

[문제 1] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

해당 구현 클래스 다이어그램과 클래스 구조를 참고하여 프로젝트를 완성하시오

프로젝트명 : chap12-collection-practice-source

1. 구현 클래스 다이어그램 (Class Diagram)

com.ohgiraffers.hw1.model.dto.BookDTO
- bNo : int // 도서 번호 - category : int // 장르 분류 번호 - title : String // 도서 제목 - author : String // 도서 저자
+ BookDTO() + BookDTO(category:int, title:String, author:String) + setter() / getter() + toString() : String

com.ohgiraffers.hw1.run.Application
+ <u>main(args:String[]) : void</u>

com.ohgiraffers.hw1.view.BookMenu
- sc : Scanner - bm : BookManager
+ BookMenu() + mainMenu() : void + inputBook() : BookDTO + inputBookNo() : String + inputBookTitle() : String + selectSortedBook():List<BookDTO>

com.ohgiraffers.hw1.controller.BookManager
- booklist:ArrayList<BookDTO> - sc : Scanner
+ BookManager() + addBook(book:Book) : void + deleteBook(index:int) : void + searchBook(btitle:String) : void + displayAll():void + sortedBookList(type:int):List<BookDTO> + printBookList(printList:List<BookDTO>):void

* java.util.Comparator 인터페이스를 상속받은 정렬용 클래스들 작성함

(패키명: com.ohgiraffers.hw1.comparator)

→ AscBookNo, DescBookNo, AscBookTitle, DescBookTitle

3. 구현 클래스 설명

Package명	Class명	Method	설명
com.ohgiraffers.hw1.run	Application	+main(args:String[]) : <u>void</u>	BookMenu 클래스 객체 생성하고 mainMenu() 메소드 실행
com.ohgiraffers.hw1.view	BookMenu	+BookMenu()	기본 생성자
		+mainMenu() : void	도서관리 프로그램에 해당하는 메인 메뉴 출력, 각 메뉴에 해당하는 BookManager 클래스의 메소드 실행 → 반복 출력되게 한다.
		+inputBook() : BookDTO	BookDTO 객체의 필드 값을 키보드로 입력 받아 초기화 하고 객체 리턴 (카테고리는 (1.인문/2.자연과학/3.의료/4.기타) 으로한다.)
		+inputBookNo() : int	삭제할 도서번호를 키보드로 입력 받아 리턴
		+inputBookTitle() : String	검색할 도서제목을 키보드로 입력 받아 리턴
		+selectSortedBook():List <BookDTO>	도서출력시 정렬방식을 선택

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

Package명	Class명	Method	설명
com.ohgiraffers.hw1.controller	BookManager	+BookManager()	기본 생성자 ArrayList<BookDTO>객체 생성
		+addBook(book:BookDTO) : void	전달받은 도서정보를 필드에선언한 List에 담는다.
		+deleteBook(index:int) : void	전달받은 도서번호를 이용하여 해당하는 인덱스에 도서를 제거한다.
		+searchBook(title:String) : String	전달받은 도서 제목의 도서를 찾고 해당 도서의 정보를 출력한다. 검색된 도서가 없는경우 "조회된 도서가 목록에 없습니다" 출력

		+displayAll():void	도서의 전체목록을 출력하고, list의 개수가 0이면 “조회결과가 없습니다” 출력한다.
		+sortedBookList(type:int) :List<BookDTO>	사용자가 선택한 정렬방식에 따 라서 오름차순/내림차순 정렬하 여 값을 반환한다.
		+printBookList(printList:Li st<BookDTO>):void	정렬된 list객체의 도서를 출력한 다.

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 // printBookList() 실행
                                   => 인자로 selectSortedBook()하여 리턴받은 리스트 객체사용
        3. 도서 삭제             // deleteBook (inputBookNo()이 리턴한 도서 번호) 실행

        4. 도서 검색출력         // searchBook (inputBookTitle()이 리턴한 도서 제목) 실행
                                   => 결과가 있는경우 해당 도서를 출력하고,
                                   결과가 없는 경우 "조회한 도서가 목록에 없습니다."

        5. 전체 출력             // displayAll() 실행
                                   => 비어있을 경우 "출력결과가 없습니다."
                                   비어있지 않은 경우 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 BookDTO 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 BookDTO객체 리턴
    }

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

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

// 이어서 클래스 설명

```
Public List<BookDTO> selectSortedBook() {
```

```
    // 도서 정렬방식을 정수로 입력받아서
```

```
    // 1. 도서번호(ISBN)으로 오름차순정렬
```

```
    // 2. 도서번호(ISBN)으로 내림차순정렬
```

```
    // 3. 책 제목으로 오름차순 정렬
```

```
    // 4. 책 제목으로 내림차순 정렬
```

```
    BookManager 객체의 sortedBookList() 메소드를 이용하여 출력
```

```
}
```

```
}
```

```

public class BookManager{
    public BookManager() {
        리스트 초기화 작업
    }

    public void addBook(BookDTO book) {
        // 해당 BookDOT 객체를 리스트에 담기
    }

    public BookDTO deleteBook(int key){
        //전달받은 위치의 인덱스 값 제거
    }

    public String searchBook(String title){
        // 전달받은 제목으로 리스트에서 조회하여 일치하는 객체가 있으면
        // 해당 도서정보를 출력하고, 없는경우 "조회된 도서가 목록에 없습니다"를 출력
    }

    public void displayAll(){
        // 도서목록 전체 출력
    }

    Public List<BookDTO> sortedBookList(int type) {
        //정렬방식별 오름차순/내림차순 정렬 (switch문이용)
    }

    Public void printBookList(List<BookDTO> printList) {
        // 전달받은 리스트 객체의 내용 출력
    }
}

```