           &lt;Console 창&gt;            항목을 선택하세요(1.마이페이지 2.도서검색 3.대여 4.반납 5.로그아웃) :         </div>
	public void process()	2) 사용자로부터 1~5 정수를 입력받는다.
		3-1) 1을 입력할 경우 마이페이지 (개인정보 및 대여목록) 출력 3-2) 2를 입력할 경우 도서검색 진행 3-3) 3을 입력할 경우 대여 진행 3-4) 4를 입력할 경우 반납 진행 3-5) 5를 입력할 경우 로그아웃 진행  4) 로그아웃이 아니라면 1)~3) 반복
	private void myPage()	1) 개인정보와 대여목록을 출력 2) 개인정보수정여부 선택 <div style="border: 1px solid black; padding: 5px; margin: 5px 0;">             &lt;Console 창&gt;              개인정보를 수정하시겠습니까?              (y/n) :           </div> 2-1) y면 ModifyPrivateInfo 메뉴로 이동
	private void searchBook()	1) 검색어를 입력받고 BookService에서 검색어를 포함하는 도서 목록 반환  2) 반환된 값을 PrintListUtil 클래스를 사용하여 출력
	private boolean borrowBook()	1) 사용자에게 ISBN을 입력받아 BorrowService에서 대여 처리

		2) 결과 출력 2-1) 입력받은 ISBN이 없을 경우 에러 출력 2-2) 이미 대여 중인 도서일 경우 에러출력 2-3) 대여 성공 여부 출력
	private boolean returnBook()	1) 사용자에게 ISBN을 입력받아 BorrowService에서 반납 처리  2) 결과 출력 2-1) 입력받은 ISBN이 대여 목록에 없을 경우 에러 출력 2-3) 대여 성공 여부 출력

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.controller	
인터페이스 이름	ModifyPrivateInfoController	
상위인터페이스	MenuController	
추상메소드	public void process()	1) 사용자로부터 1~5 정수를 입력받는다. <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> &lt;Console 창&gt;  수정하고 싶은 항목을  선택하세요(1.비밀번호 2.이메일  3.전화번호 4.회원탈퇴  5.이전메뉴) : </div> 2-1) 1은 비밀번호 수정 진행 2-2) 2는 이메일 수정 진행 2-3) 3은 전화번호 수정 진행 2-4) 회원탈퇴 진행 후 로그아웃
	private void modifyPwd()	1) 비밀번호를 입력받아 UserService 클래스로 비밀번호 수정 2) 결과 출력 2-1) 입력받은 2개의 비밀번호가 일치 하지 않을 경우 에러 출력 2-2) 비밀번호 수정 성공 여부 출력
	private void modifyEmail()	1) 이메일을 입력받아 UserService 클래스로 이메일 수정 2) 이메일 수정 성공 여부 출력
	private void modifyPhone()	1) 전화번호 입력받아 UserService 클래스로 전화번호 수정 2) 전화번호 수정 성공 여부 출력

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.controller	
클래스 이름	ManagerMenuController	사용자로부터 회원가입, 로그인, 종료를 선택받고 그에 따른 일을 수행함.
상위클래스	UserMenuController	
메소드	public void process()	<p>1) 관리자에게 관련된 화면을 출력한다. 관리자도 도서를 대여할 수 있다.</p> <div data-bbox="898 631 1340 828" data-label="Text"> <p>&lt;Console 창&gt;  항목을 선택하세요(1.마이페이지  2.도서검색 3.도서대여 4.도서반납  5.회원관리 6.도서관리  7.로그아웃) :</p> </div> <p>2) 사용자로부터 1~7 정수를 입력받는다.</p> <p>3-1) 1~4는 UserMenuController와 동일하게 진행  3-2) 5는 회원관리 진행  UserManagementMenuController.process()  3-2) 6은 도서관리 진행  BookManagementMenuController.process()  3-3) 7을 입력할 경우 로그아웃</p> <p>4) 로그아웃이 아니라면 1)~3) 반복</p>

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.controller	
클래스 이름	UserManagmentMenuController	사용자로부터 회원가입, 로그인, 종료를 선택받고 그에 따른 일을 수행함.
상위인터페이스	MenuController	
메소드	public void process()	<p>1) 사용자에게 관련된 화면을 출력한다.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>&lt;Console 창&gt;</p> <p>항목을 선택하세요(1.회원등록 2.회원삭제 3.회원검색 4.관리자승인) :</p> </div> <p>2-1) 1을 입력할 경우 회원등록 진행 2-2) 2를 입력할 경우 회원삭제 진행 2-3) 3을 입력할 경우 회원검색 진행 2-4) 4를 입력할 경우 대기 중인 관리자 승인 진행</p>
	private boolean addUser()	<p>1) 등록할 회원 수를 입력받음 2) 입력받은 수만큼 AccountController.signup() 진행</p>
	private boolean removeUser()	<p>1) 삭제할 회원 수를 입력받음 2) 입력받은 수만큼 회원 삭제 진행 3) 결과 출력</p> <p>3-1) 해당 아이디가 존재하지 않을 경우 에러 출력 3-2) 해당 아이디가 대여한 도서가 있을 경우 에러 출력 3-3) 회원삭제 성공 여부 출력</p>
	private boolean searchUser()	<p>1) 검색어를 입력받고 UserService에서 검색어를 포함하는 회원 목록 반환</p> <p>2) 반환된 값을 PrintListUtil 클래스를 사용하여 출력</p>
	private boolean managerApproval()	<p>1) 관리자로 회원가입한 회원을 승인... (NOTE::TODO)</p>



