2024 소프트웨어프로젝트 프로젝트 3

사진/사물 검색 프로그램 (Picture Search)

1. 내용

휴대폰의 사진 관리 프로그램을 보면, 사진 속의 사물을 지정해서 사진 검색을 하는 기능이 있다 (예, 구글 포토) 사진 속 사물을 인식하는 기능은 외부에서 따로 제공된다고 가정하고, 사진과 사물 정보를 연계해서 유지하면서 사진 검색기능을 제공하는 프로그램을 작성한다. 프로젝트 3에서는 사진 정보를 관리하면서 파일과 콘솔로 입출력하는 버전을 만들고, 향후 프로젝트에서 GUI 버전을 만들게 된다.
단순히 입출력으로 사진/사물 정보를 보여주는 프로그램이 아니고, 사진/사물 정보를 입력, **메모리 내 관리**, 출력하는 프로그램을 작성하는 것임. 향후 사진 정보의 추가/수정/삭제/검색 기능을 추가하는 것이 용이하게 **객체 지향 개념으로 작성**하여야 한다. 따라서 다음 요구 사항에 맞게 프로그램을 작성하여야 한다.

주의)
1. 이번 학기를 좌우하는 과제. 이 과제를 대충해서 넘어가면, 이어지는 내용은 제대로 소화할 수 없음. 결국, 시간과 노력만 낭비하고 Java는 어렵다는 인식만 남게 됨. **제대로 하는 것이 매우 중요.**
2. 중간시험과도 직결되는 내용. (과제를 스스로 했는지 재확인하는 문제 출제)
3. 문제가 복잡하기 때문에, 설명도 복잡할 수밖에 없음. 복잡한 실제 생활 문제를 해결하는 것이 컴퓨터공학의 본질. 문제 설명이 복잡하다고 포기하고 거부하면, 팀장이 짜라는 대로 짜는 저급 프로그래머가 될 수밖에 없다. **이 설명서를 읽고 또 읽어서 스스로 문제를 이해하여야 한다.**
4. 제대로 프로그램을 작성하려면, 고려해야 할 사항이 많기 때문에, 일찍 시작하는 것이 중요.
5. Eclipse에서 package는 따로 지정하지 않는다. (default package). 뒤에 나오는 autotest 관련 사항

1) Picture class
- 사진과 그 속의 사물 정보, 또 기타 사진 관련 정보 (예, 태그)를 하나의 객체로 모아서 나타냄.
- 즉, Picture App에서 저장하는 사진/사물/정보 하나를 담당.
- 다음 정보들을 포함하여야 한다.
  ID, timestamp, 사진 이미지 정보, [사물 정보], [태그], [주석]. ([] 항목은 생략 가능)
- 정보들을 다루는 각종 메소드들 포함 (향후, 계속 추가되어야 함)
- 하나의 사진에 대한 정보를 콘솔로 출력하여 주는 public void print() 메소드는 **필수**
- 생성자의 인자로 image 파일 이름을 주면, 현재 시각으로 timestamp 및 ID를 만드는 생성자,
  Picture(String imageFileName) **필수**

2) PictureList class
- 사진/사물 객체들의 묶음을 나타냄.
- 즉, Pictrure 인스턴스들을 모아서 관리하는 클래스.
- Excel로 사진/사물 관리를 한다고 가정한다면, Picture는 각 행, PictureList는 테이블 전체.
- 자료구조는 array 사용 (향후, ArrayList로 갱신)
- 리스트에 있는 전체 사진 숫자를 알려주는 public int size() 메소드는 **필수**
- 리스트의 i 번째 사진 정보를 알려주는 public Picture get(int i) 메소드는 **필수**
- 리스트의 끝에 사진 정보를 추가하는 public void add(Picture pic) 메소드는 **필수**
- 인자로 파일 이름을 주면 해당 파일에 기술된 사진/사진 정보들로 리스트를 만드는 생성자,
  PictureList(String infoFileName) **필수**
- 모든 사진/사물 정보들을 timestamp 시간 순으로 정렬하는 public void sortByDate() **필수**

