과제 5

사진, 사물 검색 GUI 프로그램

20232907 정현승

목차

1. GUI 설계 방법

GUI 설계는 다섯 클래스로 나누어 설계하였다. 표현해야 하는 **3개의 화면은 전부 다른 클래스**로 설계하였고, 여기에 **사진 하나를 담는 Panel 하나와 Stuff 하나를 담는 Panel을 별도의 클래스**로 분리하였다. Panel을 그릴 때 **과제 3의 클래스를 이용**했으며, 이때 만든 객체를 활용하는 것뿐만 아니라 **이 클래스를 기준으로 Panel을 설계**했다. 이 과제 3의 클래스 중에서 자주 쓰고 복잡해 보이는 클래스 위주로 Panel을 그리는 클래스를 분리하였으며(따라서 Picture 클래스와 Stuff 클래스 부분을 분리하였다. PictureList는 그 자체로 Frame이므로 논외이다), Tag 정보는 간단하므로, Image 정보는 화면에 표시되지 않으므로, StuffList 또한 화면에 표시되지 않으므로 Panel 그리는 부분을 별도의 클래스로 분리하지 않았다. 이렇게 되면 생성자의 코드 길이가 길어질 수 있겠으나, 어차피 저번 과제 3에서 문자열을 읽어 Picture 클래스를 만든 코드보다 길지 않으므로 문제가 되지 않는다고 생각하였다. 각 Panel의 부분별로 메서드를 나누는 것도 오히려 적합하지 않다고 판단해 대부분의 GUI 코드를 생성자에 넣었다. 다만 이러한 설계는 나중에 변경될 수 있다. 특히 event handling을 추가할 때, 지금도 event handling을 고려한다고 버튼과 JTextField를 멤버변수에 포함되도록 짜기는 했으나, 아직 event handling에 대한 자세한 것을 알지 못하는 상황인지라 event handling을 진행할 때 필요한 것이 현재 작성한 것과 완전히 다르면 현재의 설계에서 완전히 뜯어고치게 될 수도 있다.

GUI 설계 시 항상 같은 글자가 들어가는 JLabel같은 건 static 변수로 두고 하나만 만들어서 이를 돌려쓰는 방식으로 최적화를 진행할 수도 있으나(예를 들면 Stuff 정보를 나타내는 Panel에서 "type"을 표시하는 JLabel을 static 변수로 두어 모든 Stuff 정보 Panel이 하나의 JLabel을 공유한다든가 하는 등) 이는 과제가 원하는 방향이 아니라 생각해 진행하지 않았다.

이번에도 Panel을 먼저 코딩하고 이후 Frame을 코딩하는, Button-up 방식의 설계를 적용했다. 또 우선 만들어 실행해 본 후 실행 결과 디자인을 보고 코드를 조금씩 수정하는 방식으로 진행하였다.

GUI를 설계하면서 과제 3의 코드와도 연결했고, 또 과제 3의 코드 일부분을 수정했다. 이번 GUI에 필요한 부분은 GUI 설계를 진행하면서 수정했고, 그 외 부분은 GUI 코딩을 완료한 후 시행하였다.

A. StuffPanel 설계

StuffPanel은 Stuff 정보를 표시하는 Panel이다. 기본적인 형식의 Panel은 **생성자에서 Stuff 객체 하나를 인자로 받는** 것이고, 이 객체를 받으면 그 Stuff의 type, name, tags를 표현하는 Panel을 생성하는 형태이다. 이 Panel의 형태는 한 줄의 왼편에 type/name/tags를 구분하는 JLabel이 있고,

오른편에 그 내용이 적힌 JLabel이 있는 형태가 3줄 반복되는 형태이다. 이때 JLabel이 가로 2개 세로 3개 들어간다고 Panel의 Layout을 GridLayout으로 설정한 후 JLabel 6개를 한 번에 넣으면, 왼편에 위치한 JLabel이 오른편에 위치한 JLabel만큼 커지는 문제를 발견하였다. 왼편의 JLabel에는 각각 type, name, tags가 들어가는데, 최대 4글자만 필요한 곳에 그보다 긴 분량의 내용이 들어가는 오른편의 JLabel과 크기가 같은 점은 옳은 설계를 하지 않은 것으로 생각해서 설계를 수정했다. StuffPanel의 Layout을 **BorderLayout으로 수정하고, 왼편에 들어가는 JLabel끼리 모아 JPanel을 만들고 오른편에 들어가는 JLabel끼리 모아 JPanel을 따로따로** 만들었다. 그리고 이 **두 JPanel을 각각 WEST와 CENTER에 넣으면 왼편에 들어가는 JLabel은 딱 4글자 들어가는 크기로 만들어지고, 그만큼 오른쪽에 들어가는 내용의 크기가 커지게 된다. 즉 본래의 의도대로 모양을 만들 수 있다.** 따라서 그렇게 설계했다. 여기에 추가로 오른쪽에 들어가는 JLabel은 글씨체의 Bold체 설정을 해제했고 LIGHT_GRAY 색의 테두리도 둘러 주었다. Bold체 설정 해제는 Font 설정을 처음부터 직접 하는 방식으로 했는데, 이때 Bold체 부분을 제외하고 글씨체와 글자 크기는 기존과 똑같은 것으로 직접 다시 설정해야 한다. 따라서 글씨체는 강의자료에서 Font를 설명할 때 나와 있던 SensSerif를 사용했더니 Font 설정을 하지 않은 글씨체와 일치하는 것 같아 이를 그대로 사용했고, 크기는 10을 썼더니 조금 작은 것 같아 12로 수정했다. 또 Stuff 여러 개가 들어가면 서로 구분해 주어야 하므로, Panel 전체에 GRAY 색의 테두리를 둘러 주었다.

위와 같은 방식으로 Stuff 하나를 받아 Panel 하나를 만들었다. 그러나 설계를 더 하면서 보니 사진을 더하거나 검색할 때 **Stuff 정보를 처음부터 입력받는 상황**이 존재했다. 어쨌든 Stuff에 대한 Panel이라는 점은 둘이 같고, 둘의 GUI 모양도 같고, Stuff 정보를 입력받는 상황과 출력하는 상황은 독립적으로 일어나지 않을 것이라고도 생각해 이 **Stuff 입력 Panel도 같은 클래스에 작성**했다. Stuff를 입력받을 때는 아직 Stuff 정보가 없는 상황이므로 생성자의 인자는 없으며, Panel 생성 시 Stuff 정보를 표시하는 위의 Panel과는 생성자의 인자 유무로 판단하도록 했다. 이 Panel도 위의 Panel과 크게 다른 부분은 없으며, 오른편에 있던 JLabel이 JTextField로(입력을 받아야 하기 때문) 변경된 것 말고는 차이가 없다. 다만 앞서 말했듯 위 정보 출력 Panel과의 연계, 그리고 다음에 추가할 event를 고려해, 오른편에 있는 Component 3개는 클래스의 멤버 변수로 빼 놓았다.

B. PicturePanel 설계

PicturePanel은 사진 정보 하나를 표시하는 Panel이다. 위쪽에 시각과 태그, 왼쪽에 사진, 오른쪽에 Stuff 목록, 아래쪽에 comment가 들어가야 한다. 상하좌우에 들어가야 하는 게 명확하기 때문에 본 Panel의 Layout은 BorderLayout으로 했다. 또한 다른 Panel과는 달리 사진과 Stuff, 이와 시간 정보, 설명 사이에도 일정한 간격이 존재한다는 걸 확인해 간격 설정도 진행했다.

우선 위쪽에는 두 가지가 들어가야 하므로 별도의 Panel을 만들어 두 정보를 넣은 뒤 본 Panel에 add 했다. 별도 Panel의 Layout은 GridLayout으로 했는데, 위 StuffPanel을 처음 코드 작성할 당시 왼쪽 JLabel의 크기와 오른쪽 JLabel의 크기가 같은 것을 보고 여기에 적용하기 괜찮다고 생

각해 그렇게 진행했다. 처음에는 두 정보(JLabel)가 양쪽 끝에 있으므로 BorderLayout으로 해서 중앙은 비워두고 시각 정보와 태그 정보를 왼쪽과 오른쪽에 넣었다. 그러니 태그 정보가 오른쪽 정렬이 되고, 또 태그 문자열 길이만큼만 JLabel이 만들어졌다. 이것을 보고 "이건 내가 원하던 게 아닌데" 하는 생각을 하다가, 위 StuffPanel에 GridLayout이 적용된 결과가 왼쪽과 오른쪽에 들어가는 JLabel의 크기가 같은 것을 보고 이를 여기에 적용하기로 했다. 또 두 JLabel이 정말 같은 크기인가 의문도 가지긴 했으나, 막상 화면에 찍어본 결과를 보니 매우 적합한 것 같아 이대로 진행했다. 이 점 외에도 tag가 아예 없으면(빈 문자열) tag JLabel이 보이지도 않는 문제가 발생하였는데, tag 찾은 결과가 빈 문자열이면 공백 문자를 넣어주는 것으로 문제를 해결했다. 또 tag에는 StuffPanel처럼 Bold체 설정을 해제해 주었으며(방식은 위 StuffPanel에서 쓰던 것과 같다), LIGHT_GRAY 색의 Border를 그려주었다.

