

[LAB2]

실습1

1. <https://github.com/KIMHYUNUK98/StudentApp.git>
2. <https://github.com/KIMHYUNUK98/FruitApp.git>
3. <https://github.com/KIMHYUNUK98/ToDoListApp.git>

실습2

1. ToDoListApp 실행

1) Add 명령

: Title 과 description을 추가할 수 있다.

```
1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
add

===== Create item Section
enter the title

test1
enter the description
test1_desc

1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
ls
Item Title: test1  Item Description:  test1_desc
```

2) Del 명령어

: 현재 저장되어 있는 Title 중 test1과 일치하는 파일을 지우는 명령어

```
1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
del
test1

===== Delete Item Section
enter the title of item to remove

1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
ls
Item Title: test2 Item Description: test2_desc
```

3) Edit 명령어

: 현재 저장된 파일의 이름을 변경할 수 있게 해주는 명령어

```
1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
edit

===== Edit Item Section
enter the title of the item you want to update

test2
enter the new title of the item
test2_update
enter the new description
test2_update_desc
item updated

1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
ls
Item Title: test2_update Item Description: test2_update_desc
```

4) Ls_name_asc 명령어

: 이름의 알파벳 순서대로 저장된 파일을 정렬해주는 명령어

```
1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
ls_name_asc
|
inside list_All method

test1test1_desc
test2_updatetest2_update_desc
test32test32_desc
test7test7_desc
```

5) Ls_name_desc 명령어

: 이름의 알파벳 순서의 반대로 저장된 파일을 정렬해주는 명령어

```
1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
ls_name_desc
|
inside list_All method

test7test7_desc
test32test32_desc
test2_updatetest2_update_desc
test1test1_desc
```

6) Ls_date 명령어

: 파일을 작성한 날짜 순서대로 정렬해주는 명령어

```
1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
ls_date
|
inside list_All method

test2_updatetest2_update_desc
test1test1_desc
test32test32_desc
test7test7_desc
```

7) Exit 명령어

: 실행되고 있는 프로그램을 종료하는 명령어

```
1. Add a new item ( add )
2. Delete an existing item ( del )
3. Update an item ( edit )
4. List all items ( ls )
5. sort the list by name ( ls_name_asc )
6. sort the list by name ( ls_name_desc )
7. sort the list by date ( ls_date )
8. exit (Or press escape key to exit)
Enter your choice >
exit
|
```

2. TodoListApp 프로젝트에서 사용하는 Java class 조사하기

1) 각 클래스 용도

Scanner : 기본적인 데이터 타입을 Scanner 클래스의 메소드를 이용해 입력받을 수 있다.

Comparator : Collection.sort 메소드를 사용해서 저장된 인자들을 정렬시킬 수 있다.

ArrayList : 자료를 순차적으로 저장하여 관리하는 프로그램을 구현할 때 사용되는 클래스

2) 자주 사용하는 메소드

addItem() : 파일을 추가하는 메소드

deleteItem() : 사용자 지정 파일을 삭제하는 메소드

editItem() : 사용자가 지정한 파일의 이름을 업데이트 하는 메소드

sortByName() : 이름을 순서대로 정렬하는 메소드

listAll() : 저장된 모든 파일을 보여주는 메소드

sortByDate() : 저장한 date 순서대로 파일을 정렬하는 메소드

3) 간단한 예제 찾아서 테스트

Scanner 클래스를 통해서 사용자가 입력하는 것들을 프로그램이 받아드릴 수 있다.

Comparator 객체를 생성하여 Collections 정렬을 사용하면 사용자가 원하는 ex) 문자열 길이로 정렬할 수 있다.

```
Collections.sort(strings, new Comparator<String>() {  
    @Override  
    public int compare(String s1, String s2) {  
        return s1.length() - s2.length();  
    }  
});
```

ArrayList<데이터 타입> 변수 를 통해서 공간을 할당한다.

```
public class Main {  
    public static void main(String[] args) {  
        ArrayList<String> arrayList = new ArrayList<String>();  
    }  
}
```

3. TodoListApp class 분석

1) 각 클래스의 역할과 실행되는 메뉴와의 관계 설명

TodoMain 클래스는 Main함수에서 호출하여 시작되는 클래스.

TodoMain 클래스에서 TodoList 클래스와 Menu 클래스가 선언되며

Menu 클래스를 통해서 사용자에게 메뉴를 보여준다.

사용자가 원하는 단어를 입력하면 TodoUtil 클래스가 실행된다.

TodoUtil 클래스가 실행되면 추가/제거/업데이트 함수에 따라서 TodoItem 클래스가 선언된다.

