

해당 구현 클래스 다이어그램과 클래스 구조를 참고하여 프로젝트를 완성하시오

## 1. 구현 클래스 다이어그램 (Class Diagram)

<b>com.ohgiraffers.hw2.run.Application</b>
<u>+ main(args:String[]) : void</u>

<b>com.ohgiraffers.hw2.view.BoardMenu</b>
- bm : BoardManager - ib:InputBoard
+ mainMenu():void + sortSubMenu():void

<b>com.ohgiraffers.hw2.view.ResultPrinter</b>
+ printAllList(list:ArrayList<BoardDTO>):void + printBoard(b:BoardDTO):void + errorPage(msg:String):void + successPage(msg:String):void + noSearchResult():void

<b>com.ohgiraffers.hw2.view.InputBoard</b>
+ InputBoard():BoardDTO + InputBoardNo():int + InputBoardTitle():String + InputBoardContent():String

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

→ `AscBoardDate,AscBoardNo,AscBoardTitle,DescBoardDate,DescBoardNo,AscBoardTitle`

### 3. 구현 클래스 설명

Package명	Class명	Method	설명
<b>com.ohgiraffers.hw2.run</b>	<b>Application</b>	+ <u>main(args:String[]) : void</u>	BoardMenu 클래스 객체 생성하고 mainMenu() 메소드 실행
<b>com.ohgiraffers.hw2.view</b> .	<b>BoardMenu</b>	+ mainMenu():void	게시글 서비스 프로그램 메인 메뉴 출력, 각 메뉴에 해당하는 BoardManger클래스의 메소드 실행, 도서에 대한 정보는 InputBoard클래스의 메소드를 호출하여 사용 -> 반복 출력
		+ sortSubMenu()	게시글 정렬메뉴 (정렬은 오름차순,내림차순) 정렬대상: 글번호,날짜,제목

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

Package명	Class명	Method	설명
com.ohgiraffers.hw2.view	InputBoard	+ inputBoard:BoardDTO	게시글 등록: BoardDTO객체를 생성하여 글제목,작성자,를 입력받고, 글내용도 입력을 받는다. 글작성은 한번 입력 받을때마다 줄바꿈적용, "exit"키워드를 입력받으면 글작성을 종료한 다. 날짜는 현재날짜로 등록
		+ inputBoardNo():int	글번호를 입력받아 리턴
		+ inputBoardTitle():String	제목을 입력받아 리턴
		+ inputBoardContent():String	내용을 입력받아 리턴
com.ohgiraffers.hw2.view	ResultPrinter	+ printAllList (list:ArrayList<BoardDTO>):void	전달받은 객체안에 있는 값 출력 출력할내용이 없을 시에 "게시물이 없습니다."를 출력
		+ printBoard(b:BoardDTO):void	전달받은 내용 출력
		+ errorPage(msg:String):void	에러 메시지 출력 (클래스 예시를 참조)
		+ successPage(msg:String) :void	성공 메시지 출력 (클래스 예시를 참조)
		+ noSearchResult():void	검색결과없을시 출력 ("검색결과가 없습니다.")

Package명	Class명	Method	설명
com.ohgiraffers.hw2 .model.dto	BoardDTO	+BoardDTO()	기본 생성자
		+BoardDTO(board:int,boardTitle:String,boardWriter:String,boardDate:Date,boardContent:String,readCount:int)	매개변수있는 생성자
		+ setter()/ getter()	setter()/getter 메소드
		+toString() : String	필드 값 문자열 합친 후 리턴

Package명	Class명	Method	설명
com.ohgiraffers .hw2.controller	BoardManager	+ insertBoard(b:BoardDTO) :void	새 게시글 등록용 메소드 boardList객체에 추가
		+ selectAllList():void	게시글 전체 조회용 메소드 (boardList객체이용)
		+ selectOneBoard(boardNo:int):void	게시글1개 조회용 메소드 (boardList객체이용)
		+ updateBoardTitle (boardNo:int,title:String) :void	게시글 제목 변경용 메소드 (boardList객체이용)
		+ updateBoardContent (boardNo:int,content:String) :void	게시글 내용 변경용 메소드 (boardList객체이용)
		+ deleteBoard(boardNo:int):void	게시글 삭제용 메소드 (boardList객체이용)
		+ searchBoard(title:String):void	게시글 검색용 메소드 (boardList객체이용)
		+ sortList(c:Comparator<BoardDTO> :void	정렬 처리용 메소드 (boardList객체이용)
com.ohgiraffers .hw2.model .comparator	AscBoardDate	+ compare(b1:BoardDTO,b2:BoardDTO):int	등록일 오름차순
	AscBoardNo	+ compare(b1:BoardDTO,b2:BoardDTO):int	글번호 오름차순
	AscBoardTitle	+ compare(b1:BoardDTO,b2:BoardDTO):int	제목 오름차순
	DescBoardDate	+ compare(b1:BoardDTO,b2:BoardDTO):int	등록일 내림차순
	DescBoardNo	+ compare(b1:BoardDTO,b2:BoardDTO):int	글번호 내림차순
	DescBoardTitle	+ compare(b1:BoardDTO,b2:BoardDTO):int	제목 내림차순

