

[LAB3]

<https://github.com/KIMHYUNUK98/ToDoListApp.git>

STEP1

1. 화면 표시 메시지 수정

1. ADD 명령

```
<===== MENU =====>
- add (ADD New Data)
- del (Delete exist data)
- edit (Update exist data)
- ls (List all data)
- ls_name_asc (List Name in Ascending)
- ls_name_desc (List Name in Descending)
- ls_date (List by Date)
- exit (Exit the Program)
```

Command > add

```
TITLE: C_Programming
CONTENT: 김인중 교수님 / 5교시 수업
Add Complete!!!
```

Command >

2. Ls 명령어

```
Command > ls
[Whole TITLE]
[ C_Programming ][ 김인중 교수님 / 5교시 수업 ] - Sat Sep 18 14:43:07 KST 2021
[ JAVA_Programming ][ 김기석 교수님 / 3교시 수업 ] - Sat Sep 18 14:44:05 KST 2021
[ Discrete ][ 홍신 교수님 / 6교시 수업 ] - Sat Sep 18 14:44:31 KST 2021
[ Linear_Algebra ][ 김성욱 교수님 / 2교시 ] - Sat Sep 18 14:45:08 KST 2021
```

Command >

3. Del 명령어

```
Command > del

===== DELETE MENU =====

TITLE: Discrete
Delete Complete!!!
```

```
Command > ls
[Whole TITLE]
[ C_Programming ][ 김인중 교수님 / 5교시 수업 ] - Sat Sep 18 14:43:07 KST 2021
[ JAVA_Programming ][ 김기석 교수님 / 3교시 수업 ] - Sat Sep 18 14:44:05 KST 2021
[ Linear_Algebra ][ 김성욱 교수님 / 2교시 ] - Sat Sep 18 14:45:08 KST 2021
```

Command >

4. Edit 명령어

```
Command > ls
[Whole TITLE]
[ C_Programming ][ 김인중 교수님 / 5교시 수업 ] - Sat Sep 18 14:43:07 KST 2021
[ JAVA_Programming ][ 김기석 교수님 / 3교시 수업 ] - Sat Sep 18 14:44:05 KST 2021
[ Linear_Algebra ][ 김성욱 교수님 / 2교시 ] - Sat Sep 18 14:45:08 KST 2021

Command > edit

===== DELETE MENU =====

DELETE TITLE: C_Programming
NEW TITLEPython_Programming
CONTENT: 이번에 수강하지 않습니다
item updated
Update Complete!!!

Command > ls
[Whole TITLE]
[ JAVA_Programming ][ 김기석 교수님 / 3교시 수업 ] - Sat Sep 18 14:44:05 KST 2021
[ Linear_Algebra ][ 김성욱 교수님 / 2교시 ] - Sat Sep 18 14:45:08 KST 2021
[ Python_Programming ][ 이번에 수강하지 않습니다 ] - Sat Sep 18 14:47:00 KST 2021

Command >
```

5. Ls_name_asc 명령어

```
Command > ls_name_asc
Ascending Sort Complete!!!

===== Show List =====
[ JAVA_Programming ][ 김기석 교수님 / 3교시 수업 ] - Sat Sep 18 14:44:05 KST 2021
[ Linear_Algebra ][ 김성욱 교수님 / 2교시 ] - Sat Sep 18 14:45:08 KST 2021
[ Python_Programming ][ 이번에 수강하지 않습니다 ] - Sat Sep 18 14:47:00 KST 2021

Command >
```

6. Ls_name_dsc 명령어

```
Command > ls_name_dsc
Descending Sort Complete!!!

===== Show List =====
[ Python_Programming ][ 이번에 수강하지 않습니다 ] - Sat Sep 18 14:47:00 KST 2021
[ Linear_Algebra ][ 김성욱 교수님 / 2교시 ] - Sat Sep 18 14:45:08 KST 2021
[ JAVA_Programming ][ 김기석 교수님 / 3교시 수업 ] - Sat Sep 18 14:44:05 KST 2021

Command >
```

7. Ls_date 명령어

```
Command > ls_date
Sort by Date order Complete!!!

===== Show List =====
[ JAVA_Programming ][ 김기석 교수님 / 3교시 수업 ] - Sat Sep 18 14:44:05 KST 2021
[ Linear_Algebra ][ 김성욱 교수님 / 2교시 ] - Sat Sep 18 14:45:08 KST 2021
[ Python_Programming ][ 이번에 수강하지 않습니다 ] - Sat Sep 18 14:47:00 KST 2021

Command >
```

2. 명령어 사용법(Help) 메뉴 별도 실행 기능 구현

```
Main.java  TodoMain.java  TodoUtil.java  Menu.java  TodoItem.java  TodoList.java
1 package com.todo.menu;
2 public class Menu {
3
4     public static void displaymenu()
5     {
6         System.out.println();
7         System.out.println("<===== MENU =====>");
8         System.out.println("- add (ADD New Data)");
9         System.out.println("- del (Delete exist data)");
10        System.out.println("- edit (Update exist data)");
11        System.out.println("- ls (List all data)");
12        System.out.println("- ls_name_asc (List Name in Ascending)");
13        System.out.println("- ls_name_desc (List Name in Descending)");
14        System.out.println("- ls_date (List by Date)");
15        System.out.println("- exit (Exit the Program)");
16    }
17    public static void prompt()
18    {
19        System.out.print("\n Command > ");
20    }
21 }
22
```

