

[문제 2] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

MVC패턴을 이용한 ArrayList 문제이다. 해당 구현 클래스 다이어그램과 클래스 구조를 참고하여 프로젝트를 완성하시오

### 1. 구현 클래스 다이어그램 (Class Diagram)

Book
- bNo : int      // 도서 번호 - category : int    // 장르 분류 번호 - title : String    // 도서 제목 - author : String   // 도서 저자
+ Book() + Book(category:int, title:String, author:String) + setter() / getter() + toString() : String

Run
+ <u>main(args:String[]) : void</u>

BookMenu
- sc : Scanner - bm : BookManager
+ BookMenu() + mainMenu() : void + inputBook() : Book + inputBookNo() : int + inputBookTitle() : String

BookDao
- bookList : ArrayList<Book>
+ BookDao() + BookDao(list:ArrayList<Book>) + getLastBookNo() : int + addBook(book:Book) : void + deleteBook(no:int) : int + searchBook(title:String) : int + selectBook(index:int) : Book + selectAll() : ArrayList<Book> + sortedBookList() : ArrayList<Book>

BookManager
- bd : BookDao - sc : Scanner
+ BookManager() + addBook(book:Book) : void + deleteBook(no:int) : int + searchBook(title:String) : int + selectBook(index:int) : Book + selectAll() : ArrayList<Book> + sortedBookList() : Book[] + printBookList(br:Book[]) : void

\* Book클래스의 setter() / getter() 메소드는 직접 구현

\* java.util.Comparator 인터페이스를 상속받은 정렬용 클래스들 작성함

→ AscCategory, DescCategory

### 3. 구현 클래스 설명

Package명	Class명	Method	설명
com.kh.collection.silsub2. run	Run	+main(args:String[]) : <u>void</u>	BookMenu 클래스 객체 생성하 고 mainMenu() 메소드 실행
com.kh.collection.silsub2. view	BookMenu	+BookMenu()	기본 생성자
		+mainMenu() : void	도서관리 프로그램에 해당하는 메인 메뉴 출력, 각 메뉴에 해당 하는 BookManager 클래스의 메 소드 실행 → 반복 출력되게 함
		+inputBook() : Book	Book 객체의 필드 값을 키보드 로 입력 받아 초기화 하고 객체 리턴
		+inputBookNo() : int	삭제할 도서번호를 키보드로 입 력 받아 리턴
		+inputBookTitle() : String	검색할 도서제목을 키보드로 입 력 받아 리턴

\* 위와 같이 추가, 삭제, 검색에 필요한 정보는 키보드로 입력 받도록 각각의 메소드 따로 구현

Package명	Class명	Method	설명
com.kh.collection.silsub2. controller	BookManager	+BookManager()	기본 생성자
		+addBook(book:Book) : void	전달받은 Book객체의 도서번호 를 setter메소드를 통해 설정 → BookDao의 addBook() 메소드로 해당 Book객체 전달
		+deleteBook(no:int) : int	전달받은 도서 번호를 BookDao 의 deleteBook() 메소드로 전달하 고 그 결과 값을 리턴
		+searchBook(title:String) : int	전달받은 도서 제목을 BookDao 의 searchBook() 메소드로 전달 하고 그 결과 값(해당 도서 인덱 스) 리턴
		+selectBook(index:int) : Book	전달받은 인덱스를 BookDao의 selectBook() 메소드로 전달하고 그 결과 값(해당 도서 객체) 리턴
		+selectAll() : ArrayList<Book>	BookDao의 selectAll() 메소드를 통해 전달 받은 도서 전체 리스 트를 리턴

		+sortedBookList() : Book[]	BookDao의 sortedBookList() 메 소드를 통해 전달 받은 정렬된 도서 전체 리스트를 for문을 통 해 Book[] 배열에 넣고 리턴
		+printBookList(br:Book[]) : void	전달 받은 Book[] 배열을 for each문을 이용하여 출력

