

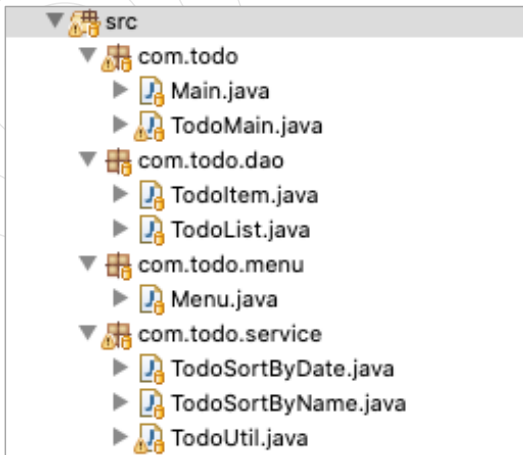
프로그래밍 스튜디오

동영상 및 강의자료 무단 복제 및 배포를 금지합니다.

14

TodoList App Update

Step1: TodoList CRUD project 소스 분석



- 패키지별 클래스
 - com.todo : main함수, 사용자 입력에 따른 처리 기능
 - com.todo.dao : 데이터 클래스 (TodoItem, TodoList)
 - com.todo.menu : 사용자에게 출력하는 메뉴
 - com.todo.service : 다양한 기능처리를 위해 필요한 클래스
- 기능
 - TodoItem 추가
 - TodoItem 삭제
 - TodoItem 수정
 - TodoItem 조회
 - TodoItem 이름 정렬
 - TodoItem 이름 역순정렬
 - TodoItem 날짜 정렬

Step1: TodoList CRUD project 소스 분석

1. TodoListApp 소스를 기반으로 자신의 IntelliJ에 포팅한 후, 실행해 볼 것.
2. TodoListApp 프로젝트에서 사용하는 Java class 확인하기
 1. 각 클래스 용도 (Scanner, ArrayList, Comparator)
 2. 자주 사용하는 메소드
 3. 간단한 예제 찾아서 테스트
3. TodoListApp의 각 클래스 분석
 1. 각 클래스의 역할과 실행되는 메뉴와의 관계
 2. 멤버변수 역할
 3. 멤버메소드 : 함수명, 파라미터, 리턴값, 동작기능

Step2: TodoListApp UI 개선

1. 화면 표시 메시지 수정 - **설명 동영상과 정확하게 일치시킬 필요는 없음! 적절하게 수정할 것!**
 - 표시되는 메시지를 일관성 있게 정리할 것! (영문 / 한글 사용 자유)
 - 각 명령어 실행 중에 친절한 메시지 표시하면 좋음
 - 관련 소스: TodoUtil.java, Menu.java, TodoMain.java
2. 명령어사용법(Help) 메뉴에 대한 기능 구현
 - 사용법을 매번 표현하지 말고 help 명령어 입력 시에 나타나도록 수정
 - Menu.displaymenu() 수정
 - Menu.prompt() 추가
 - TodoMain.java 수정
3. 내용 입력 기능 개선
 - 항목 추가할 때, 내용에 단어만 입력되는 오류 개선하여 문장이 입력되도록 개선할 것!
 - TodoUtil.java updateItem(), createItem() 수정

5

Step3: TodoListApp File 기능 구현

1. 항목 내 날짜 정보를 문자열로 다루도록 수정
 - 파일에 저장하고 불러오는 기능의 편의성을 위해 문자열로 보관하는 것이 좋음!
 - TodoItem.java 내 current_date 를 String 타입으로 변경 & 관련 메소드 수정
 - TodoItem 생성자에서 SimpleDateFormat을 사용하여 현재시간을 문자열로 저장할 것!
2. TodoItem.java 내에 toSaveString() 추가

```
public String toSaveString() {  
    return title + "##" + desc + "##" + current_date + "\n";  
}
```
3. 데이터파일(todolist.txt) 저장/읽기 기능 구현 - **프로그램 시작 시 읽기 & 종료 시 저장**
 - TodoUtil.java 내에 메소드 구현
 - **public static void** saveList(TodoList l, String filename)
 - FileWriter 사용
 - **public static void** loadList(TodoList l, String filename)
 - BufferedReader, FileReader, StringTokenizer 사용

6

데이터파일(todoList.txt) 샘플

Command > **ls**

[전체 목록]

[과제] 운영체제, 알고리즘 과제하기 - 2021/09/10 10:00:01

[회식] 동아리 동기들과 밥먹기 - 2021/09/12 12:10:21

[운동] 1킬로미터 빠르게 걷기 - 2021/09/13 20:30:31

[세미나] 랩실 세미나 준비하기 - 2021/09/14 09:20:45



TodoUtil.saveList(l, "todoList.txt");

```
1 과제##운영체제, 알고리즘 과제하기##2021/09/10 10:00:01
2 회식##동아리 동기들과 밥먹기##2021/09/12 12:10:21
3 운동##1킬로미터 빠르게 걷기##2021/09/13 20:30:31
4 세미나##랩실 세미나 준비하기##2021/09/14 09:20:45
```