사진 정보는 ImageIcon으로 읽어 크기를 조정한 후 그 결과를 다시 ImageIcon에 담아 WEST에 넣었다. 사진을 먼저 ImageIcon으로 읽고 거기서 getImage를 하고 크기를 조정한 결과를 가지고 new ImageIcon을 다시 하는 것은, 비효율적이겠지만 떠오른 방법이 그것밖에 없었기 때문이다. 단순히 getImage().getScaledInstance(int, int, int)를 호출만 하니 그 결과가 ImageIcon에 적용되지 않았다. getScaledInstance의 반환값이 Image인 것을 보고, 이를 이용해 ImageIcon을 만들어야겠다고 생각했으며, ImageIcon 클래스의 코드를 찾아보니 실제로 Image를 인자로 받아 ImageIcon을 생성하는 생성자가 있어, 크기를 조정한 결과를 기준으로 다시 ImageIcon을 만들면 되겠다고 생각했다. 여기에 더해 중간에 호출하는 getImage() 메서드가 불필요하다고 생각하여 Image를 생성하고 크기를 조정한 결과를 ImageIcon에 넣을 생각으로(즉 ImageIcon 호출은 한 번만), 파일 경로를 주면 Image를 읽는 생성자가 있는지 확인하기 위해 Image 클래스의 코드를 확인해 본 결과 애초에 Image 클래스는 abstract 클래스라 생성자 호출 자체가 불가능했다. 따라서 ImageIcon 클래스를 두 번 만드는, 복잡하고 비효율적인 코드가 만들어지게 되었다. 코드만 보면 이 과정을 한 번에 이해하기 어려울 수도 있는데, 이 코드는 파일 경로로 ImageIcon을 만들고 getImage() 메서드, getScaledInstance(int, int, int) 메서드를 연속으로 사용하고, 이 결과를 가지고 새로운 ImageIcon을 만든 후 이 결과를 가지고 새로운 JLabel을 만드는 형태이다. 또 이 메서드는 인자가 3개이다. 앞의 2개 인자는 크기를 나타내는 상수로 보여 직접 수정했지만, 맨 뒤에 있는 건 용도를 알 수 없었다. 이 부분은 혼자서는 해결할 방법이 없다고 생각하여 인터넷에서 정보를 찾아보아 해결했다. 찾아보니 "이미지 품질 유지가 우선이면 Image.SCALE_AREA_AVERAGING을 활용하면 된다"[1] 라고 하기에 이를 활용했다. 이렇게 만든 ImageIcon을 CENTER가 아닌 WEST에 넣었는데, 바로 다음에 설명할 StuffList를 EAST에 넣고 사진을 CENTER에 넣으면 StuffList는 오른쪽에 조그맣게 만들어지고, 사진은 정중앙 넓은 공간에 들어가나 양쪽 여백이 많이 남아 보기 불편한

[1] 최범균, 「[팁] 이미지 크기 변환할 때 품질 유지하기」, 『자바캔(Java Can Do IT)』, 2006. 6. 15. (https://javacan.tistory.com/entry/124)

상황이 되었다. 따라서 크기가 고정된 사진을 WEST에 넣고 StuffList를 중앙에 넣었다.

StuffList는 Stuff의 모음이나, 이미 앞에서 Stuff 하나의 Panel을 만들었기에 Stuff 목록을 보고 Stuff의 수만큼 Panel을 만들어 준 후 이를 JScrollPane에 담아 CENTER에 추가해 주면 된다. Stuff가 매우 많을 것을 고려해 JScrollPane을 붙였다. 이러고 실행해 보니 Stuff의 수에 따라 각 Panel의 크기가 바뀌는 문제가 있었지만, 이 정도는 큰 문제가 되지 않을 것으로 생각하고 무시했다. setSize 메서드를 이용해 각 Panel의 크기를 변경하려는 시도도 했지만 변경된 크기가 적용되지 않아 이 문제를 처리하지 않는 것을 선택했다. 다만 이렇게 두면 Stuff 수가 많아질 때 스크롤이 생기지 않고 Panel의 크기만 늘어나는 문제를 발견하였다. 원인은 후술할 PictureListSection에 JScrollPane이 적용되어 이 Panel의 길이가 무한해진 것으로 추측했다. 이 원인대로면 Stuff를 담은 JScrollPane의 크기를 제한해야 하나 떠오른 방법이 없어 결국 인터넷을 찾아가며 답을 얻었다. 찾아보니 setPreferredSize를 사용하면 되는 쉬운 문제였다.[2] 처음에는 GridBagLayout 사용을 해야만 해결할 수 있는 문제라고 생각해 이를 위주로 검색했으나 setPreferredSize 사용이 문제임을 확인한 이후 관련된 코드는 전부 삭제했다.

South에는 comment 정보를 넣어주었다. 여기도 빈 문자열이 들어올 때의 처리와 Border를 입히는 작업을 진행했다.

위와 같은 방식으로 PicturePanel 설계를 완료했다.

C. PictureListSection 설계

PictureSection은 프로그램의 메인 화면으로, 사진 목록과 다른 기능으로의 접근이 가능한 여러 버튼이 포함된다. Layout은 위쪽, 중앙, 오른쪽에 들어가야 하는 게 정해져 있으니 BorderLayout을 사용하였다. 그러나 사진 Panel을 생성하는 작업이 PicturePanel로 분리되었기 때문에, 사진 Panel 여러 개를 생성해 JScrollPane에 넣고 버튼 몇 개 만들어주면 설계는 끝난다.

사진 정보는 PictureList 클래스를 인자로 받아오면서 가져왔는데, 이 부분에서 과제 3에서 작성한 클래스를 사용하게 되었다. 이 PictureList 클래스는 처음에 들어온 전체 List를 저장하는 멤버변수와 현재 표시할 List를 저장하는 멤버변수 총 2개를 두었으며, 이는 검색 작업이 자주 일어날 것임을 고려한 설계이다. 사진 패널 여러 개를 생성해 JScrollPane에 넣는 작업은, 이후 사진 검색을 진행할 때마다 같은 작업을 다시 진행할 게 뻔하니, 즉 반복이 많이 일어날 것이 예상되니

[2] SosoCEO,「[자바 swing] java JTable, JScrollPane의 size 결정 - setPreferredSize」,『네이버 블로그』, 2013. 3. 8. (https://blog.naver.com/since201109/150162040984)
예숑,「사이즈 조절과 테두리」,『yejin.log』, 2024. 1. 16. (https://velog.io/@jdc05163/%EC%82%AC%EC%9D%B4%EC%A6%88-%EC%A1%B0%EC%A0%88%EA%B3%BC-%ED%85%8C%EB%91%90%EB%A6%AC)

JScrollPane을 멤버변수로 빼는 것과 동시에 별도의 메서드로 분리했다. 검색 시 currentList 멤버 변수를 수정하고 사진 패널 여러 개를 생성하는 메서드를 호출한 뒤 repaint()를 사용하면, 그 즉시 검색 결과가 적용될 수 있도록 한 것이다.

오른쪽에 들어가는 버튼은 5개를 단순히 BorderLayout.EAST에 추가하다 보니 덮어씌워지는 문제를 발견해 Panel을 이용했고, 각종 버튼과 PictureList는 event 추가를 대비해 클래스 멤버 변수로 설정하였다. Search를 진행할 때 현재 PictureList에서 진행하고, 모든 사진을 보이는 버튼을 누르면 저장하고 있던 모든 사진을 보여야 하므로 PictureList는 전체 사진과 현재 사진 각각 저장할 용도로 2개를 놓았다. 그리고 ADD 버튼과 SEARCH 버튼을 누르면 새로운 창이 나오도록 event를 설정했다. event 설정은 별도의 클래스를 빼서 작성하는 방식이 아니라, 간단하게 new ActionListener(){public void actionPerformed (ActionEvent e){/*(내용)*/}} 방식으로 작성했다. 이 방식은 GUI 강의자료 32페이지 왼쪽 하단에 나온 방식이기도 하며, 아직은 event가 복잡하지 않기 때문에 클래스를 분리해 작성하지 않았다. **마지막으로 DefaultCloseOperation을 설정한 후 화면이 보이도록 setVisible(true)로 설정했다.**

이렇게 일단 만들고 화면을 띄워보며 창의 크기를 조절했다. 또 화면의 이름이 없으니, 창이 허전해 보여 Frame의 title도 추가하였다.

위와 같은 방식으로 PictureListSection 설계를 완료했다.

D. AddPictureSection 설계

AddPictureSection은 사진을 추가할 때 띄워지는 화면으로, 사진 파일 입력과 각종 정보를 입력하는 창이 포함된다. 이 Section은 사진 정보를 입력받는 Panel과 버튼이 있는 Panel로 나누고 두 Panel을 넣는 방식으로 진행하였는데, 이렇게 나누지 않으면 아래 두 개의 버튼을 넣을 방법이 없다고 판단했기 때문이다. 각 Panel과 전체 Section의 Layout은 전부 BorderLayout으로 설정했다. 이제 각 Panel 설계를 설명해야 하는데, 설명과 이해의 편의를 위해 각각 '사진 정보'와 '버튼이 있는 Panel'로 나누어 부른다.

사진 정보 Panel의 위와 아래에 들어가는 것(시간, 태그, 설명)은 각각 별도의 Panel을 만들었다. 먼저 위에 들어가는 시간과 태그 정보를 입력받는, JLabel 2개 JTextField 2개 총 4개의 Component를 GridLayout으로 설정된 Panel에 넣고, 사진 정보를 입력받는 Panel의 NORTH 부분에 넣었다.

그리고 사진 정보 왼쪽에 "Select Image File" 버튼을 넣고 오른쪽에 Stuff Panel의 List를 넣었다. 이 버튼을 BorderLayout의 WEST에 넣고 Stuff Panel을 CENTER에 넣었는데, 이런 형태는 PicturePanel과 같고 이유 역시 같다. StuffPanel은 아직 Stuff 정보가 무엇인지 모르는 상태이므로 Panel 생성 시 인자를 넣지 않았고, 위 StuffPanel에서 설명한 대로 Stuff 정보를 입력받는 형태의 Panel이 생성된다. **Stuff의 수가 더 늘어날 수 있기에, 클래스의 멤버 변수로 Stuff의 List를 만들**

고, List에 포함된 Object의 수만큼 StuffPanel을 만드는 방향으로 진행했다. 이 List에는 처음에 빈 Stuff 하나만 넣어 놓고, Stuff를 추가할 때마다 빈 Stuff를 더 만들어 두는 방향으로 만들 생각이다(아직 event handling을 하는 게 아니므로 생각만 해 놓고 구현은 하지 않았다). 이를 위해 List에 하나의 빈 Stuff만 넣어 두었다. 이때 기존 과제 3에 만들어 두었던 **Stuff 클래스는 빈 Stuff 생성을 허가하지 않으므로, Stuff 클래스 코드를 수정해 no-arg 생성자를 추가**하였다. 당연히 기존 Stuff 클래스는 빈 Stuff를 생각하고 만들지 않았으므로, Stuff 클래스의 다른 메서드를 호출하기에는 무리가 있으나 애초에 그렇게 활용할 생각이 아니기에 상관하지 않았다. Stuff 입력을 받고 나서 이 Stuff에 정보를 저장할지, 새로운 Stuff 객체를 만들어 기존에 만들어 둔 생성자를 사용할지는 아직 결정하지 못했으나 아직 event handling을 구현하는 단계가 아니므로 놔두었다.

