

[문제 1] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

MVC패턴을 이용한 ArrayList와 파일에 대한 Object 입/출력을 사용하는 문제이다. 해당 구현 클래스 다이어그램과 클래스 구조를 참고하여 프로젝트를 완성하시오

1. 구현 클래스 다이어그램 (Class Diagram)

Board
- boardNo : int // 게시글 번호 - boardTitle : String // 게시글 제목 - boardWrtier : String // 작성자 - boardDate : Date // 작성 날짜 - boardContent : String // 게시글 내용 - readCount : int // 조회수
+ Board() + Board(boardNo:int, boardTitle:String, boardWriter:String, boardDate:Date, boardContent:String) + setter() / getter() + toString() : String

BoardDao
- list : ArrayList<Board>
+ BoardDao() + getLastBoardNo() : int + writeBoard(board:Board) : void + displayAllList() : ArrayList<Board> + displayBoard() : Board + upReadCount(no:int) : void + modifyTitle(no:int, title:String) : void + modifyContent(no:int, content:String) : void + deleteBoard(no:int) : void + searchBoard(title:String) : ArrayList<Board> + saveListFile() : void

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

BoardMenu
- sc : Scanner - bm : BoardManager
+ BoardMenu() + mainMenu() : void + sortSubMenu() : void

BoardManager
- bd : BoardDao - sc : Scanner
+ BoardManager() + writeBoard() : void + displayAllList() : void + displayBoard() : void + modifyTitle() : void + modifyContent() : void + deleteBoard() : void + searchBoard() : void + saveListFile() : void + sortList(item:int, isDesc:Boolean) : void

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

* java.util.Comparator 인터페이스를 상속받은 정렬용 클래스들 작성함

→ DescBoardNo, AscBoardNo, DescBoardDate, AscBoardDate, DescBoardTitle, AscBoardTitle

3. 구현 클래스 설명

Package명	Class명	Method	설명
com.kh.collection.silsub1. run	Run	<u>+main(args:String[]) : void</u>	BoardMenu 클래스 객체 생성 하고 mainMenu() 메소드 실행
com.kh.collection.silsub1. view	BoardMenu	+BoardMenu()	기본 생성자
		+mainMenu() : void	게시글 프로그램에 해당하는 메 인 메뉴 출력, 각 메뉴에 해당 하는 BoardManager 클래스의 메소드 실행 → 반복 출력되게 함
		+sortSubMenu() : void	메인 메뉴의 9번을 눌렀을 때 출력되는 서브 메뉴 각 서브 메뉴에 해당하는 BoardManager 클래스의 메소 드 실행 → 반복 출력되게 함

Package명	Class명	Method	설명
com.kh.collection.silsub1. controller	BoardManager	+BoardManager()	기본 생성자
		+writeBoard() : void	키보드 입력을 통해 새로운 게 시글을 입력 받음 BoardDao의 writeBoard() 메소 드로 해당 게시글 전달
		+displayAllList() : void	BoardDao의 displayAllList() 메 소드를 통해 list를 전달받고 Iterator를 이용하여 전체 리스 트 출력
		+displayBoard() : void	키보드를 통해 조회할 글 번호 를 입력 받고 BoardDao의 displayBoard() 메소드로 전달 결과 값으로 해당 글 번호의 게 시글을 전달받고 출력 조회수를 올리기위해 BoardDao 의 upReadCount() 메소드를 호 출

		+modifyTitle() : void	키보드를 통해 수정할 글 번호와 수정할 제목을 입력 받고 BoardDao의 modifyTitle() 메소드로 해당 글 번호와 수정할 제목 값을 전달
		+modifyContent() : void	키보드를 통해 수정할 글 번호와 수정할 내용을 입력 받고 BoardDao의 modifyContent() 메소드로 글 번호와 수정할 내용 값을 전달
		+deleteBoard() : void	키보드를 통해 삭제할 글 번호를 입력 받고 BoardDao의 deleteBoard() 메소드로 전달
		+searchBoard() : void	키보드를 통해 검색할 제목을 입력 받고 BoardDao의 searchBoard() 메소드로 전달, 결과 값을 list로 받고 출력
		+saveListFile() : void	BoardDao의 saveListFile() 메소드 호출
		+sortList(item:int, isDesc:boolean) : void	BoardDao의 displayAllList() 메소드를 통해 전체 리스트를 전달 받고 이 전체 리스트를 전달 받은 item과 isDesc를 가지고 6개의 정렬 방법 구현

- * **modifyTitle(), modifyContent(), deleteBoard() 메소드에서 키보드로 입력 받은 수정 또는 삭제할 글 번호를 BoardDao의 displayBoard() 메소드로 전달하여 결과 값으로 해당 글 번호의 게시글을 전달 받고 출력하는 부분 추가**

Package명	Class명	Method	설명
com.kh.collection.silsub1. model.vo	Board	+Board()	기본 생성자
		+toString() : String	필드 값 문자열 합친 후 리턴
com.kh.collection.silsub1. model.comparator	DescBoardNo	+compare() : int	글 번호순 내림차순 정렬 처리
	AscBoardNo		글 번호순 오름차순 정렬 처리
	DescBoardDate		작성날짜순 내림차순 정렬 처리
	AscBoardDate		작성날짜순 오름차순 정렬 처리
	DescBoardTitle		글 제목순 내림차순 정렬 처리
	AscBoardTitle		글 제목순 오름차순 정렬 처리
com.kh.collection.silsub1. model.dao	BoardDao	+BoardDao()	기본 생성자 board_list.dat 파일의 내용을 읽어서 list에 저장함 null 될 때까지 저장함
		+getLastBoardNo() : int	게시글의 마지막 번호 리턴
		+writeBoard(board:Board) : void	전달받은 게시글을 list에 추가
		+displayAllList() : ArrayList<Board>	게시글 전체 list를 리턴
		+displayBoard(no:int) : Board	전달받은 글 번호와 일치하는 게시글 리턴
		+upReadCount(no:int) : void	전달받은 글 번호와 일치하는 게시글의 조회수 1 올려주기
		+modifyTitle(no:int, title:String) : void	전달받은 글 번호와 일치하는 게시글의 제목을 수정하기
		+modifyContent(no:int, content:String) : void	전달받은 글 번호와 일치하는 게시글 내용 수정
		+deleteBoard(no:int) : void	전달받은 글 번호와 일치하는 게시글 삭제
		+searchBoard(title:String) : ArrayList<Board>	전달받은 제목이 포함되어있는 게시글들 list 리턴
		+saveListFile() : void	board_list.dat 파일 저장하기