#### 4) Service

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.service	
클래스 이름	<a href="#">UserService</a>	계정과 관련된 서비스 및 회원등록, 삭제 검색 서비스를 제공
메소드	public boolean isUser()	입력된 아이디와 비밀번호를 가진 회원의 존재 여부 확인
	public ErrorType modifyPwd(User user, String nerPwd1, String newPwd2)	비밀번호 수정 2) 수정 성공 여부 반환 2-1)입력된 두 비밀번호 일치 여부 예러
	public ErrorType modifyEmail(User user, String newEmail)	이메일 수정 수정 성공 여부 반환
	public ErrorType modifyPhoneNumber(User user, String newPhoneNumber)	전화번호 수정 수정 성공 여부 반환
	public ErrorType checkId(String id)	새로운 아이디 입력 값이 정상인지 확인 1) 공백일 경우 예러 2) 존재하는 아이디일 경우 예러
	public boolean containID(String id)	아이디가 존재 여부 반환
	public ErrorType addUser(User e)	회원 저장
	public ErrorType addUser(User newUser, String pwd2)	새로운 회원 저장
	public User remove(User e)	id를 가지는 회원 삭제
	public ErrorType removeUser(String id)	id를 가지는 회원 삭제
	public ArrayList<User> search(String value)	해당 value를 info에 포함하는 모든 User return
	public User getUser(String id)	해당 id를 가진 회원 반환
	public User getUser(String id, String pwd)	해당 id와 pwd를 가진 회원 반환

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.service	
클래스 이름	BookService	도서저장, 삭제 검색 서비스를 제공
메소드	public ErrorType addBookCheckISBN(String isbn)	새로운 도서의 ISBN 입력 값이 정상인지 확인 1) 공백일 경우 에러 2) 존재하는 ISBN일 경우 에러
	public boolean containISBN(String isbn)	isbn을 가진 도서 존재 여부 반환
	public ErrorType addBook(Book e)	도서 저장
	public ErrorType removeBook(User e)	해당하는 isbn을 가지는 도서 삭제
	public ErrorType removeBook(String isbn)	해당하는 isbn을 가지는 도서 삭제 1) 존재하지 않은 isbn일 경우 에러 2) 대여 중인 isbn일 경우 에러
	public ArrayList<Book> searchBook(String value)	해당 value를 info에 포함하는 모든 Book return
	public Book getBook(String isbn)	해당 isbn을 가진 도서 반환

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.service	
클래스 이름	BorrowService	대여 대출 대출현황제공 서비스 제공
메소드	public ErrorType borrowBook(String isbn, String userId)	새로운 도서의 ISBN 입력 값이 정상인지 확인 1) 공백일 경우 에러 2) 존재하는 ISBN일 경우 에러
	public ErrorType returnBook(String isbn, String userId)	isbn을 가진 도서 반환 1) 대여목록에 존재하지 않다면 에러 2) 현재 로그인 중인 회원이 대여한 것이 아니라면 에러
	public ErrorType addBook(Book e)	도서 저장
	public boolean containISBN(String isbn)	isbn을 가진 도서 존재 여부 확인
	public ArrayList<Book> searchBorrow(String value)	해당 value를 info(회원id, 책 제목, 대여일, 반납일)에 포함하는 모든 Borrow return
	public ArrayList<Borrow> searchBorrowWithID	회원id가 대여한 모든 Borrow return
	public Book getBorrow(String isbn)	해당 isbn을 가진 도서 반환

## 5) Data Access

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.da	
인터페이스 이름	MapDA<K, V>	
추상메소드	public boolean add(K k, V v)	
	public V remove(K k)	
	public V get(K k)	
	public List<V> getList(String value)	

### - inMemory

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.da.inMemory	
클래스 이름	UserDA	계정을 저장하고 있는 저장소
상위인터페이스	MapDA<String, User>	String: 아이디
상수	private Map<String, User> accountMap	회원 저장
메소드	public boolean add(String k, User v)	k를 key로 v를 저장
	public User remove(String k)	k를 key로 v를 삭제
	public User get(String k)	k를 key로 v를 반환
	public ArrayList<User> getList(String value)	value를 매개변수로 포함하는 회원목록 모두 반환

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.da.inMemory	
클래스 이름	BookDA	도서를 저장하고 있는 저장소
상위인터페이스	MapDA<String, Book>	String: ISBN
상수	private Map<String, Book> bookMap	Book 저장
메소드	public boolean add(String k, Book v)	k를 key로 v를 저장
	public Book remove(String k)	k를 key로 v를 삭제
	public Book get(String k)	k를 key로 v를 반환
	public ArrayList<Book> getList(String value)	value를 매개변수로 포함하는 도서목록 모두 반환
	public boolean containISBN(String k)	해당 isbn이 존재하는지 여부 반환