사진 정보 아래쪽에는 comment를 입력하는 부분이 있다. 이 부분도 Panel을 만들어 JLabel과 JTextField를 넣어 사진 정보 Panel의 SOUTH에 add 하였다. 처음에는 이 Panel의 Layout을 FlowLayout으로 설정했는데 JTextField의 크기가 매우 작아져 이 안에 입력된 글을 알아볼 수 없다는 문제를 발견하였다. 이를 해결하기 위해 Layout을 GridLayout으로 변경했다. "comments"로 적힌 부분과 텍스트 입력 부분이 멀리 떨어져 있다는 문제는 있지만 그래도 JTextField의 크기 문제는 확실히 고쳐졌고, 앞에도 언급했듯 인터넷에서 정보를 찾아보는 것은 과제가 원하는 방향이 아닌 것 같아 진행하지 않았다. GridBagLayout도 별로 아는 게 없어 사용하지 않았다. (지금까지도 과제를 하며 Eclipse의 Java 코드 설명을 보는 행위와 기존에 알고 있던 지식 활용은 많이 했지만, 인터넷을 찾아보거나 코드를 가져오는 행위는 다른 방법으로 해결이 불가능해 반드시 필요한 상황이 아니면 하지 않았다. 이상한 거 베껴서 감점받는 것보다 아는 게 별로 없더라도 확실하게 하는 게 더 좋다고 판단했기 때문이다. 특히 copy 의심을 많이 하는 과목이므로 더더욱 그렇다. 인터넷을 찾아본 부분도 필요한 부분만 사용하고 필요 없는 부분까지 활용하지는 않았다. 저번 과제 4도 type-safe와 "<E>"를 넣는 건 몰랐지만 "이상한 것을 과제로 들고 오지 않았겠지" 하는 생각으로 일단 작성해 보니 문제가 없었고 또 실행해 보니 정답이라 인터넷을 찾아볼 일은 없었다. Eclipse의 Java 코드 설명은 확실하고 믿음직한 정보라고 생각해 여기까지는 활용했다.)

맨 아래 버튼이 있는 Panel은 각각의 버튼을 WEST와 EAST에 놓는 것으로 끝이다. 이 부분은 중앙을 비우고 양쪽 끝에 버튼이 있도록 해야 하므로 CENTER를 사용하지 않았다.

마지막으로 **사진 정보 Panel을 전체 Frame의 CENTER에, 버튼이 있는 Panel을 전체 Frame의 SOUTH에 각각 넣고 setVisible(true)로 설정**함으로써 AddPictureSection 설계를 완료했다. 위 PictureListSection처럼 Frame title도 설정했고, DefaultCloseOperation을 설정했더니 이 창을 닫았는데 메인 화면인 PictureListSection도 같이 닫히는 문제를 발견해 삭제하였다. 크기 설정은 최종 설계 결과를 계속 띄워보면서 조절했다.

E. SearchPictureSection 설계

SearchPictureSection은 사진을 검색(필터링)할 때 띄워지는 화면으로, 검색할 정보를 담는 창과 검색 조건(and/or)이 담긴 버튼이 포함된다. 이 Frame은 위아래로 크게 3개의 부분으로 나눌 수 있어, BorderLayout을 사용하였다.

NORTH에 들어가는 Time Search는 별도의 Panel에 JTextField 2개와 이를 설명하는 JLabel 3개를 GridLayout으로 넣어 완성하였고, TitledBorder로 감싸도록 했다.

CENTER에 들어가는 Keyword Search는 BorderLayout 설정의 별도 Panel을 만들었다. 태그 정보와 설명을 입력하는 부분을 GridLayout으로 묶어 이 Panel의 WEST에 넣고, Stuff 정보는 위 AddPictureSection처럼 StuffPanel을 가져와 이 Panel의 CENTER에 넣었다. 이번에는 Stuff를 추가하는 기능이 보이지 않아 하나의 StuffPanel만 넣을 수 있게 했다. 이 Panel의 아래에는 General Search 입력이 들어가는데, 이 부분을 FlowLayout으로 설정하니 위 AddPictureSection처럼 JTextField 크기가 매우 작아지는 문제가 발생했다. 여기서도 이 부분을 GridLayout으로 설정함으로써 문제를 해결했다.

SOUTH에 들어가는 버튼 3개도 별도 Panel을 만들었는데, 이 버튼은 크기가 정해져 있어서인지 FlowLayout으로 설정해도 아무런 문제가 발생하지 않고 중앙 정렬까지 자동으로 되었다. 따라서 FlowLayout으로 놔두었다.

이것도 AddPictureSection처럼 DefaultCloseOperation을 설정했다 지웠고, Frame title을 추가했으며 창의 크기는 결과를 띄워가며 수정했다. 이러한 방법으로 SearchPictureSection 설계를 완료했다.

F. 기존 코드 수정

기존의 코드에서 배열을 ArrayList로 변경한 것과 Comparable interface를 implements 한 것 외에도 GUI를 만들기 위한 몇 가지 변경 사항이 필요했다.

우선 **이번 GUI와는 상관없는 변경 사항**이다. 기존 코드에서 **배열을 ArrayList로 변경**했다. Tag, Picture, StuffList, PictureList 총 4개 클래스에 해당하며, 배열로 정의된 곳을 모두 List로 변경하였다. 메서드도 배열과 관련된 부분은 List에 적용될 수 있게 코드를 수정했다. 다만 코드가 추가된 부분보다는 삭제가 많은데, 배열과는 달리 ArrayList에 객체를 더할 때 배열을 새로 정의해 옮기는 수고를 하지 않아도 되고, 삭제할 때 삭제 대상 찾는 것도 관련 기능이 있기 때문에 이를 구현할 필요가 없었기 때문이다. for each 문은 배열과 List 모두 문제없이 작동하기에 고치지 않고 놔두었다. 다만 예외적으로 String.trim(String)의 반환값을 받을 때는 계속 배열로 받는데, 메서드의 반환값이 배열이기 때문이다.

그리고 **Comparable interface를 implement** 했다. Tag, Stuff, Picture 총 3개 클래스에 해당하며, 기존에 정의된 compareTo 메서드를 그대로 이용했다. compareTo 메서드가 2종류 있었던 picture 클래스에서는 당시 과제 요구사항이었던 Date 기준 비교를 compareTo 메서드로 삼았다. 다만 Comparable interface를 그냥 implement 하면 인자가 Object이므로, 명시적 casting을 진행해 주었다. casting 진행 전 instanceof 메서드를 이용해 casting이 문제없는지 미리 확인하고, 문제가 생기면 예외를 던졌다. implements comparable을 작성하니 parameterizing을 작업하라는 경고가 나왔지만 무시하고 진행했다. 이렇게 두 가지를 수정하니 **정렬을 진행하는 메서드도 수정이 필요**해졌는데, Arrays.sort는 배열만이 대상이 되어 List를 정렬할 수 없다. 따라서 List.sort 메서드를 활용했다. 인자가 필요하긴 하고 이 인자의 type을 알 수 없기는 하나 Eclipse의 설명을 보니 인자에 null을 주면, compareTo 메서드를 활용해 알아서 정렬한다고 쓰여 있어, 이렇게 활용했다. (앞서 언급했던 이유로 인터넷 검색을 이용해 다른 코드를 가져올 생각은 하지 않았다.) 이는 아래 그림 1.1에서 3번째 단락에 해당한다.

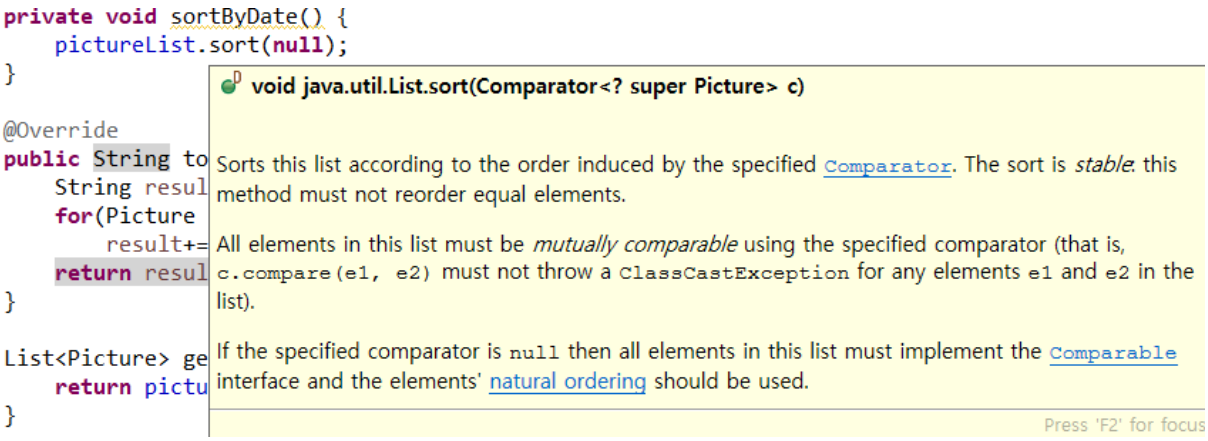

그림 1.1

compareTo 이외의 메서드를 기준으로 정렬하는 코드는 아직 사용하지 않기 때문에 주석처리를 진행했다(이를 구현하는 방법을 모르는 상태지만, 어차피 아직 필요가 없기에 주석으로 처리했다). 위의 수정 사항도 테스트를 진행했다. ArrayList를 사용하는 부분은 GUI 화면이 잘 띄워지는지 확인하며 테스트를 진행하고, Comparable와 sort 부분은 정렬이 잘 되는지 확인하는 방식으로, 별도의 코드를 작성해 테스트를 진행했다. 다만 이번 과제의 초점은 여기가 아니므로, 테스트는 간단하게(1번만) 진행했다.

다음은 **GUI를 띄우기 위한 변경 사항**이다. 다만 다른 건 크게 없고 앞에서 설명한 Stuff 클래스에 no-arg 생성자 추가하는 작업과 기존 클래스에서 getter를 만드는 작업 외에는 없다. 다만 이렇다 보니 6개 클래스 전부를 수정하게 되었다. Image 클래스마저도 파일 경로를 가져오는 getter가 필요해 코드를 수정하게 되었다. Tag 클래스는 getter 대신 toString() 메서드를 사용해 getter가 필요 없었으나, Comparable을 추가하며 코드를 수정한 부분이 존재하게 되었다. Picture의 시간 정보는 단순히 getter를 만든 게 아니라 저장된 시간을 문자열로 바꾸어서 반환하도록

만들었다. 여기서 바꾸지 않으면 어차피 GUI 생성 과정에서 바꾸어야 하기 때문이다. 그리고 clone 메서드는 사용하지 않고 저장된 그대로를 반환하도록 했다. 클래스 내부 멤버 변수가 getter를 통해 변경될 수도 있지만 아직은 여기까지 고려하지 않았다. 이 getter를 작성하면서 GUI 관련 클래스를 JFrame 또는 JPanel이 아닌 Picture, PictureList, Stuff를 상속받도록 하고 싶었지만(아니면 GUI 만드는 작업이 메서드 하나라는 점을 고려해 기존 클래스에 Panel을 생성하는 메서드를 넣는 방법도 존재한다) 이는 과제 6에서 할 단계라고 생각해 시행하지 않았다.