4. class 구조

```
public class BoardMenu{
    public void mainMenu {
        ***** 게시글 서비스 프로그램 *****

        1. 게시글 쓰기          // writeBoard() 실행
        2. 게시글 전체 보기      // displayAllList()
        3. 게시글 한 개 보기      // displayBoard()
        4. 게시글 제목 수정       // modifyTitle()
        5. 게시글 내용 수정       // modifyContent()
        6. 게시글 삭제           // deleteBoard()
        7. 게시글 검색           // searchBoard()
        8. 파일에 저장하기        // saveListFile()
        9. 정렬하기              // BookMenu 의 sortSubMenu() 실행
        10. 끝내기               // main() 으로 리턴함

        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 오름차순정렬    // BookManager 의 sortList(1, false) 실행
        2. 글번호순 내림차순정렬    // sortList(1, true) 실행
        3. 작성날짜순 오름차순정렬   // sortList(2, false) 실행
        4. 작성날짜순 내림차순정렬   // sortList(2, true) 실행
        5. 글제목순 오름차순정렬     // sortList(3, false) 실행
        6. 글제목순 내림차순정렬     // sortList(3, true) 실행
        7. 이전 메뉴로 이동         // return 처리

        메뉴 번호 선택 : 입력 받음

        // 메뉴 화면 반복 실행 처리
    }
}
```

```

public class BoardManager{
    public void writeBoard() {
        // "새 게시글 쓰기 입니다." 출력
        // "글 제목 : " >> 입력 받음 (공백 포함)
        // "작성자 : " >> 입력 받음 (공백 없이)
        // 작성날짜는 현재 날짜로 처리함
        // "글 내용 : " >> 여러 줄로 입력 받음 >>"exit" 입력하면 입력종료
        // BoardDao의 getLastBoardNo() 메소드를 통해 게시글의 마지막 번호를 알아옴
        // Board 객체 생성시 초기값으로 사용함
        // BoardDao의 writeBoard() 메소드 호출 시 해당 Board 전달
        // 파일이 존재하지 않을 경우, 즉 첫 글 등록일 경우 예외 발생
        ➔ 어떤 예외처리가 발생하는지 알아보고 try catch문을 이용하여 오류 해결
    }

    public void displayAllList(){
        // BoardDao의 displayAllList() 메소드를 호출하여 ArrayList를 전달받음
        // Iterator를 이용하여 list 에 기록된 정보를 모두 화면에 출력시킴
    }

    public void displayBoard(){
        // "조회할 글 번호 : " >> 입력 받음
        // BoardDao의 displayBoard() 메소드에 글 번호를 전달하여
        => 해당 글 번호의 게시글 전달 받아 출력하고
        // 조회수를 올리기 위해 BoardDao의 upReadCount() 메소드에 해당 글 번호 전달
        // 만약 없을 경우 "조회된 글이 없습니다." 출력
    }

    public void modifyTitle(){
        // "수정할 글 번호 : " >> 입력 받음
        // BoardDao의 displayBoard() 메소드에 글 번호를 전달하여
        => 해당 글 번호의 게시글 전달 받아 출력하고
        // "변경할 글 제목 : " >> 입력 받음
        // BoardDao의 modifyTitle() 메소드에 해당 글 번호와 변경할 제목 전달

        // 만약 없을 경우 "조회된 글이 없습니다." 출력
    }
}

```

```

public void modifyContent(){
    // 위의 modifyTitle() 메소드 구현 내용과 동일 >> title을 content로만 변경
}

public void deleteBoard() {
    // "삭제할 글 번호 : " >> 입력 받음
    // BoardDao의 displayBoard() 메소드를 호출하여 해당 글 번호의 게시글 전달 받아 출력
    // 만약 없을 경우 "조회된 글이 없습니다." 출력
    // 있을 경우 "정말로 삭제하시겠습니까? (y/n) : " >> 입력 받음
    // 대소문자 관계없이 Y 일 경우 BoardDao의 deleteBoard() 메소드에 해당 글 번호 전달
}

public void searchBoard(){
    // "검색할 제목 : " >> 입력 받음
    // BoardDao의 searchBooard() 메소드에 입력 값 전달
    // 제목은 중복될 수 있기 때문에 결과값 다중 → list로 결과값 전달받음
    // 결과값이 없으면 "검색 결과가 없습니다." 출력
    // 결과값이 있으면 list출력
}

public void saveListFile(){
    // BoardDao의 saveListFile()의 메소드 호출
}

public void sortList(int item, boolean isDesc){
    // BoardDao의 displayAllList()의 메소드를 호출하여 전체 글 리스트 전달 받아
    // item 이 1이면 번호순 정렬
    // item 이 2이면 날짜순 정렬
    // item 이 3이면 제목순 정렬 작동되게 정렬용 클래스 사용함
    // isDesc 가 true 이면 내림차순정렬 작동
    // isDesc 가 false 이면 오름차순정렬 작동되게 함
}
}

```