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.da.inMemory	
클래스 이름	BorrowDA	대여 목록을 저장하고 있는 저장소
상위인터페이스	MapDA<String, Borrow>	String ISBN, String: 아이디
상수	private Map<String, Borrow> borrowMap	대여 목록 저장
메소드	public boolean add(String k, Borrow v)	k를 key로 v를 저장
	public String remove(String k)	k를 key로 v를 삭제
	public Book get(String k)	k를 key로 v를 반환
	public ArrayList<String> getList(String value)	value를 매개변수로 포함하는 대여목록 모두 반환
	public boolean containsISBN(String k)	해당 isbn이 존재하는지 여부 반환

- inFile

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.da.inFile	
클래스 이름	UserDA	계정을 저장하고 있는 저장소
상위인터페이스	MapDA<String, User>	String: 아이디
상수	private Map<String, User> accountMap	회원 저장
메소드	public boolean add(String k, User v)	k를 key로 v를 저장 변경된 map을 파일로 저장
	public User remove(String k)	k를 key로 v를 삭제 변경된 map을 파일로 저장
	public User get(String k)	k를 key로 v를 반환
	public ArrayList<User> getList(String value)	value를 매개변수로 포함하는 회원목록 모두 반환

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.da.inFile	
클래스 이름	BookDA	도서를 저장하고 있는 저장소
상위인터페이스	MapDA<String, Book>	String: ISBN
상수	private Map<String, Book> bookMap	Book 저장
메소드	public boolean add(String k, Book v)	k를 key로 v를 저장 변경된 map을 파일로 저장
	public Book remove(String k)	k를 key로 v를 삭제 변경된 map을 파일로 저장
	public Book get(String k)	k를 key로 v를 반환
	public ArrayList<Book> getList(String value)	value를 매개변수로 포함하는 도서목록 모두 반환
	public boolean containsISBN(String k)	해당 isbn이 존재하는지 여부 반환

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.da.inFile	
클래스 이름	BorrowDA	대여 목록을 저장하고 있는 저장소
상위인터페이스	MapDA<String, Borrow>	String ISBN, String: 아이디
상수	private Map<String, Borrow> borrowMap	대여 목록 저장
메소드	public boolean add(String k, Borrow v)	k를 key로 v를 저장 변경된 map을 파일로 저장
	public String remove(String k)	k를 key로 v를 삭제 변경된 map을 파일로 저장
	public Book get(String k)	k를 key로 v를 반환
	public ArrayList<String> getList(String value)	value를 매개변수로 포함하는 대여목록 모두 반환
	public boolean containISBN(String k)	해당 isbn이 존재하는지 여부 반환

#### 6) type

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.type	
enum 이름	ErrorType	Scanner를 이용해 사용자와 값을 주고 받음
상수	NOTEQUALPWD,	패스워드가 일치하지 않음
	OUTOFFORM,	입력된 값의 형식이 맞지 않음
	EXIST,	목록에 존재
	NOTEXIST,	목록에 존재하지 않음
	BORROWED,	대여 중
	FAIL,	실패
	SUCCESS,	성공

## 7) Util

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.util	
클래스 이름	ScannerIOUtil	Scanner를 이용해 사용자와 값을 주고 받음
상수	private static final Scanner sc	
메소드	public int getInt(String msg)	msg를 출력하고 int값을 받음
	public char getChar(String msg)	msg를 출력하고 char값을 받음
	public String getString(String msg)	msg를 출력하고 String값을 받음
	public void print(String msg)	msg 출력
	public void println(String msg)	msg 출력 후 줄띄움

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.util	
클래스 이름	PrintLibrarySystemListUtil	도서관리시스템에서 목록을 출력할 경우 사용
상수	ScannerIOUtil su	
메소드	public void printUserList(ArrayList<User> list)	사용자 목록을 출력
	public void printBookList(ArrayList<Book> list)	도서 목록을 출력
	public void printBorrowList(ArrayList<Borrow> list)	대여 목록 출력

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.util	
클래스 이름	PrintBoundaryUtil	도서관리시스템에서 목록을 출력할 경우 사용
상수	ScannerIOUtil su	
메소드	public void boundaryOfMenuStart()	메뉴 시작 구분선 출력
	public void boundaryOfMenuEnd()	메뉴 끝 구분선 출력
	public void boundaryOfWork()	메뉴 단락 구분선 출력

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.util	
클래스 이름	FileClose	
메소드	public static void close(InputStream is)	
	public static void close(OutputStream os)	
	public static void close(InputStream fis, InputStream is)	
	public static void close(OutputStream fos, OutputStream os)	
	public static void close(Reader r)	
	public static void close(Writer w)	
	public static void close(Reader fr, Reader r)	
	public static void close(Writer fw, Writer w)	

분류	식별자	<부연설명>
패키지 이름	kr.ac.kopo.util	
클래스 이름	FileReadWrite	
메소드	public static boolean write (String filename, Map<?, ?> map)	
	public static Object read(String filename)	