2. 테스트 결과

이번에는 단순히 GUI 모양만 만드는 것이기 때문인지 AutoTest도 없고, **화면 몇 개 띄우는 것 정도로 테스트도 끝**난다. 기존 코드 수정한 부분은 이번에 초점을 맞추는 부분이 아니므로 여기에 초점을 맞추어 테스트할 이유는 전혀 없다고 생각하며, 이러면 여러 번 상황을 바꾸어가며 테스트를 진행할 이유는 더더욱 없다. 다만 자신만의 사진 하나 추가를 요구하였기에, 이 점을 picture-normal.data 파일에 적용하고 테스트를 진행하였다. 추가한 그림의 Stuff 목록은 Stuff Panel의 JScrollPane이 작동되는지 확인하기 위해 많이 넣어 만들었다. 이외에도 sort와 Comparable 테스트를 위해 파일에 저장된 내용의 순서를 섞었다.

우선 파일을 읽을 대상인 picture-normal.data 파일에 저장된 내용은 다음과 같다.

```
// Picture information - saved by GUI
< m_2024-01-03_10:00:00:000 > < 2024-01-03_10:00:00:000 > < IMG2024-01-07_15:33:00:827; images/toolbox.jpg; ;  > <  [00000003; toolbox; my toolbox; #red ] [00000004; tool; wrench; ]> < #home #clean > <>
< m_2024-03-03_09:30:00:000 > < 2024-03-03_09:30:00:000 > < IMG2024-01-07_15:45:27:943; images/subway.jpg; ;  > <> < #subway > < to the school >
< m_2024-03-03_14:00:00:000 > < 2024-03-03_14:00:00:000 > < IMG2024-01-07_15:50:35:480; images/classroom.jpg; ;  > <  [00000007; room; class meeting; #study ]> < #class #lecture > <>
< m_2024-01-03_17:00:00:000 > < 2024-01-03_17:00:00:000 > < IMG2024-01-07_15:37:38:009; images/dinner.jpg; ;  > <  [00000005; food; dinner; #korean ]> < #friend #meet > < good restaurant >
< m_2024-03-03_14:20:00:000 > < 2024-03-03_14:20:00:000 > < IMG2024-01-08_19:12:57:384; images/eclipse.jpg; ;  > <  [00000008; study; eclipse; #java ]> <> <>
< m_2024-01-03_06:30:00:000 > < 2024-01-03_06:30:00:000 > < IMG2024-01-07_15:28:42:083; images/morning-excercise.jpg; ;  > <  [00000002; exercise; ; #morning ]> < #happy > < new year >
<m_2024-05-06_23:22:01:000><2024-05-
```

```
06_23:22:01:000><KakaoTalk_20240525_034441188;images/KakaoTalk_20240525_034441188.jpg;;
><[;moniter;;#LCD][;board;post;#platform1][;line;metro
line;#GyeonguiJoongang][;tool;wrench;#void][;stuff;testStuff;#testonly]><#delay#accident#power
Failure#Yongsan>< Yongsan station when power failure happened at Guri station >
// end of Picture information saved by GUI
```

그리고 테스트 코드는 다음과 같다. 정렬을 먼저 진행하고 GUI 화면을 띄우도록 했다. 테스트할 때는 private로 설정된 메서드를 임시고 default로 놓은 상태에서 진행했으며, 테스트 완료 후에는 다시 private로 돌려놓았다.

```
public class TestPicture {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		PictureList pictureList=new PictureList("picture-normal.data");
		pictureList.sortByDate();
		System.out.println(pictureList);
		System.out.println();
		StuffList.sortById();
		StuffList.print();
		PictureSection mainWindow=new PictureSection(pictureList);
	}
}
```

아래 그림 2.1과 2.2는 메인 화면의 모습이다. 그림 2.1은 사진의 스크롤을 내리지 않은 상태이고 그림 2.2는 스크롤을 내려 Stuff 정보 부분에 스크롤이 생성되는 모습을 담은 그림이다. 그림 2.2의 마지막 사진은 요구했던 자신만의 사진이며, 뭐로 할지 찾아보다 저번 5월 6일 학교에서 귀가할 때 경의·중앙선이 사고로 인해 크게 지연되었던 상황을 담은 사진을 사용했다. Stuff 5개를 달았는데 스크롤도 문제없이 작동된다는 점을 알 수 있다. 창 맨 위 "Show All Pictures" 버튼의 글씨에 테두리가 둘려 있는 건 가끔은 안 보일 때도 있고 보일 때도 있던데, 중요한 부분이 아니라고 생각해 수정하지 않고 두었다.

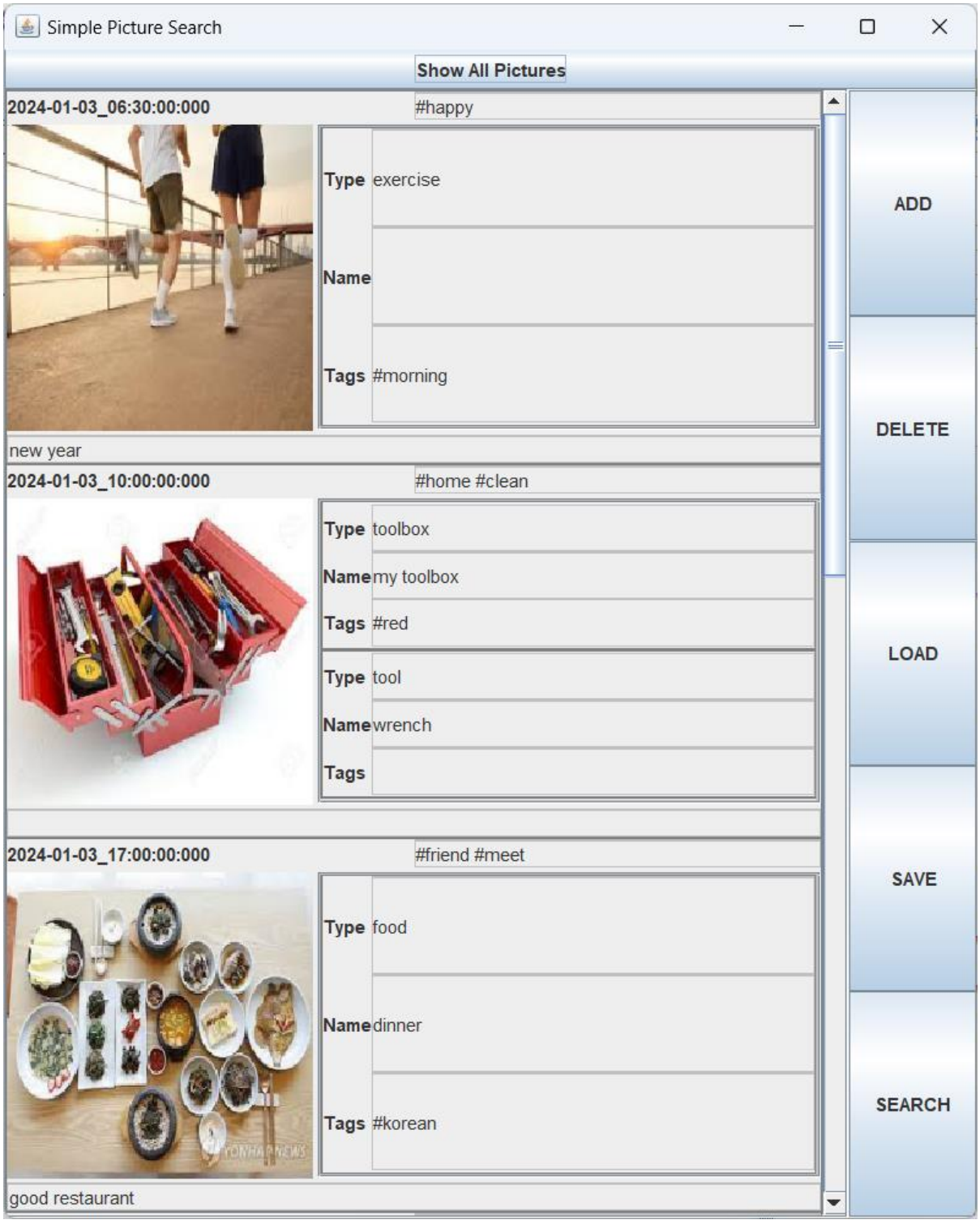
Simple Picture Search
Show All Pictures
2024-01-03_06:30:00:000
#happy
Type exercise
Name
Tags #morning
new year
2024-01-03_10:00:00:000
#home #clean
Type toolbox
Name my toolbox
Tags #red
Type tool
Name wrench
Tags
2024-01-03_17:00:00:000
#friend #meet
Type food
Name dinner
Tags #korean
good restaurant
ADD
DELETE
LOAD
SAVE
SEARCH