5 |

todoList.txt

7

Step2 참고

- 내용 입력 기능 개선
 - 항목 추가할 때, 내용에 단어만 입력되는 오류 개선하여 문장이 입력되도록 개선
- TodoUtil.java
 - updateItem()
 - createItem()

```
public static void createItem(TodoList l) {
    String title, desc;
    Scanner sc = new Scanner(System.in);

    System.out.print("[항목 추가]\n"
        + "제목 > ");

    title = sc.next();
    if (l.isDuplicate(title)) {
        System.out.println("제목이 중복됩니다!");
        return;
    }
    sc.nextLine();
    System.out.print("내용 > ");
    desc = sc.nextLine().trim();

    TodoItem t = new TodoItem(title, desc);
    l.addItem(t);
    System.out.println("추가되었습니다.");
}
```

8

Step2 참고

- 항목 내 날짜 정보를 문자열로 다루도록 수정
 - 파일에 저장하고 불러오는 기능의 편의성을 위해 문자열로 보관하는 것이 좋음!
 - TodoItem.java
 - String current_date;
 - 생성자 변경 SimpleDateFormat 사용

```
public class TodoItem {
    private String title;
    private String desc;
    private String current_date;
```

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

Step2 참고

- Class SimpleDateFormat

Constructors

Constructor and Description

SimpleDateFormat()

Constructs a SimpleDateFormat using the default pattern and date format.

SimpleDateFormat(String pattern)

Constructs a SimpleDateFormat using the given pattern and the default locale.

SimpleDateFormat(String pattern, DateFormatSymbols formatSymbols)

Constructs a SimpleDateFormat using the given pattern and date format symbols.

SimpleDateFormat(String pattern, Locale locale)

Constructs a SimpleDateFormat using the given pattern and the default locale.

Letter	Date or Time Component
G	Era designator
y	Year
Y	Week year
M	Month in year
w	Week in year
W	Week in month
D	Day in year
d	Day in month
F	Day of week in month
E	Day name in week
u	Day number of week (1 = Monday, ..., 7 = Sunday)
a	Am/pm marker
H	Hour in day (0-23)
k	Hour in day (1-24)
K	Hour in am/pm (0-11)
h	Hour in am/pm (1-12)

<https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>

Step3 참고

- 데이터파일 (todolist.txt) 저장 기능 구현
 - 프로그램 종료 시 저장
 - TodoUtil.java
 - saveList()
 - FileWriter 사용

```
public static void saveList(TodoList l, String filename) {
    try {
        Writer w = new FileWriter(filename);
        for (TodoItem item : l.getList()) {
            w.write(item.toSaveString());
        }
        w.close();
        System.out.println("모든 데이터가 저장되었습니다.");
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

TodoItem.java

```
public String toSaveString() {
    return title + "##" + desc + "##" + current_date + "\n";
}
```

11

Step3 참고

- 데이터파일 (todolist.txt) 읽기 기능 구현
 - 프로그램 시작 시 읽기
 - TodoUtil.java
 - loadList()
 - BufferedReader, FileReader, StringTokenizer 사용

todolist.txt

```
1 과제##운영체제, 알고리즘 과제하기##2021/09/10 10:00:01
2 회식##동아리 동기들과 밥먹기##2021/09/12 12:10:21
3 운동##1킬로미터 빠르게 걷기##2021/09/13 20:30:31
4 세미나##랩실 세미나 준비하기##2021/09/14 09:20:45
5
```

```
public static void loadList(TodoList l, String filename) {
    try {
        BufferedReader br = new BufferedReader(new FileReader(filename));
        String line;
        int count = 0;
        while((line = br.readLine()) != null) {
            StringTokenizer st = new StringTokenizer(line, "##");
            String title = st.nextToken();
            String description = st.nextToken();
            String current_date = st.nextToken();
            TodoItem item = new TodoItem(title, description);
            item.setCurrent_date(current_date);
            l.addItem(item);
            count++;
        }
        br.close();
        System.out.println(count+"개의 항목을 읽었습니다.");
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        System.out.println(filename+" 파일이 없습니다.");
        //e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```


Step4: TodoListApp 기능 변경

1. 항목 목록에서 개수를 표현할 것!