- 파일에서 사진/사물 정보 내용을 한 줄씩 읽고 파싱을 해서, Picture와 PictureList 객체를 만들고,

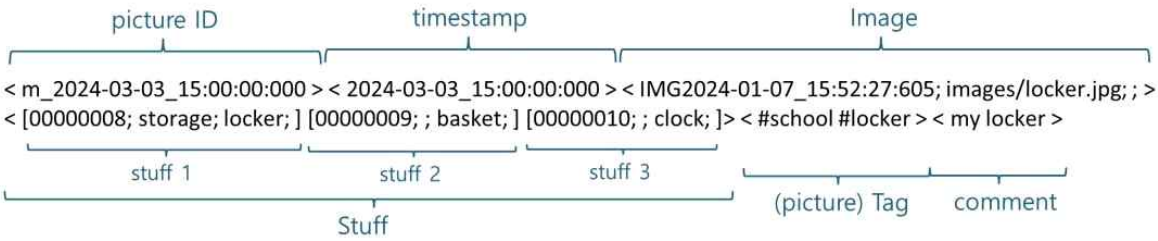
- 반대로, PictureList 객체를 파일에 저장하는 메소드 구현 추천
- 검색 조건에 따라서 Picture 리스트를 만드는 메소드들은 향후, ArrayList를 배우고 난 뒤에, 추가
- 그 외의 내부 구현 사항은 보여서는 안 됨.
- main()을 포함하면 안 됨.

3) Stuff class
- 사진 속 등장하는 사물의 정보들을 하나의 객체들로 묶어서 나타냄.
- (ID, 종류, [이름], [태그]) 정보들을 String 형으로 나타낸다. (여기의 태그는 사물에 대한 태그)
- 예, (00000008, storage, locker, #school)

4) StuffList class
- 사물 객체들의 리스트
- 즉, PictrureList 등장하는 모든 사물들의 정보를 보관한다.
- 하나의 사물이 여러 사진들에 나오는 경우, 그들은 **반드시 같은 사물 정보로 취급**되어야 한다.
- 따라서 프로그램에서는 하나의 StuffList가 공유된다. (즉, static을 사용하여 class 변수/메소드로)
- 같은 사물이면 해당 ID를, 다른 사물이 처음 등장하면 새로운 ID를 할당하는 메소드 **필수**
- 현재 등록된 모든 사물 정보들을 콘솔로 출력하여 주는 public static void print() 메소드는 **필수**

5) Tag, Image도 class화 하는 것 추천

6) 사진/사물, 즉, Picture 정보를 기술하는 파일은 다음 양식을 따른다.
- 하나의 줄(line)이 하나의 Picture를 기술.
- “<”과 “>”로 둘러싸인 최소 3개, 최대 6개 항목으로 구성
  * Picture ID (내부 식별자).
  * timestamp : 사진/사물 정보가 생성된 시각.
    시간 표시는 "yyyy-MM-dd_HH:mm:ss:SSS" Format으로 표시.
    (DateTimeFormatter 클래스 사용. 맨 뒤의 SSS는 1/1000초.)
  * 사진 정보 : “;”로 분리된 (image id, image 파일 이름, image tags)
  * 사물 정보 : “[”와 “]”로 싸여진 복수 개의 (stuff id, 종류, 이름, 태그). 생략 가능
  * 태그 : picture에 대한 태그 (단어들의 리스트). 생략 가능
  * 컴멘트 : picture에 대해 자유롭게 붙일 수 있는 설명. 생략 가능
  * 시각을 제외한 나머지 필드는 String으로 처리



- 앞뒤의 공백문자는 무시
- 빈줄, 또는 처음을 //로 시작하면 comment 처리
- 정보 사항은 모두 영어로. (즉, 한글은 사용하지 않음)
- 파일에 나오는 사진 수는 100개 이하로 가정한다.

7) 잘못된 파일명, 잘못된 규칙으로 기술된 파일 내용 등 오류에 대해 콘솔에 보고하여야 함.
   - 즉, 잘못된 입력 경우에 대처.

8) main()을 포함하는 별도의 Test 클래스를 만들어서 테스트. (뒤의 AutoTest 참조)

2. 목적

- Java String 구조 사용법 연습
- File I/O 사용 연습
- 인터페이스 정의를 통한 프로그램 배분/검사 등 분업화된 프로그래밍 맛보기

3. 추진 방법

0) **제대로 하려면, 일찍 시작하여야 한다.**
   - 마감에 쫓기면 자기 프로그램을 뒤돌아볼 수 없다.
   - 먼저 하는 것이 절대로 손해 보는 것이 아니다. 자신이 직접 겪은 실행 착오는 '약'
1) 문제 정의에 입각해서 프로그램 구조 설계. 즉, 사용할 클래스들 정의.
   - Picture, PictureList, Stuff, StuffList
2) 사진/사물 정보 기술방법 ("위 1-6)")이해
3) String이 제공하는 method들 공부