그림 2.1

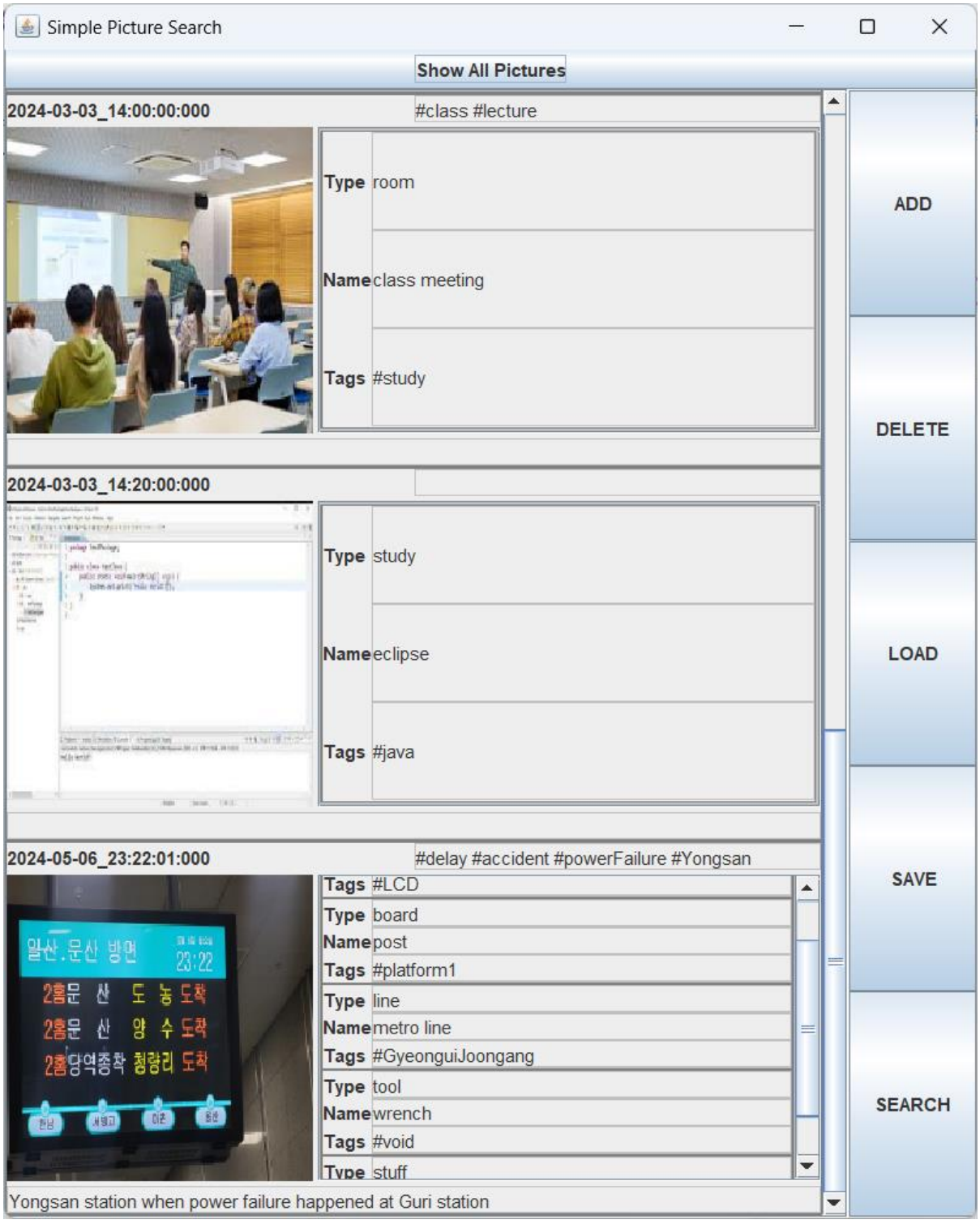
Simple Picture Search
Show All Pictures
2024-03-03_14:00:00:000
#class #lecture
Type room
Name class meeting
Tags #study
2024-03-03_14:20:00:000
Type study
Name eclipse
Tags #java
2024-05-06_23:22:01:000
#delay #accident #powerFailure #Yongsan
Tags #LCD
Type board
Name post
Tags #platform1
Type line
Name metro line
Tags #GyeonguiJoongang
Type tool
Name wrench
Tags #void
Type stuff
Yongsan station when power failure happened at Guri station
ADD
DELETE
LOAD
SAVE
SEARCH

그림 2.2

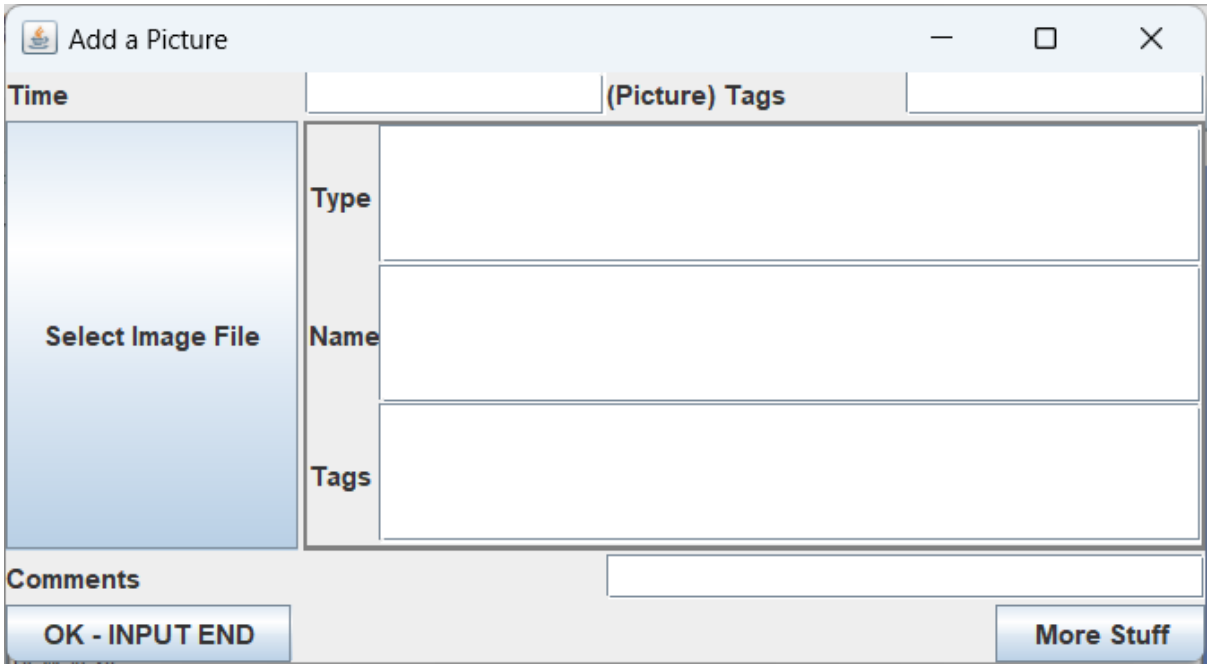

그림 2.3

위 그림 2.3은 그림 2.1이나 그림 2.2에서 오른쪽의 ADD 버튼을 눌렀을 때 나오는 화면이다. 구성요소 간 비율은 이상하지만 그래도 큰 틀은 문제없음을 확인할 수 있다. 버튼은 당연히 동작하지 않는다.

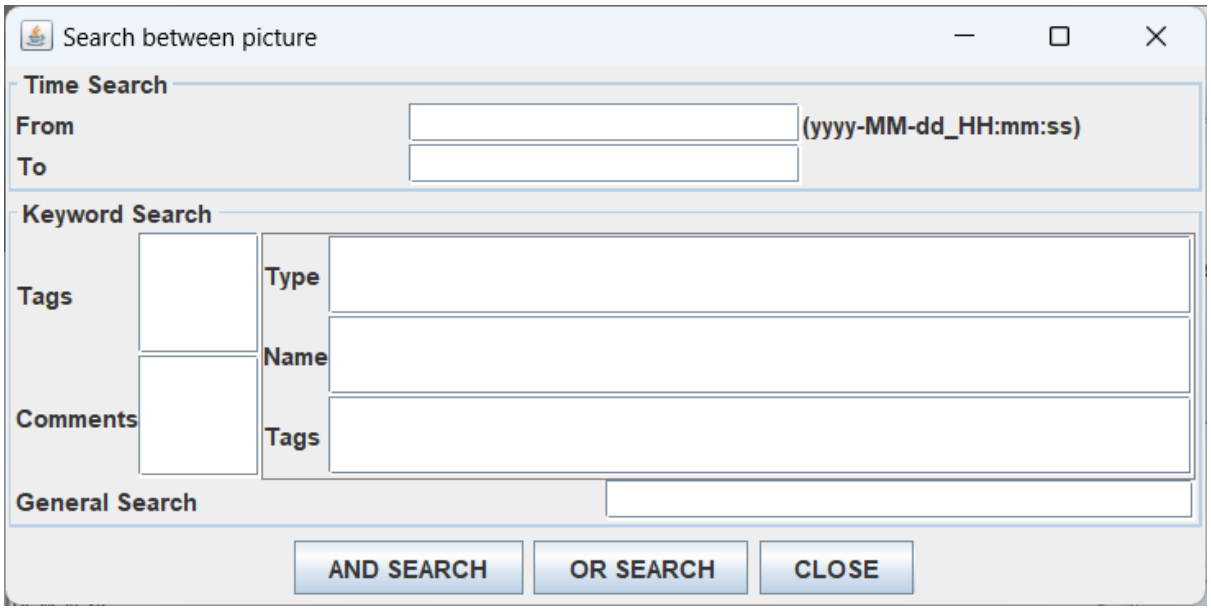

그림 2.4

위 그림 2.4는 그림 2.1이나 그림 2.2에서 오른쪽의 SEARCH 버튼을 눌렀을 때 나오는 화면이다. 역시 구성 요소 간 비율은 이상하지만 그래도 큰 틀은 문제없음을 확인할 수 있다. 버튼도 아직 동작하지 않는다.

```
<m_2024-01-03_06:30:00:000>        <2024-01-03_06:30:00:000>        <IMG2024-01-07_15:28:42:083;  images/morning-excercise.jpg;  ;  >  <[00000006;  exercise;  ;  #morning]> <#happy> <new year>
<m_2024-01-03_10:00:00:000>        <2024-01-03_10:00:00:000>        <IMG2024-01-07_15:33:00:827; images/toolbox.jpg; ; > <[00000001; toolbox; my toolbox; #red] [00000002; tool; wrench; ]> <#home #clean> <>
<m_2024-01-03_17:00:00:000>        <2024-01-03_17:00:00:000>        <IMG2024-01-07_15:37:38:009;  images/dinner.jpg;  ;  >  <[00000004;  food;  dinner;  #korean]> <#friend #meet> <good restaurant>
<m_2024-03-03_09:30:00:000>        <2024-03-03_09:30:00:000>        <IMG2024-01-07_15:45:27:943; images/subway.jpg; ; > <> <#subway> <to the school>
<m_2024-03-03_14:00:00:000>        <2024-03-03_14:00:00:000>        <IMG2024-01-07_15:50:35:480;  images/classroom.jpg;  ;  >  <[00000003;  room;  class  meeting;  #study]> <#class #lecture> <>
<m_2024-03-03_14:20:00:000>        <2024-03-03_14:20:00:000>        <IMG2024-01-08_19:12:57:384; images/eclipse.jpg; ; > <[00000005; study; eclipse; #java]> <>
<>
<m_2024-05-06_23:22:01:000>                                  <2024-05-06_23:22:01:000>
<KakaoTalk_20240525_034441188;   images/KakaoTalk_20240525_034441188.jpg;   ;   >
<[00000007; moniter; ; #LCD] [00000008; board; post; #platform1] [00000009; line; metro line; #GyeonguiJoongang] [00000002; tool; wrench; #void] [00000010; stuff; testStuff; #testonly]> <#delay #accident #powerFailure #Yongsan> <Yongsan station when power failure happened at Guri station>


All Stuffs now saved: < [00000001; toolbox; my toolbox; #red] [00000002; tool; wrench; ] [00000002; tool; wrench; #void] [00000003; room; class meeting; #study] [00000004; food; dinner; #korean] [00000005; study; eclipse; #java] [00000006; exercise;  ; #morning] [00000007; moniter; ; #LCD] [00000008; board; post; #platform1] [00000009; line; metro line; #GyeonguiJoongang] [00000010; stuff; testStuff; #testonly] >
```

위는 콘솔에 출력된 결과이다. 테스트 코드부터 콘솔 출력을 하도록 작성해 출력이 존재한다. Picture-normal.data 파일에 의도적으로 날짜 순서를 변경하였으나 문제없이 날짜순으로 정렬된 것을 확인할 수 있고, Stuff도 정렬되어 태그가 다르나 ID가 같은 2개의 Stuff [00000002; tool; wrench]가 이웃하게 되었음을 통해 정상적으로 정렬되었음을 알 수 있다.