3.. 내용 입력 기능 개선

```
1 package com.todo.service;
2
3 import java.util.*;
4
5
6
7
8 public class TodoUtil {
9
10     public static void createItem(TodoList list) {
11
12         String title, desc;
13         Scanner sc = new Scanner(System.in);
14
15         System.out.print("\n" + "TITLE: ");
16
17         title = sc.next();
18         if (list.isDuplicate(title)) {
19             System.out.printf("title can't be duplicate");
20             return;
21         }
22
23         sc.nextLine();
24         System.out.print("CONTENT: ");
25         desc = sc.nextLine().trim();
26
27         TodoItem t = new TodoItem(title, desc);
28         list.addItem(t);
29     }
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48     public static void updateItem(TodoList l) {
49
50         Scanner sc = new Scanner(System.in);
51
52         System.out.println("\n===== DELETE MENU =====\n");
53         System.out.print("DELETE TITLE: ");
54         String title = sc.next().trim();
55         if (!l.isDuplicate(title)) {
56             System.out.println("title doesn't exist");
57             return;
58         }
59
60         System.out.print("NEW TITLE");
61         String new_title = sc.next().trim();
62         if (l.isDuplicate(new_title)) {
63             System.out.println("title can't be duplicate");
64             return;
65         }
66
67         sc.nextLine();
68         System.out.print("CONTENT: ");
69         String new_description = sc.nextLine().trim();
70         for (TodoItem item : l.getList()) {
71             if (item.getTitle().equals(title)) {
72                 l.deleteItem(item);
73                 TodoItem t = new TodoItem(new_title, new_description);
74                 l.addItem(t);
75                 System.out.println("item updated");
76             }
77         }
78     }
79 }
80
```

STEP2

1. 항목 내 날짜 정보를 문자열로 다루도록 수정

```
12 public TodoItem(String title, String desc){
13     this.title=title;
14     this.desc=desc;
15     SimpleDateFormat f = new SimpleDateFormat("yyyy/mm/dd kk:mm:ss");
16     this.current_date = f.format(new Date());
17 }

35 public String getCurrent_date() {
36     return current_date;
37 }
38
39 public void setCurrent_date(String current_date) {
40     this.current_date = current_date;
41 }
42
```

2. TodoItem.java 내에 toSaveString() 추가

```
47
48 public String toSaveString() {
49     return title + "##" + desc + "##" + current_date + "\n";
50 }
51 }
```

3. 데이터 파일 저장/읽기 기능 구현

- saveList 메소드

```
122 public static void saveList(TodoList l, String string) {
123     // TODO Auto-generated method stub
124     try {
125         Writer w = new FileWriter(string);
126
127         for(TodoItem item : l.getList()) {
128             w.write(item.toSaveString());
129         }
130         w.close();
131
132     } catch (FileNotFoundException e) {
133         e.printStackTrace();
134     } catch (IOException e) {
135         e.printStackTrace();
136     }
137 }
```


- loadList 메소드

```

94 public static void loadlist(TodoList l, String string) {
95     // TODO Auto-generated method stub
96     try {
97         BufferedReader br = new BufferedReader(new FileReader(string));
98
99         String oneline;
100         int line = 0;
101         while((oneline = br.readLine()) != null) {
102             StringTokenizer st = new StringTokenizer(oneline, "##");
103             String title = st.nextToken();
104             String desc = st.nextToken();
105             String date = st.nextToken();
106
107             TodoItem item = new TodoItem(title, desc);
108             l.addItem(item);
109             line++;
110         }
111
112         br.close();
113         System.out.println(line + " 개의 항목을 읽었습니다.\n");
114     } catch (FileNotFoundException e) {
115         System.out.println("todolist.txt 파일이 없습니다.\n");
116     } catch (IOException e) {
117         e.printStackTrace();
118     }
119 }
120 }

```

- todolist.txt 파일 내용

```

todolist - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
과제##실전 프로젝트 HW3 / 컴퓨터 구조 HW2 / C 프로그래밍 HW2##2021/36/18 16:36:43
과목##선형대수학 / 이산수학 / C 프로그래밍 / JAVA / 컴퓨터 구조 / 실전프로젝트##2021/36/18 16:36:43
스터디##컴퓨터 구조 / 이산 수학##2021/36/18 16:36:43
공동체##성가대 공동체##2021/36/18 16:36:43
관심분야##AI / 자율 주행 / 컴퓨터 비전..##2021/36/18 16:36:43
SW_대회##대경권 프로그래밍 대회 / 자율 주행 대회##2021/36/18 16:36:43
취약과목##컴퓨터 구조 / 이산수학##2021/36/18 16:36:43
강점과목##C Programming / 선형대수학##2021/37/18 16:37:04
보완언어##JAVA 프로그래밍 / 파이썬##2021/37/18 16:37:22
개인정보##김현욱 / 21800201 / 980220 / 전산전자공학부##2021/38/18 16:38:17

```

<실행하면서 느낀점>

자바를 배우지 않은 관점에서 LAB3 과제 설명을 듣고 숨이 막혔지만 기존에 배운 FILE I/O 이론을 응용시켜서 프로그래밍을 할 수 있는 실습이었습니다. C 프로그래밍을 알고 있으면 금방 이해할 수 있었고, C 와 비교를 해보자면 더 간편하고 쉽게 읽을 수 있는 것 같습니다. Class간 메소드 연동을 해서 사용하는 것이 C와 비교해서 정말 편리한 부분이었습니다. 언어를 아직 이론으로 배우지 않았지만 이렇게 실습하면서 언어를 배우는 새로운 과정이 언어를 금방 습득하는데 도움이 될 듯합니다.