

[문제2] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

이전 문제가 ArrayList관련 프로젝트였다면 이 문제는 ArrayList를 HashMap을 이용하여 푸는 문제이다. ArrayList문제와 아래의 클래스 다이어그램과 클래스 구조를 참고하여 수정해 가면서 어떤 차이가 있는지 확인해보자. 오히려 헷갈린다면 다시 처음부터 구현하자.

1. 구현 클래스 다이어그램 (Class Diagram)

Book
- bNo : String // 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 + compareTo(arg0:Object) : int

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

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

BookDao
- bookMap : HashMap<String, Book>
+ BookDao() + BookDao(map:HashMap<String, Book>) + getLastBookNo() : int + addBook(book:Book) : void + deleteBook(key:String) : Book + searchBook(title:String) : String + selectBook(key:String) : Book + selectAll() : HashMap<String, Book> + sortedBookList() : ArrayList<Book>

BookManager
- bd : BookDao - sc : Scanner
+ BookManager() + addBook(book:Book) : void + deleteBook(key:String) : Book + searchBook(title:String) : String + selectBook(key:String) : Book + selectAll() : HashMap<String, Book> + sortedBookList() : Book[] + printBookList(br:Book[]) : void

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

* Book클래스에서 Comparable 인터페이스를 상속받아 compareTo를 오버라이딩하여 도서 제목 순 오름차순 정렬 구현

3. 구현 클래스 설명

Package명	Class명	Method	설명
com.collection.silsub3.run	Run	+main(args:String[]) : void	BookMenu 클래스 객체 생성하고 mainMenu() 메소드 실행
com.collection.silsub3.view	BookMenu	+BookMenu()	기본 생성자
		+mainMenu() : void	도서관리 프로그램에 해당하는 메인 메뉴 출력, 각 메뉴에 해당하는 BookManager 클래스의 메소드 실행 → 반복 출력되게 함
		+inputBook() : Book	Book 객체의 필드 값을 키보드로 입력 받아 초기화 하고 객체 리턴
		+inputBookNo() : String	삭제할 도서번호를 키보드로 입력 받아 리턴
		+inputBookTitle() : String	검색할 도서제목을 키보드로 입력 받아 리턴

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

Package명	Class명	Method	설명
com.collection.silsub3.controller	BookManager	+BookManager()	기본 생성자
		+addBook(book:Book) : void	전달받은 Book객체의 도서번호를 setter메소드를 통해 설정 → BookDao의 addBook() 메소드로 해당 Book객체 전달
		+deleteBook(key:String) : Book	전달받은 도서 번호를 BookDao의 deleteBook() 메소드로 전달하고 그 결과 값을 리턴
		+searchBook(title:String) : String	전달받은 도서 제목을 BookDao의 searchBook() 메소드로 전달하고 그 결과 값(해당 도서 key) 리턴
		+selectBook(key:String) : Book	전달받은 key를 BookDao의 selectBook() 메소드로 전달하고 그 결과 값(해당 도서 객체) 리턴
		+selectAll() : HashMap<String, Book>	BookDao의 selectAll() 메소드를 통해 전달 받은 도서 전체를 리턴

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

Package명	Class명	Method	설명
com.collection.silsub3.m odel.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
		+compareTo() : int	제목 별 오름차순 내용 구현
com.collection.silsub3.m odel.dao	BookDao	+BookDao()	기본 생성자 bookMap객체 생성함
		+BookDao(map:HashMap< String, Book>)	매개변수 생성자 다른 map을 전달 받아 bookMap의 초기값으로 사용함
		+getLastBookNo() : int	도서의 마지막 도서 번호 리턴 Iterator, keySet() 이용하여
		+addBook(book:Book) : void	전달받은 도서 객체 bookMap 에 추가
		+deleteBook(key:String) : Book	keySet()을 이용하여 전달받은 도서 번호와 일치하는 도서 bookMap에서 삭제, 삭제한 객 체 리턴
		+searchBook(title:String) : String	entrySet()을 이용하여 전달받은 도서 제목을 포함하는 bookMap의 key값 리턴
		+selectBook(key:String) : Book	전달받은 key값을 이용하여 bookMap의 해당 key값의 value인 Book 객체 리턴
		+selectAll() : HashMap<String, Book>	전체 도서 리턴
		+sortedBookList() : ArrayList<Book>	entrySet()을 이용하여 ArrayList<Book>을 가져오고 Collections.sort()를 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
                                   => Book 리턴 받아 null이 아닐 경우 "성공적으로 삭제"
                                   null일 경우 "삭제할 글이 존재하지 않음"
        4. 도서 검색출력          // searchBook (inputBookTitle()이 리턴한 도서 제목) 실행
                                   => key 리턴 받아 null일 경우 "조회한 글이 존재하지 않음"
                                   null이 아닐 경우 selectBook(key) 출력
        5. 전체 출력              // selectAll() 실행
                                   => 결과 map 리턴 받아 비어있을 경우 "없습니다."
                                   아닐 경우 Iterator, keySet()을 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 String 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
        // 첫 글일 경우 0으로 받아오게끔 BookDao의 getLastBookNo() 내용 구현 할 것!
        // setter를 이용하여 도서 번호를 마지막 도서 번호 +1 처리
        // BookDao의 addBook()메소드에 해당 Book 객체 전달

    }

    public Book deleteBook(String key){

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

    }

    public String searchBook(String title){

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

    }

    public Book selectBook(String key){

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

    }

    public HashMap<String, Book> selectAll(){

        // BookDao의 selectAll() 메소드를 통해 도서 전체를 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

    }

}
```