--- 정상 입력 파일에 대한 동작 완성 ----------

4) 1단계로 파일에서 가장 간단한 형태 한 줄을 읽어서 정해진 필드를 분리하는 파싱 메소드를 작성하고, 그 결과에 해당하는 **하나의 사진/사물 정보**(class Picture) **인스턴스/객체를** 만드는 작업을 완성.
5) 시간 처리에 Java가 제공하는 LocalDateTime class를 사용하면, LocalDateTime.parse() 메소드로 쉽고 정확하게 파싱 가능
6) print()를 코딩하여 제대로 파싱을 하는지 확인
7) 여러 줄의 파일 입력, 즉, 여러 사진 정보들의 처리 코딩.
   - 즉, PictureList 코딩

--- 비정상 입력 파일에 대응하도록 확장 ----------

8) comment 라인을 처리하는 문장을 추가해서 파일 읽기 메소드를 완성
9) timestamp가 지정이 안 된 입력, 시간이 형식이 맞지 않는 사진, ID가 중복되는 사진 등을 점검하고 출력해서 알려주는 기능 추가
10) 각종 정상/비정상 입력에 대해 동작 점검 및 프로그램 수정
11) 비정상 입력 발견 뒤, 나머지 입력을 계속할지 여부는 개인이 결정. 즉, 평가 대상이 아님.

---

12) **구조 > 기능 > 성능**
    구조를 맞추지 않고 스트링 처리만 하면, 이어지는 프로젝트에 적용할 수가 없음.
    제대로 구조를 완성하는 것이 무엇보다 중요하며, 이를 위해서는 문제의 이해가 선행되어야 함.
13) **스스로 충분히 디버깅을 수행한 후, 자동 실행검사 (AutoTest) 이용해서 최종 점검**
14) 리포트 작성

4. AutoTest를 이용한 구현 검증 (배포된 별도 pdf 설명 자료 참조)

1) 주어진 문제의 요구사항을 만족하는지를 웹을 통해 자동 검사
   - 인터페이스 요구 사항, 즉, 앞의 1-1), 1-2), 1-3), 1-4) 규정을 준수
   - 기능적 요구 사항 만족. 즉, 앞의 1-6)부터 7)까지 준수
2) 검사 방법 (배포 자료 참조)
   a. 자체 테스트용 main()이 포함된 **class를 제외한** 나머지 class들을 Eclipse export 기능을 이용하여 "jar"를 파일로 만든다. (예, TestPictureSearch.jar)
   - 주의: 프로젝트 전체를 export 하지 말고, 소스 파일들을 선택해서 export. (배포 자료 참조)

b. 165.194.35.156에 접속하여 준비한 jar 파일, 즉, TestPictureSearch.jar를 upload하고 테스트 결과를 확인한다.
   - 서버 소프트웨어는 수년간 검증이 된 것임. 즉, 테스트가 이상한 경우는 없음.
   - 과부하로 서버시스템에 접속이 안 될 경우, 비상 서버인 165.194.35.160 에 접속.

c. 먼저, 정상 입력 case에 집중해서 AutoTest를 수행하여 완성한다.

d. 비정상 모든 항목의 결과가 "Correct Answer"와 **유사하게** 나오면, 구현이 올바르게 된 것이므로 구현을 마치고, 그렇지 않은 경우 구현을 수정하여 다시 AutoTest를 수행한다.

5. 평가항목
   - 프로그램의 동작 여부 (70%)
     * 동작 여부를 확연히 알아볼 수 있는 결과. 즉, AutoTest 결과 화면
     * 본인이 작성한 프로그램의 진위 여부를 판단하는데 도움 되는 인증 자료 포함
   - 리포트 적정성 (30%)
     * 설계 및 구현 시 고민 했던 사항과 결정 내용을 간략히 기술
       + 본인이 직접 프로그램을 했음을 자연스럽게 입증
     * 평가자 입장에서, 평가사항을 찾기 쉽게 작성하였는가?

6. 리포트 제출
   1) 기한 : 04/17(수) 까지
      - 최종 테스트 과정 화면, 설계 노트, 소스 프로그램, 자체평가표
      - 하나로 묶은 리포트를 eClass로 제출

<참고 1> file 입력 및 try/catch 예

```
File file = new File("Picture-normal.data");

try {
        input = new Scanner(file);  // file is an instance of File
}
catch(Exception e) {
        // print error message and return;
}

while (input.hasNext()) {
        String line = input.nextLine();
        // 사진 정보 분해(파싱)
}
```

<참고 2> 사진/사물 (picture) 예 및 향후 검색 예