위와 같이 테스트 결과 코드가 정상 작동한다는 것을 알 수 있다.

3. 소스코드

이번에 새로 작성한 GUI 코드 말고도, GUI를 만들며 기존 과제 3의 코드를 변경했으므로 변경된 코드도 전부 첨부하였다. 처음에는 달라진 부분만 넣으려고 했으나 그렇게 하면 알아보기는 어렵고 보고서 길이만 길어져 시행하지 않았다.

A. StuffPanel.java 소스코드

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;

import javax.swing.JComponent;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.LineBorder;

public class StuffPanel extends JPanel {

	private JComponent type;
	private JComponent name;
	private JComponent tags;

	StuffPanel(){
		super(new BorderLayout());
		setBorder(new LineBorder(Color.GRAY,1));

		JPanel left=new JPanel(new GridLayout(3,1));
		JPanel right=new JPanel(new GridLayout(3,1));

		left.add(new JLabel("Type"));
		type=new JTextField();
		right.add(type);

		left.add(new JLabel("Name"));
		name=new JTextField();
		right.add(name);

		left.add(new JLabel("Tags"));
		tags=new JTextField();
		right.add(tags);

		add(left,BorderLayout.WEST);
		add(right,BorderLayout.CENTER);
	}

	StuffPanel(Stuff stuff) {
		super(new BorderLayout());
		Font planefont=new Font("SansSerif",Font.PLAIN,12); //단순히
PLAIN 으로 변경
		LineBorder objectBorder=new LineBorder(Color.LIGHT_GRAY,1);
		setBorder(new LineBorder(Color.GRAY,1));

		JPanel left=new JPanel(new GridLayout(3,1));
		JPanel right=new JPanel(new GridLayout(3,1));

		left.add(new JLabel("Type"));
		type=new JLabel(stuff.getType());
		type.setBorder(objectBorder);
```

```
                type.setFont(planefont);
                right.add(type);

                left.add(new JLabel("Name"));
                name=new JLabel(stuff.getName());
                name.setBorder(objectBorder);
                name.setFont(planefont);
                right.add(name);

                left.add(new JLabel("Tags"));
                tags=new JLabel(stuff.getTags().toString());
                tags.setBorder(objectBorder);
                tags.setFont(planefont);
                right.add(tags);

                add(left,BorderLayout.WEST);
                add(right,BorderLayout.CENTER);
        }

}
```

B. PicturePanel.java 소스코드

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Image;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.border.LineBorder;

public class PicturePanel extends JPanel {

        PicturePanel(Picture picture) {
                super(new BorderLayout(3,3));
                LineBorder border=new LineBorder(Color.LIGHT_GRAY,1);
                Font planefont=new Font("SansSerif",Font.PLAIN,12); //단순히
PLAIN 으로 변경
                setBorder(new LineBorder(Color.GRAY,1));

                JPanel northPanel=new JPanel(new GridLayout(1,2));
                northPanel.add(new JLabel(picture.getTimeStampToString()));
                String tagString=picture.getTags().toString();
                if(tagString.isBlank()) tagString=" "; //이거라도 있어야 모양이 나옴
                JLabel tags=new JLabel(tagString);
                tags.setBorder(border);
                tags.setFont(planefont);
                northPanel.add(tags);
                add(northPanel,BorderLayout.NORTH);

                add(new JLabel(
```

```
                                new ImageIcon(new
ImageIcon(picture.getPictureFileName())
                                .getImage().getScaledInstance(200, 200,
Image.SCALE_AREA_AVERAGING))
                                ),BorderLayout.WEST);
                //new ImageIcon(read file) -> getImage -> getScaledInstance -> new
ImageIcon -> new JLabel
                JPanel stuffs=new JPanel(new
GridLayout(picture.getStuffList().size(),1));
                for(Stuff stuff:picture.getStuffList()) {
                        stuffs.add(new StuffPanel(stuff));
                }
                stuffs.setBorder(new LineBorder(Color.LIGHT_GRAY,1));
                JScrollPane stuffPane=new JScrollPane(stuffs);
                stuffPane.setPreferredSize(new Dimension(300,200));
                add(stuffPane,BorderLayout.CENTER);

                String commentString=picture.getComment();
                if(commentString.isBlank()) commentString=" ";
                JLabel comment=new JLabel(commentString);
                comment.setBorder(border);
                comment.setFont(planefont);
                add(comment,BorderLayout.SOUTH);
        }

}
```

C. PictureListSection.java 소스코드

```
import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;

public class PictureSection extends JFrame {

        private PictureList fullList;
        private PictureList currentList;
        private JScrollPane picturePane;
        private JButton showAll;
        private JButton addPicture;
        private JButton deletePicture;
        private JButton load;
        private JButton save;
        private JButton search;

        PictureSection(PictureList pictureList) {
                super();
                fullList=pictureList;
                currentList=pictureList;
                setLayout(new BorderLayout());
                setSize(650,800);
```

```
		setTitle("Simple Picture Search");
		showAll=new JButton("Show All Pictures");
		add(showAll,BorderLayout.NORTH);

		JPanel eastPanel=new JPanel(new GridLayout(5,1));
		addPicture=new JButton("ADD");
		addPicture.addActionListener(new ActionListener() {
			@Override
			public void actionPerformed(ActionEvent e) {
				new AddPictureSection();

			}
		});
		eastPanel.add(addPicture);
		deletePicture=new JButton("DELETE");
		eastPanel.add(deletePicture);
		load=new JButton("LOAD");
		eastPanel.add(load);
		save=new JButton("SAVE");
		eastPanel.add(save);
		search=new JButton("SEARCH");
		search.addActionListener(new ActionListener() {
			@Override
			public void actionPerformed(ActionEvent e) {
				new SearchPictureSection();
			}
		});
		eastPanel.add(search);
		add(eastPanel,BorderLayout.EAST);

		picturePane=new JScrollPane(drawPictureListPanelByCurrentList());
		add(picturePane,BorderLayout.CENTER);

		setDefaultCloseOperation(EXIT_ON_CLOSE);
		setVisible(true);
	}

	private JPanel drawPictureListPanelByCurrentList() {
		JPanel pictures=new JPanel(new GridLayout(currentList.size(),1));
		for(Picture picture:currentList.getPictureList()) {
			pictures.add(new PicturePanel(picture));
		}
		return pictures;
	}
}
```

D. AddPictureSection.java 소스코드

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.GridLayout;
import java.util.ArrayList;
import java.util.List;

import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.LineBorder;

public class AddPictureSection extends JFrame {

	private JTextField time;
	private JTextField tags;
	private JComponent image;
	private List<Stuff> stuffList;
	private JTextField comments;

	AddPictureSection() {
		super();
		setLayout(new BorderLayout(2,2));
		setSize(550,300);
		setTitle("Add a Picture");
		JPanel pictureInfo=new JPanel(new BorderLayout(2,2));

		JPanel north=new JPanel(new GridLayout(1,4));
		north.add(new JLabel("Time"));
		time=new JTextField();
		north.add(time);
		north.add(new JLabel("(Picture) Tags"));
		tags=new JTextField();
		north.add(tags);
		pictureInfo.add(north,BorderLayout.NORTH);

		image=new JButton("Select Image File");
		pictureInfo.add(image,BorderLayout.WEST);

		stuffList=new ArrayList<>();
		stuffList.add(new Stuff());
		JPanel stuffs=new JPanel(new GridLayout(stuffList.size(),1));
		stuffs.setBorder(new LineBorder(Color.GRAY,1));
		for(Stuff stuff:stuffList) {
			stuffs.add(new StuffPanel());
		}
		pictureInfo.add(stuffs,BorderLayout.CENTER);

		JPanel south=new JPanel(new GridLayout());
		south.add(new JLabel("Comments"));
		comments=new JTextField();
		south.add(comments);
		pictureInfo.add(south,BorderLayout.SOUTH);
		add(pictureInfo,BorderLayout.CENTER);

		JPanel buttom=new JPanel(new BorderLayout());
		buttom.add(new JButton("More Stuff"),BorderLayout.EAST);
		buttom.add(new JButton("OK - INPUT END"),BorderLayout.WEST);
		add(buttom,BorderLayout.SOUTH);
		setVisible(true);
	}

}
```

E. SearchPictureSection.java 소스코드

```
import java.awt.BorderLayout;
import java.awt.GridLayout;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.TitledBorder;

public class SearchPictureSection extends JFrame {

	private JTextField timeFrom;
	private JTextField timeTo;
	private JTextField tags;
	private JTextField comments;
	private JTextField generalSearch;
	private StuffPanel stuffToSearch;
	private JButton andSearch;
	private JButton orSearch;
	private JButton close;

	private static String dateTimeFormatToString="yyyy-MM-dd_HH:mm:ss";

	SearchPictureSection() {
		super();
		setSize(600,300);
		setLayout(new BorderLayout());
		setTitle("Search between picture");

		JPanel timeSearch=new JPanel(new GridLayout(2, 3));
		timeSearch.add(new JLabel("From"));
		timeFrom=new JTextField();
		timeSearch.add(timeFrom);
		timeSearch.add(new JLabel("("+dateTimeFormatToString+")"));
		timeSearch.add(new JLabel("To"));
		timeTo=new JTextField();
		timeSearch.add(timeTo);
		timeSearch.setBorder(new TitledBorder("Time Search"));
		add(timeSearch,BorderLayout.NORTH);

		JPanel wordSearch=new JPanel(new BorderLayout());
		JPanel otherObjects=new JPanel(new GridLayout(2,2));
		otherObjects.add(new JLabel("Tags"));
		tags=new JTextField();
		otherObjects.add(tags);
		otherObjects.add(new JLabel("Comments"));
		comments=new JTextField();
		otherObjects.add(comments);
		wordSearch.add(otherObjects,BorderLayout.WEST);

		stuffToSearch=new StuffPanel();
		wordSearch.add(stuffToSearch,BorderLayout.CENTER);
```