#### 4. class 구조

```
public class BoardMenu {

    public void mainMenu() {
        *** 게시글 서비스 프로그램 ***

        1. 게시글 쓰기          // insertBoard(inputBoard())이 리턴한 객체) 실행
        2. 게시글 전체보기      // selectAllList() 실행
        3. 게시글 한 개보기     // selectOneBoard(inputBoardNo()) 실행
        4. 게시글 제목 수정     // updateBoardTitle(inputBoardNo(),inputBoardTitle()) 실행
        5. 게시글 내용 수정     // updateBoardContent(inputBoardNo(),inputBoardContent()) 실행
        6. 게시글 삭제         // deleteBoard(inputBoardNo()) 실행
        7. 게시글 제목 검색     // searchBoard(inputBoardTitle()) 실행
        8. 정렬하기           // sortSubMenu() 실행
        9. 끝내기             // "프로그램이 종료되었습니다." 출력

        메뉴 번호 선택 : >> 입력 받음

        // 메뉴 화면 반복 실행 처리

        // 해당 메뉴 번호에 따라 BoardManager 클래스 메소드 실행
    }

    public void sortSubMenu(){
        ***** 게시글 정렬 메뉴 *****

        1. 글번호순 오름차순 정렬 // sortList(AscBoardNo())
        2. 글번호순 내림차순 정렬 // sortList(DescBoardNo())
        3. 작성날짜순 오름차순 정렬 // sortList(AscBoardDate())
        4. 작성날짜순 내림차순 정렬// sortList(DescBoardDate())
        5. 글제목순 오름차순 정렬 // sortList(AscBoardTitle())
        6. 글제목순 내림차순 정렬 // sortList(DescBoardTitle())
        7. mainMenu()메소드로 이동

        메뉴번호를 입력받아 선택한 후에 출력
    }
}
```

```

public class InputBoard {
    public BoardDTO inputBoard(){
        // 게시글 등록
        // 글제목,작성자,글 내용(exit입력시 종료)는 각각 입력받기
        // 날짜는 따로 입력받지않고 현재날짜로 등록
    }
    public int inputBoardTitle(){
        제목을 입력받아서 리턴
    }
    public String inputBoardContent(){
        내용을 입력받아서 리턴
    }
}

```

```

public class ResultPrinter{
    public void printAllList(ArrayList<BoardDTO> list){
        // list에 출력할 내용이 있으면 출력
        // list에 출력할 내용이 없으면 "게시물이 없습니다" 출력
    }
}

```

```

public void printBoard(BoardDTO b){
    // 전달받은 내용 출력
}

```

```

public void errorPage(String msg){
    전달받은 문자열에 따라서 실패내용 출력
    insertBoard : 게시물 등록 실패
    selectOne: "조회된 글이 없습니다."
    updateTitle:"게시물 제목 수정 실패!"
    updateContent:"게시글 내용 수정 실패!"
    deleteBoard:"게시글 삭제 실패!"
}

```

```
public void successPage(String msg){
```

```
    전달받은 문자열에 따라서 실패내용 출력
```

```
    insertBoard : "게시글 등록 성공!" 출력
```

```
    updateTitle: "게시글 제목 수정 성공!" 출력
```

```
    updateContent:"게시글 내용 수정 성공!" 출력
```

```
    deleteBoard:"게시글 삭제 성공!" 출력
```

```
}
```

```
public void noSearchResult(){
```

```
    "검색 결과가 없습니다" 출력
```

```
}
```

```
}
```

```
public class BoardManager {  
    private ArrayList<BoardDTO> boardList;  
  
    /* 새 게시글 등록용 메소드 */  
    public void insertBoard(BoardDTO b){  
        // 글번호 0으로 초기화  
        // 조회내역이 있는 경우  
        -   조회된 내역이 있는 경우 새로운 리스트 생성  
        -   글 번호를 1로 변경  
        // 조회내역이 없는 경우  
        -   조회 내역이 있는 경우 기존 리스트의 마지막 인덱스의 글번호 조회  
  
        // 출력값 결정(successPage)  
    }  
    /* 게시글 전체 조회용 메소드 */  
    public void selectAllList(){  
        // 게시글 전체 조회 후 전체 출력 (printAllList() 이용)  
    }  
  
    /* 게시물 1개 조회용 메소드 */  
    public void selectOneBoard(int boardNo){  
        // 전체 게시물을 조회  
        // 조회에 성공하면 게시물 번호와 일치하는 게시물을 리스트에서 탐색  
        // 조회성공 시 조회수 1증가  
  
        // 일치하는 게시물이 없으면 에러출력,  
        // 일치하는 내용이 있으면 게시물 출력  
    }  
}
```



```

/* 게시물 제목 변경용 메소드 */
public void updateBoardTitle(int boardNo, String title) {
    // 전체 리스트 조회

    // 일치하는 게시물 조회
    조회 결과가 있는 경우 제목 수정 (
    // 결과 출력(successPage)
}

/* 게시물 내용 변경용 메소드 */
public void updateContent(int boardNo, String content) {
    // 전체 리스트 조회
    // 일치하는 게시물 조회
    조회 결과가 있는 경우 내용 수정 (
    수정 후 다시 파일에 저장
    // 결과 출력(successPage)

}

/* 게시물 삭제용 메소드 */
public void deleteBoard(int boardNo) {
    // 전체 게시물 조회
    // 게시물 번호와 일치하는 게시물 조회
    -> 일치하는 게시물이 있는 경우 삭제
    // 결과 출력(successPage())
}

```

```
/* 제목으로 조회 */
```

```
public void searchBoard(String title) {
```

```
    // 전체 게시물 조회
```

```
    // 검색 결과를 담은 리스트 생성
```

```
    // 제목을 포함하고 있는 게시물을 결과 리스트에 저장
```

```
    // 검색결과 리스트가 비어있는 경우 에러 출력, 비어 있지 않은 경우 결과 출력
```

```
}
```

```
/* 정렬 처리용 메소드 */
```

```
public void sortList(Comparator<BoardDTO> c) {
```

```
    // 리스트 전체 조회
```

```
    // 정렬
```

```
    // 출력
```

```
}
```

```
}
```