[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

BookMenu
~ sc : Scanner // 생성 - bm : BookManager // 생성
+ BookMenu() + mainMenu() : void + insertBook() : void // 도서 추가 + deleteBook() : void // 도서 삭제 + searchBook() : void // 도서 검색 + selectList() : void // 전체 조회

Run
+ main(args:String[]) : void

BookManager
- bookList : ArrayList<Board> // 생성
+ BookManager() + insertBook(book:Book) : void + deleteBook(bNo:int) : int + searchBook(title:String) : ArrayList<Book> + selectList() : ArrayList<Book>

3. 구현 클래스 설명

Package명	Class명	Method	설명
run	Run	+main(args:String[]) : void	BookMenu 클래스 객체 생성하고 mainMenu() 메소드 실행
model.vo	Book	+ Book()	기본생성자
		+ Book(title:String, category:int, author:String)	매개변수 3개짜리 생성자 (도서번호는 매개변수로 받지 않아요)
		+ toString() : String	필드 값 문자열 합친 후 리턴 category 분류 별로 출력 1 : 인문 / 2 : 자연과학 3 : 의료 / 4 : 기타
view	BookMenu	+ BookMenu()	기본 생성자
		+ mainMenu() : void	도서관리 프로그램에 해당하는 메인 메뉴 출력, 각 메뉴에 해당하는 메소드 실행 → 반복 출력되게 함
		+ insertBook() : void	제목, 카테고리, 저자명을 키보드로 입력 받고 입력 받은 값을 가지고 Book객체 생성 생성한 Book 객체를 BookManager의 insertBook 메소드로 전달
		+ deleteBook() : void	삭제할 도서번호를 키보드로 입력 받아 BookManager의 deleteBook 메소드로 전달 → 리턴 받은 결과 값을 가지고 성공, 실패 여부 출력
		+ searchBook() : void	검색할 도서명을 키보드로 입력 받아 BookManager의 searchBook 메소드로 전달 리턴 받은 리스트가 비어있는 경우 "검색 결과 없음" 출력 리턴 받은 리스트가 비어있지 않을 경우 검색 결과 목록 출력
		+ selectList() : void	BookMangaer의 selectList 메소드 호출하여 리턴 받은 리스타가 비어있는 경우 "도서가 없습니다." 출력 비어있지 않으 경우 전체 리스트 목록 출력

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

Package명	Class명	Method	설명
controller	BookManager	+ BookManager()	기본 생성자
		+ insertBook(book:Book) : void	전달받은 Book객체의 도서번호를 setter메소드를 통해 설정 한 후에 bookList에 추가
		+ deleteBook(bNo:int) : int	bookList 의 도서들 중 전달 받은 bNo값이 일치하는 도서가 있을 경우 도서 삭제 → 성공하면 1 리턴 → 실패하면 0 리턴
		+ searchBook(title:String) : ArrayList<Book>	bookList의 도서들 중 전달 받은 title값을 포함(contains)한 도서들 searchList에 추가 → searchList 리턴
		+ selectList() : ArrayList<Book>	→ bookList 리턴

4. class 구조

```
public class BookMenu{
    // Scanner 객체 생성
    // BookManager 객체 생성 (bm)

  public void mainMenu() {
      *** 도서 관리 프로그램 ***
      1. 새 도서 추가          → insertBook()
      2. 도서 삭제             → deleteBook()
      3. 도서 검색 출력        → searchBook()
      4. 전체 출력             → selectList()
      0. 끝내기                → "프로그램 종료" 출력 후 main()으로 리턴
      메뉴 번호 선택 :         >> 입력 받음
                               >> 메뉴 화면 반복 실행 처리
  }

  public void insertBook(){
      // "도서 제목 : "                                    >> 입력 받음 (title)
      // "도서 장르 (1:인문 / 2:자연과학 / 3:의료 / 4:기타) : "   >> 입력 받음 (category)
      // "도서 저자 : "                                    >> 입력 받음 (author)

      // 위에서 입력 받은 title, category, author를 매개변수로 한 Book 객체 생성 (book)
      // BookManager의 insertBook 메소드로 book 전달
  }

  public void deleteBook(){
      // "도서 번호 : "                                    >> 입력 받음 (bNo)
      // BookManager의 deleteBook 메소드로 bNo 전달
      // → 리턴 값 전달 받음 (result)

      // result가 1일 경우                          >> "성공적으로 삭제" 출력
      // result가 0일 경우                          >> "삭제할 글이 존재하지 않습니다."출력
  }
```

```
    public void searchBook() {
        // "도서 제목 : "                                    >> 입력 받음 (title)
        // BookManager의 searchBook 메소드로 title 전달
        // → 리턴 값 전달 받음 (searchList)

        // searchList가 비어 있을 경우                    >> "검색 결과가 존재하지 않습니다."출력
        // searchList가 비어있지 않을 경우                >> for문을 이용하여 searchList 출력
                                                          >> 또는 Iterator 이용하여 출력
    }
    public void selectList(){
        // BookManager의 selectList() 메소드 호출
        // → 리턴 값 전달 받음 (bookList)

        // bookList가 비어 있을 경우                      >> "도서 목록이 존재하지 않습니다."출력
        // bookList가 비어있지 않을 경우                  >> "for문을 이용하여 bookList 출력
                                                          >> 또는 for each 문 이용하여 출력
    }
}
```

```
public class BookManager{
    // ArrayList 객체 생성 (bookList)

    public void insertBook(Book book) {
        // 전달 받은 book은 현재 도서번호가 null인 채로 들어오는데
        // 새로운 도서가 추가될 때마다 추가되는 도서의 도서번호는
        // 리스트 마지막 도서 번호의 다음번호로 부여해야됨
        int lastNo = 0;    // 우선 변수 생성 및 초기화
        lastNo = bookList.get(bookList.size()-1).getbNo() + 1;  // 마지막 도서 번호 + 1
        // 하지만 리스트에 도서가 없는 경우, 즉 첫 도서 등록일 경우 위의 문장에서 예외 발생
        ➔ 어떤 예외처리가 발생하는지 알아보고 try catch문을 이용하여 오류 해결
        // 해당 예외 발생 시 lastNo = 1; 로 초기화
        // setter를 이용하여 book 도서 번호를 lastNo로 적용
        // bookList의 book 추가
    }

    public int deleteBook(int bNo){
        // for문을 이용하여 전달받은 도서번호가 존재하는 도서 삭제
        // 성공적으로 삭제 할 경우 1 리턴
        // 삭제 되지 않은 경우 즉, 존재하는 도서 번호가 없는 경우 0 리턴
    }

    public ArrayList<Book> searchBook(String title){
        ArrayList<Book> searchList = new ArrayList<Book>(); // 검색 결과값들을 보관할 리스트
        // for문을 이용하여 전달받은 제목을 포함한 도서를 searchList에 추가
        // searchList 리턴
    }

    public ArrayList<Book> selectList(){
        // bookList 리턴
    }
}
```