- TodoUtil 클래스 내 listAll() 변경

2. TodoItem 필드 추가 및 기능에 반영

- 카테고리 필드 - String category; / 마감일자 필드 - String due_date;
- 추가, 삭제, 수정, 목록출력, 파일읽기 등의 기능에 변경 사항 반영

3. 기능 변경

- 목록 출력 시 일련번호를 함께 출력하도록 하여, 특정 항목을 선택할 때에 이 번호를 사용할 것.
- [항목 수정], [항목 삭제] 에서 제목으로 찾는 대신, 번호를 입력하도록 변경하여 기능 수행

```
Command > ls
```

```
[전체 목록, 총 4개]
```

1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
2. [기타] 동아리회식 - 동아리 동기들과 밥먹기 - 2021/09/29 - 2021/09/12 12:10:21
3. [취미] 운동 - 1킬로미터 빠르게 걷기 - 2021/09/15 - 2021/09/13 20:30:31
4. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45

13

```
Command > ls
```

```
[전체 목록, 총 4개]
```

1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
2. [기타] 동아리회식 - 동아리 동기들과 밥먹기 - 2021/09/29 - 2021/09/12 12:10:21
3. [취미] 운동 - 1킬로미터 빠르게 걷기 - 2021/09/15 - 2021/09/13 20:30:31
4. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45

```
Command > del
```

```
[항목 삭제]
```

```
삭제할 항목의 번호를 입력하시오 > 3
```

```
3. [취미] 운동 - 1킬로미터 빠르게 걷기 - 2021/09/15 - 2021/09/13 20:30:31
```

```
위 항목을 삭제하시겠습니까? (y/n) > y
```

```
삭제되었습니다.
```

```
Command > ls
```

```
[전체 목록, 총 3개]
```

1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
2. [기타] 동아리회식 - 동아리 동기들과 밥먹기 - 2021/09/29 - 2021/09/12 12:10:21
3. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45

```
Command > edit
```

```
[항목 수정]
```

```
수정할 항목의 번호를 입력하시오 > 3
```

```
3. [취미] 운동 - 1킬로미터 빠르게 걷기 - 2021/09/15 - 2021/09/13 20:30:31
```

```
새 제목 > 운동하기
```

```
새 카테고리 > 취미
```

```
새 내용 > 평봉필드 2바퀴 돌기
```

```
새 마감일자 > 2021/10/10
```

```
수정되었습니다.
```

```
Command > ls
```

```
[전체 목록, 총 4개]
```

1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
2. [기타] 동아리회식 - 동아리 동기들과 밥먹기 - 2021/09/29 - 2021/09/12 12:10:21
3. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45
4. [취미] 운동하기 - 평봉필드 2바퀴 돌기 - 2021/10/10 - 2021/09/22 21:09:00

Step4: TodoListApp 기능 변경

4. 검색 기능 추가

- 명령어 형식: **find <키워드>**
- 현재 목록 내의 제목과 내용에서 <키워드> 를 포함하고 있는 모든 항목을 출력한다.
- help 명령의 메뉴에 추가, 명령어에 대한 기능 추가

```
Command > ls
[전체 목록, 총 4개]
1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
2. [기타] 동아리회식 - 동아리 동기들과 밥먹기 - 2021/09/29 - 2021/09/12 12:10:21
3. [취미] 운동 - 1킬로미터 빠르게 걷기 - 2021/09/15 - 2021/09/13 20:30:31
4. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45

Command > find 3주
1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
총 1개의 항목을 찾았습니다.

Command > find 하기
1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
4. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45
총 2개의 항목을 찾았습니다.
```

15

Step5: TodoListApp 기능 변경 (선택과제)

5. 최신순 정렬 기능(선택기능)

- 명령어 형식, **ls_date_desc** (ls_date 의 역순으로 정렬)
- 현재 목록을 날짜 역순으로 출력한다.
- help 명령의 메뉴에 추가, 명령어에 대한 기능 추가

6. 카테고리 검색 기능(선택기능)

- 명령어 형식, **find_cate <키워드>**
- 현재 목록의 카테고리에 검색 <키워드>를 포함하고 있는 모든 항목을 출력한다.
- help 명령의 메뉴에 추가, 명령어에 대한 기능 추가

```
Command > find_cate 과제
1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
4. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45
총 2개의 항목을 찾았습니다.
```

16

Step5: TodoListApp 기능 변경 (선택과제)

7. 카테고리 목록 출력 기능(선택기능)

- 명령어 형식, `ls_cate`
- 현재 등록되어 있는 카테고리를 중복되지 않도록 출력한다.
- help 명령의 메뉴에 추가, 명령어에 대한 기능 추가

```
Command > ls
[전체 목록, 총 4개]
1. [과제] 3주차과제하기 - 운영체제, 알고리즘 과제하기 - 2021/09/21 - 2021/09/10 10:00:01
2. [기타] 동아리회식 - 동아리 동기들과 밥먹기 - 2021/09/29 - 2021/09/12 12:10:21
3. [취미] 운동 - 1킬로미터 빠르게 걷기 - 2021/09/15 - 2021/09/13 20:30:31
4. [과제] 세미나발표 - 랩실 세미나 준비하기 - 2021/09/27 - 2021/09/14 09:20:45

Command > ls_cate
과제 / 기타 / 취미
총 3개의 카테고리가 등록되어 있습니다.
```