TodoItem 클래스에서는 get과 set 메소드를 통해서 파일의 저장, 제거가 진행된다.

다시 TodoMain클래스의 do while 반복문으로 돌아와서 해당 과정을 반복한다.

사용자가 quit을 입력하면 TodoMain 클래스가 종료되면서 전체 프로그램이 종료된다.

2) 멤버 변수: 역할 설명

Islist 변수 : 사용자가 ls 명령을 할 경우 true를 할당하고 if(islist)인 경우 listAll을 한다.

Do while 함수 실행하마다 false로 초기화 시켜야 한다.

Choice 변수 : 사용자가 메뉴를 보고 입력하는 값을 저장하는 String 값

l 변수 : TodoList 클래스를 생성하고 이를 연결해주는 변수

sc 변수 : Scanner type의 변수로 사용자가 입력하는 데이터를 받을 수 있게 해주는 변수

title 변수 : ADD할 경우 추가할 때 title 이름을 저장하는 변수

desc 변수 : title의 description 이름을 저장하는 변수

current_date 변수 : 입력시 현재의 date를 저장하게 되는 변수

list 변수 : TodoItem의 타입을 갖는 List 변수

index 변수 : update 할 경우 todoItem의 이름을 가지는 변수의 index값을 저장하는 변수

3) 멤버 메소드: 함수명, 파라미터, 리턴값, 동작기능 설명

Todoltem : title 과 desc 문자열을 입력받는 함수, (String title, String desc), return 없음

getTitle : 파라미터 X, title 값을 리턴받음, Title 이름을 가져오는 함수

setTitle : 파라미터 String title, return void, Title 이름을 저장하는 함수

getDesc : 파라미터 X, desc 값을 리턴받음, Desc 이름을 가져오는 함수

setDesc : 파라미터 String desc, return void, Desc 이름을 저장하는 함수

getCurrent_date : 파라미터 X, current_date 값을 리턴 받음, current_date를 가져오는 함수

setCurrent_date : 파라미터 Date current_date, return void, current_date를 저장하는 함수

addItem : 파라미터 todoltem t, return void, Todoltem의 값을 저장하는 함수

deleteItem : 파라미터 Todoltem t, return void, Todoltem의 값을 삭제하는 함수

editItem : 파라미터 Todoltem t, Todoltem update, return void, t의 index를 찾아서 삭제하고 update 된 Todoltem을 새로 추가하는 함수

getList : 파라미터 X, Todoltem의 ArrayList를 return 받는다. 모든 list를 가져오는 함수

sortByName : 파라미터 X, return void, TodoSortByName메소드를 선언해서 정렬하는 함수

listAll : 파라미터 X, return void, list에 있는 모든 내용을 for문을 반복시켜 출력시키는 함수

reverseList : 파라미터 X, return void, list에 파라미터로 Collection 메소드로 넘겨 정렬시키는 함수

sortByDate : 파라미터 X, return void, Todolist의 o1과 o2를 비교해서 정렬시키는 함수

indexOf : 파라미터 X, TodoList t의 Index를 return 한다. title이름의 index를 찾아주는 함수

isDuplicate : 파라미터 String title, return true or false, 해당 title 이름과 저장된 모든 list의 이름 중 같은 것이 있다면 true를 반환 아니면 false를 반환한다.

Compare : 파라미터 TodoItem o1, o2, return negative , 0 , 1, o1의 date와 o2의 date를 비교해서 값을 리턴한다.

Compare(Name) : 파라미터 TodoItem o1, o2, return negative, 0, 1, o1의 title과 o2의 title을 비교해서 값을 리턴한다.

createItem : 파라미터 TodoList item, return void, 사용자가 입력한 title과 desc가 기존의 저장된 파일의 이름과 중복되지 않으면 저장하는 함수

deleteItem : 파라미터 TodoList l, return void, 사용자가 title을 입력하면 item변수를 통해 List에 있는 모든 변수와 title의 이름을 비교하고 같다면 해당 List의 Item을 제거한다.

updateItem : 파라미터 TodoList l. return void. 사용자가 title을 입력하면 저장된 list의 내용과 title의 이름이 겹치는지 확인하고 없으면 new_title이름을 입력받고, new_description의 이름을 입력받는다. 그리고 title의 이름과 List 의 이름이 동일하다면 해당 title을 제거하고 new_title과 new_description TodoItem 클래스를 새로 생성하고 AddItem을 통해 새로 추가한다.

listAll : 파라미터 TodoList l, return void, 모든 list를 가져오고 해당 title과 desc를 출력하는 함수