Package명	Class명	Method	설명
com.kh.collection.silsub2. model.vo	Book	+Book()	기본 생성자
		+toString() : String	필드 값 문자열 합친 후 리턴 category 분류 별로 출력 >> 1 : 인문 / 2 : 자연과학 3 : 의료 / 4 : 기타
com.kh.collection.silsub2. model.comparator	AscCategory	+compare() : int	카테고리순 오름차순 정렬 처리
	DescCategory		카테고리순 내림차순 정렬 처리
com.kh.collection.silsub2. model.dao	BookDao	+BookDao()	기본 생성자 bookList 객체 생성함
		+BookDao(list:ArrayList<Bo ok>)	매개변수 생성자 다른 리스트를 전달 받아 bookList의 초기값으로 사용함
		+getLastBookNo() : int	도서의 마지막 도서 번호 리턴
		+addBook(book:Book) : void	전달받은 도서 객체 bookList에 추가
		+deleteBook(no:int) : int	전달받은 도서 번호와 일치하는 도서 bookList에서 삭제하고 삭 제 성공 시 1 리턴, 실패 시 0 리턴
		+searchBook(title:String) : int	전달받은 도서 제목을 포함하는 bookList의 인덱스 리턴
		+selectBook(index:int) : Book	전달받은 인덱스를 이용하여 bookList의 해당 인덱스 Book 객체 리턴
		+selectAll() : ArrayList<Book>	전체 도서 리스트 리턴
		+sortedBookList() : ArrayList<Book>	카테고리순 오름차순 정렬 클레 스를 이용하여 리스트 정렬 후 리턴

#### 4. class 구조

```
public class BookMenu{
    public void mainMenu() {
        *** 도서 관리 프로그램 ***

        1. 새 도서 추가          // addBook (inputBook()이 리턴한 객체) 실행
        2. 도서정보 정렬 후 출력 // sortedBookList() 실행
                                   => Book[] 리턴 받아 printBookList(Book[]) 실행
        3. 도서 삭제             // deleteBook (inputBookNo()이 리턴한 도서 번호) 실행
                                   => 결과값 리턴 받아 0일 경우 "성공적으로 삭제"
                                   1일 경우 "삭제할 글이 존재하지 않음"
        4. 도서 검색출력         // searchBook (inputBookTitle()이 리턴한 도서 제목) 실행
                                   => index 리턴 받아 -1일 경우 "조회한 글이 존재하지 않음"
                                   -1이 아닐 경우 selectBook(index) 출력
        5. 전체 출력             // selectAll() 실행
                                   => 결과 리스트 리턴 받아 비어있을 경우 "없습니다."
                                   아닐 경우 Iterator를 이용하여 전체 출력
        6. 끝내기               // main()으로 리턴

        메뉴 번호 선택 : >> 입력 받음
        // 메뉴 화면 반복 실행 처리
        // 해당 메뉴 번호에 따라 BookManager 클래스 메소드 실행
    }

    public Book inputBook(){
        // "도서 제목 : " >> 입력 받음
        // "도서 장르 (1:인문 / 2:자연과학 / 3:의료 / 4:기타) : " >> 입력 받음 (숫자로)
        // "도서 저자 : " >> 입력 받음
        // 매개변수 생성자를 이용하여 위의 초기값을 이용한 Book객체 리턴
    }

    public int inputBookNo(){
        // "도서 번호 : " >> 입력 받음 >> 리턴
    }

    public String inputBookTitle() {
        // "도서 제목 : " >> 입력 받음 >> 리턴
    }
}
```

```

public class BookManager{

    public void addBook(Book book) {

        // BookDao의 getLastBookNo() 메소드를 통해 도서의 마지막 도서 번호를 알아옴
        // setter를 이용하여 도서 번호를 마지막 도서 번호 +1 처리
        // 애초에 리스트에 도서가 없는 경우, 즉 첫 도서 등록일 경우 예외 발생
        ➔ 어떤 예외처리가 발생하는지 알아보고 try catch문을 이용하여 오류 해결
        // BookDao의 addBook()메소드에 해당 Book 객체 전달

    }

    public int deleteBook(int no){

        // BookDao의 deleteBook() 메소드에 전달받은 도서 번호 전달
        // 그 결과 값을 받아 리턴

    }

    public int searchBook(String title){

        // BookDao의 searchBook() 메소드에 전달받은 도서 제목 전달
        // 그 결과 값(리스트의 인덱스 값)을 받아 리턴

    }

    public Book selectBook(int index){

        // BookDao의 selectBook() 메소드에 전달받은 인덱스 전달
        // 그 결과 값(해당 Book객체)을 받아 리턴

    }

    public ArrayList<Book> selectAll(){

        // BookDao의 selectAll() 메소드를 통해 도서 전체 리스트를 전달 받아 리턴

    }

    public Book[] sortedBookList(){

        // BookDao의 sortedBookList() 메소드를 통해 정렬된 도서 리스트를 전달 받아
        // for문을 이용하여 Book[] 배열에 추가
        // 해당 배열 주소 값 리턴

    }

    public void printBookList(Book[] br) {

        // for each문을 이용하여 전달받은 Book[] 배열 전체 출력

    }

}

```