```
                JPanel options=new JPanel(new GridLayout());
                options.add(new JLabel("General Search"));
                generalSearch=new JTextField();
                options.add(generalSearch);
                wordSearch.add(options,BorderLayout.SOUTH);
                wordSearch.setBorder(new TitledBorder("Keyword Search"));
                add(wordSearch,BorderLayout.CENTER);

                JPanel searchButtons=new JPanel();
                andSearch=new JButton("AND SEARCH");
                searchButtons.add(andSearch);
                orSearch=new JButton("OR SEARCH");
                searchButtons.add(orSearch);
                close=new JButton("CLOSE");
                searchButtons.add(close);
                add(searchButtons,BorderLayout.SOUTH);

                setVisible(true);
        }

}
```

F. 이번에 수정한 Picture.java 소스코드

```
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.time.format.DateTimeParseException;
import java.util.List;

public class Picture implements Comparable{
        private String id;
        private LocalDateTime timeStamp;
        private Image pictureInfo;
        private List<Stuff> stuffList;
        private Tag tags;
        private String comment;

        private static DateTimeFormatter dateTimeFormat=
                        DateTimeFormatter.ofPattern("yyyy-MM-dd_HH:mm:ss:SSS");

        Picture(String imageFile){
                String[] pictureByString=imageFile.trim().split(">");
                if(pictureByString[pictureByString.length-1].isBlank()){
                        String[] tempPictureByString=new
String[pictureByString.length-1];
                        System.arraycopy(pictureByString, 0, tempPictureByString, 0,
pictureByString.length-1);
                        pictureByString=tempPictureByString;
                }
                if(pictureByString.length!=6) {
                        throw new IllegalStateException
                        ("Picture \""+ imageFile +"\" has something wrong with
number of \"<\" and \">\".");
                }

                pictureByString[0]=pictureByString[0].trim();
                if(pictureByString[0].startsWith("<")) {
```

```
                        id=pictureByString[0].substring(1).trim();
                } else {
                        throw new IllegalStateException
                        ("Picture \""+ imageFile +"\" has something wrong with
number of \"<\" and \">\".");
                }
                if(id.isBlank()) {
                        throw new IllegalStateException("id of Picture \""+
imageFile +"\" cannot be empty");
                }

                pictureByString[1]=pictureByString[1].trim();
                if(pictureByString[1].startsWith("<")) {
                        pictureByString[1]=pictureByString[1].substring(1).trim();
                        try {

        timeStamp=LocalDateTime.parse(pictureByString[1],dateTimeFormat);
                        } catch(DateTimeParseException e) {
                                throw new IllegalStateException
                                ("Cannot parse \"" + pictureByString[1] + "\" to
DateTimeFormat");
                        }
                } else {
                        throw new IllegalStateException
                        ("Picture \""+ imageFile +"\" has something wrong with
number of \"<\" and \">\".");
                }

                pictureByString[2]=pictureByString[2].trim();
                if(pictureByString[2].startsWith("<")) {
                        pictureInfo=new
Image(pictureByString[2].substring(1).trim());
                } else {
                        throw new IllegalStateException
                        ("Picture \""+ imageFile +"\" has something wrong with
number of \"<\" and \">\".");
                }
                if(id.isBlank()) {
                        throw new IllegalStateException
                        ("pictureInfo of Picture \""+ imageFile +"\" cannot be
empty");
                }

                pictureByString[3]=pictureByString[3].trim();
                if(pictureByString[3].startsWith("<")) {

        stuffList=StuffList.newStuffs(pictureByString[3].substring(1).trim());
                } else {
                        throw new IllegalStateException
                        ("Picture \""+ imageFile +"\" has something wrong with
number of \"<\" and \">\".");
                }

                pictureByString[4]=pictureByString[4].trim();
                if(pictureByString[4].startsWith("<")) {
                        tags=new Tag(pictureByString[4].substring(1).trim());
```

```
                } else {
                        throw new IllegalStateException
                        ("Picture \""+ imageFile +"\" has something wrong with
number of \"<\" and \">\".");
                }

                pictureByString[5]=pictureByString[5].trim();
                if(pictureByString[5].startsWith("<")) {
                        comment=pictureByString[5].substring(1).trim();
                } else {
                        throw new IllegalStateException
                        ("Picture \""+ imageFile +"\" has something wrong with
number of \"<\" and \">\".");
                }
        }

        Picture(String id, LocalDateTime timeStamp, Image pictureInfo, List<Stuff>
stuffList,
                        Tag tags, String comment) {
                this.id = id;
                this.timeStamp = timeStamp;
                this.pictureInfo = pictureInfo;
                this.stuffList = stuffList;
                this.tags = tags;
                this.comment = comment;
        }

        public void print() {
                System.out.println(this.toString());
        }

        private boolean ifStuffExist(Stuff stuff) {
                for(Stuff savedStuff:stuffList) {
                        if(stuff.equalByTypeAndName(savedStuff))
                                return true;
                }
                return false;
        }

        private void addStuff(String stuff) {
                addStuff(StuffList.addStuff(stuff));
        }

        private void addStuff(Stuff stuff) {
                stuffList.add(stuff);
        }

        private void deleteStuff(String stuff) {
                deleteStuff(new Stuff(stuff));
                //stuffList 등록을 피하기 위해 Stuff를 바로 씀
        }

        private void deleteStuff(Stuff stuff) {
                //StuffList 클래스에서는 삭제되지 않음
                stuffList.remove(stuff);
        }
```

```
	private boolean ifTagExist(String tag) {
		return tags.ifTagExist(tag);
	}

	private boolean ifTagExist(Tag tag) {
		return tags.ifTagExist(tag);
	}

	private void addTag(String tag) {
		tags.addTag(tag);
	}

	private void deleteTag(String tag) {
		tags.deleteTag(tag);
	}
	//비교대상이 greater->negative
	@Override
	public int compareTo(Object o) {
		if(!(o instanceof Picture)) throw new IllegalStateException(o+" is
not Picture");
		Picture picture=(Picture) o;
		return this.timeStamp.compareTo(picture.timeStamp);
	}

	int compareToById(Object o) {
		if(!(o instanceof Picture)) throw new IllegalStateException(o+" is
not Picture");
		Picture picture=(Picture) o;
		return id.compareTo(picture.id);
	}

	private int tagLength() {
		return tags.getLength();
	}

	boolean equalById(Picture picture) {
		return this.id.equals(picture.id);
	}

	@Override
	public String toString() {
		String stuffListString="";
		for(Stuff stuff:stuffList) {
			stuffListString+=stuff+" ";
		}
		return "<" + id + "> <" + timeStamp.format(dateTimeFormat) + "> <"
+ pictureInfo
				+ "> <" + stuffListString.trim() + "> <" + tags + ">
<" + comment + ">";
	}

	static void setDateTimeFormat(DateTimeFormatter dateTimeFormat) {
		Picture.dateTimeFormat = dateTimeFormat;
	}
```

```
	@Override
	public boolean equals(Object obj) {
		if (this == obj)
			return true;
		if (obj == null)
			return false;
		if (getClass() != obj.getClass())
			return false;
		Picture other = (Picture) obj;
		boolean result =
id.equals(other.id)&&timeStamp.equals(other.timeStamp)
				&&pictureInfo.equals(other.pictureInfo)

	&&tags.equals(other.tags)&&comment.equals(other.comment);
		if(result==false) return false;
		//이하는 stuffList 비교, 둘 중 한 쪽만 null 이거나, 개수가 다르면 false
		//else if(stuffList==null&&other.stuffList==null) return true;
		//else if(stuffList==null||other.stuffList==null) return false;
		else if(stuffList.size()!=other.stuffList.size()) return false;
		//하나하나 비교
		//other 의 중점에서 비교
		boolean flag=false;
		for(Stuff compareStuff:other.stuffList) {
			flag=false;
			for(Stuff savedStuff:stuffList)
				if(compareStuff.equals(savedStuff)) {
					flag=true;
					break;
				}
			if(flag==false) return false;
		}
		//this 의 중점에서 비교
		for(Stuff savedStuff:stuffList) {
			flag=false;
			for(Stuff compareStuff:other.stuffList)
				if(savedStuff.equals(compareStuff)) {
					flag=true;
					break;
				}
			if(flag==false) return false;
		}
		return true;
	}

	String getTimeStampToString() {
		return timeStamp.format(dateTimeFormat);
	}

	String getPictureFileName() {
		return pictureInfo.getImageFileName();
	}

	Image getPictureInfo() {
		return pictureInfo;
	}
```

```
	List<Stuff> getStuffList() {
		return stuffList;
	}

	Tag getTags() {
		return tags;
	}

	String getComment() {
		return comment;
	}
}
```

G. 이번에 수정한 PictureList.java 소스코드

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class PictureList {
	private List<Picture> pictureList;

	PictureList(){
		pictureList=new ArrayList<Picture>();
	}

	PictureList(String infoFileName){
		pictureList=new ArrayList<Picture>();
		savePictureByReadingFile(infoFileName);
	}

	PictureList(List<Picture> pictureList){
		this.pictureList=pictureList;
	}

	//중간에 추가할 수 있기에 메소드 분리, 다만 아직 쓸 일 없으니 private(프로그램
확장 시 변경)
	private void savePictureByReadingFile(String infoFileName) {
		File file=new File(infoFileName);
		Scanner input;
		try {
			input=new Scanner(file);
		} catch (FileNotFoundException e) {
			// TODO Auto-generated catch block
			System.out.println("Cannot find file :" + infoFileName);
			return;
		}
		Picture.setDateTimeFormat(DateTimeFormatter.ofPattern("yyyy-MM-
dd_HH:mm:ss:SSS"));
		while(input.hasNext()) {
			savePictureByString(input.nextLine());
		}
```

```
			input.close();
		}

		private void savePictureByString(String imageInfo) {
			String infoPicture=imageInfo.trim();
			if(infoPicture.startsWith("//")||infoPicture.isBlank())
				return;
			try {
			Picture picture=new Picture(infoPicture);
			addPicture(picture);
			} catch(IllegalStateException e) {
				System.out.println(e.getMessage());
				System.out.println("Failed to save picture "+ imageInfo);
			}
		}

		public void savePictureToFile(String infoFileName) {
			File file=new File(infoFileName);
			PrintWriter output;
			try {
				output = new PrintWriter(file);
			} catch (FileNotFoundException e) {
				// TODO Auto-generated catch block
				System.out.println("Cannot find file :" + infoFileName);
				return;
			}
			Picture.setDateTimeFormat(DateTimeFormatter.ofPattern("yyyy-MM-
dd_HH:mm:ss:SSS"));
			for(Picture picture:pictureList) {
				output.println(picture);
			}
			output.close();
		}

		public int size() {
			return this.pictureList.size();
		}

		public Picture get(int i) {
			return pictureList.get(i); //0 을 포함하는 것인가? (autotest 에서 -1
넣었다 에러 나옴)
		}

		public void add(Picture pic) {
			addPicture(pic);
		}

		private void addPicture(Picture picture) {
			for(Picture savedPicture:pictureList) { //이미 존재하는 id 에 대한
처리를 autotest 에서 요구하여 추가
				if(picture.equalById(savedPicture))
					throw new IllegalStateException("Picture with same
id already Exist");
			}
			pictureList.add(picture);
		}
```

```
	private void deletePicture(Picture picture) {
		pictureList.remove(picture);
	}
	/* 추가, 수정, 삭제, 검색은 이번에 요구하지 않았으므로 구현 없음
	private PictureList selectTags(Tag tag) {

	}

	private PictureList selectStuff(Stuff stuff) {

	}

	private void sortById() {
		throw new UnsupportedOperationException();
	}
	*/
	private void sortByDate() {
		pictureList.sort(null);
	}

	@Override
	public String toString() {
		String result="";
		for(Picture picture:pictureList)
			result+=picture+"\n";
		return result.trim();
	}

	List<Picture> getPictureList() {
		return pictureList;
	}
}
```

H. 이번에 수정한 Stuff.java 소스코드

```
public class Stuff implements Comparable{
	private String id;
	private String type;
	private String name;
	private Tag tags;

	Stuff(){
		//empty stuff
	}

	Stuff(String stuff){
		stuff=stuff.trim();
		if(stuff.startsWith("[")&&stuff.endsWith("]")) { //괄호가 섞여
들어오면 제거
			stuff=stuff.substring(1, stuff.length()-1);
		}
		String stuffs[]=(stuff+" ").split(";");
		if(stuffs.length!=4) {
			throw new IllegalStateException
```

```
                        ("Stuff \""+ stuff +"\" has something wrong with number of
\";\".");
                }
                id=null;
                type=stuffs[1].trim();
                name=stuffs[2].trim();
                if(type.isBlank()&&name.isBlank()) {
                        throw new IllegalStateException
                        ("both type and name of Stuff \""+ stuff +"\" cannot be
empty");
                }
                tags=new Tag(stuffs[3].trim());
        }

        Stuff(String type, String name, String tags) {
                this.id = null;
                this.type = type;
                this.name = name;
                this.tags = new Tag(tags);
                if(type.isBlank()&&id.isBlank()) {
                        throw new IllegalStateException("both type and name of Stuff
cannot be empty");
                }
        }

        Stuff(String id, String type, String name, Tag tags) {
                this.id = id;
                this.type = type;
                this.name = name;
                this.tags = tags;
                if(type.isBlank()&&id.isBlank()) {
                        throw new IllegalStateException("both type and name of Stuff
cannot be empty");
                }
        }

        boolean equalByTypeAndName(Stuff stuff) {
                if(stuff.type.equals(type)&&stuff.name.equals(name)) {
                        return true;
                }
                return false;
        }

        boolean equalExceptId(Stuff stuff) {

        if(stuff.type.equals(type)&&stuff.name.equals(name)&&stuff.tags.equals(tag
s)) {
                        return true;
                }
                return false;
        }

        @Override
        public int compareTo(Object o) {
                if(!(o instanceof Stuff)) throw new IllegalStateException(o+" is
not Stuff");
```

```
		Stuff stuff=(Stuff) o;
		int result = id.compareTo(stuff.id);
		if(result==0) {
			result = tags.compareTo(stuff.tags);
		}
		return result;
	}

	@Override
	public String toString() {
		return "[" + id + "; " + type + "; " + name + "; " + tags + "]";
	}

	@Override
	public boolean equals(Object obj) {
		if (this == obj)
			return true;
		if (obj == null)
			return false;
		if (getClass() != obj.getClass())
			return false;
		Stuff other = (Stuff) obj;
		return id.equalsIgnoreCase(other.id);
	}

	void setId(String id) {
		if(this.id==null)
			this.id = id;
	}

	void setId(Stuff stuff) {
		if(this.id==null)
			this.id = stuff.id;
	}

	String getType() {
		return type;
	}

	String getName() {
		return name;
	}

	Tag getTags() {
		return tags;
	}
}
```

I. 이번에 수정한 StuffList.java 소스코드

```
import java.util.ArrayList;
import java.util.List;

public class StuffList {
	private static List<Stuff> stuffList=new ArrayList<Stuff>();
	private static int stuffId=1;
```

```
	static List<Stuff> newStuffs(String stuff) {
		String[] stuffListByString=stuff.split("]");
		if(stuffListByString[stuffListByString.length-1].isBlank()) {
			String[] tempStuffList=new String[stuffListByString.length-
1];
			System.arraycopy(stuffListByString, 0, tempStuffList, 0,
tempStuffList.length);
			stuffListByString=tempStuffList;
		}

		List<Stuff> resultStuffList=new ArrayList<Stuff>();
		for(int i=0;i<stuffListByString.length;i++) {
			stuffListByString[i]=stuffListByString[i].trim();
			if(stuffListByString[i].startsWith("[")) {

	stuffListByString[i]=stuffListByString[i].substring(1);
				resultStuffList.add(addStuff(stuffListByString[i]));
			}
			else {
				throw new IllegalStateException
				("Stuffs \""+ stuff +"\" has something wrong with
\"[\" and \"]\".");
			}
		}
		return resultStuffList;
	}

	private static Stuff ifStuffExistByNameAndType(Stuff stuffToCompare) {
		for(Stuff stuff: stuffList) {
			if(stuffToCompare.equalByTypeAndName(stuff))
				return stuff;
		}
		return null;
	}

	private static Stuff ifStuffExistExceptId(Stuff stuffToCompare) {
		for(Stuff stuff: stuffList) {
			if(stuffToCompare.equalExceptId(stuff))
				return stuff;
		}
		return null;
	}

	//Id 할당은 필요함
	static Stuff addStuff(String stuffInfo) {
		stuffInfo=stuffInfo.trim();
		if(stuffInfo.startsWith("[")&&stuffInfo.endsWith("]")) {
			stuffInfo=stuffInfo.substring(1,stuffInfo.length()-
1).trim();
		}
		Stuff newStuff=new Stuff(stuffInfo);
		Stuff existingStuff=ifStuffExistByNameAndType(newStuff);
		if(existingStuff!=null) {
			Stuff existingSameStuff=ifStuffExistExceptId(newStuff);
			if(existingSameStuff!=null) {
				newStuff=existingSameStuff;
```

```
                        //전체출력시 같은 거 여러 번 나오는 불편한 상황 방지
                } else {
                        newStuff.setId(existingStuff);
                        stuffList.add(newStuff);
                }
        } else {
                newStuff.setId(String.format("%08d", stuffId));
                //00000001 부터 차례대로 ID 를 주기 위한 유일한 방법
                stuffId++;
                stuffList.add(newStuff);
        }
        return newStuff;
}

private static void sortById() {
        stuffList.sort(null);
}

public static void print() {
        System.out.println(allStuffToString());
}

private static String allStuffToString() {
        String stuffListByString="All Stuffs now saved: < ";
        for(Stuff stuff: stuffList) {
                stuffListByString+=stuff+" ";
        }
        return stuffListByString+">";
}
}
```

J. 이번에 수정한 Tag.java 소스코드

```
import java.util.ArrayList;
import java.util.List;

public class Tag implements Comparable{
        private List<String> tags;

        Tag(String tagInfo){
                tags=new ArrayList<String>();
                String[] splitResult=tagInfo.replace(" ", "").split("#");
                //중간의 공백도 삭제하기 위해 trim 이 아닌 replace 사용
                //바로 넣지 않는 이유는 맨 앞에 빈 문자열이 생기기 때문
                for(String tag:splitResult) {
                        if(!tag.isBlank())
                                addTag(tag);
                }
        }

        boolean ifTagExist(String tagToCompare) {
                tagToCompare=tagToCompare.replace("#", "").trim();
                for(String tag:tags) {
                        if(tagToCompare.equalsIgnoreCase(tag))
                                return true;
                }
```

```
			return false;
		}

		boolean ifTagExist(Tag tagToCompare) {
			boolean flag=false;
			for(String compareTag:tagToCompare.tags) {
				flag=false;
				for(String savedTag:tags)
					if(compareTag.equalsIgnoreCase(savedTag)) {
						flag=true;
						break;
					}
				if(flag==false) return false;
			}
			return true;
		}

		int getLength() {
			return tags.size();
		}

		void addTag(String tag) {
			tags.add(tag);
		}

		void deleteTag(String tag) {
			tag=tag.trim();
			if(tag.startsWith("#"))
				tag=tag.substring(1).trim();
			tags.remove(tag);
		}

		@Override
		public int compareTo(Object o) {
			if(!(o instanceof Tag)) throw new IllegalStateException(o+" is not
Tag");
			Tag tag=(Tag) o;
			int result=this.getLength()-tag.getLength();
			for(int i=0;result==0&&i<this.getLength();i++) {
				result=this.tags.get(i).compareTo(tag.tags.get(i));
			} //길이가 다르면 for 문 진입조차 하지 않음
			return result;
		}

		@Override
		public String toString() {
			String result="";
			for(String tag:tags) {
				result+="#"+tag+" ";
			}
			return result.trim();
		}

		//이하는 Eclipse 의 자동 생성
		@Override
		public boolean equals(Object obj) {
```

```
		if (this == obj)
			return true;
		if (obj == null)
			return false;
		if (getClass() != obj.getClass())
			return false;
		Tag other = (Tag) obj;
		//하나하나 비교
		//other 의 중점에서 비교
		boolean flag=false;
		for(String compareTag:other.tags) {
			flag=false;
			for(String savedTag:this.tags)
				if(compareTag.equals(savedTag)) {
					flag=true;
					break;
				}
			if(flag==false) return false;
		}
		//this 의 중점에서 비교
		for(String savedTag:this.tags) {
			flag=false;
			for(String compareTag:other.tags)
				if(savedTag.equals(compareTag)) {
					flag=true;
					break;
				}
			if(flag==false) return false;
		}
		return true;
	}
}
```

K. 이번에 수정한 Image.java 소스코드

```
public class Image {
	private String imageId;
	private String imageFileName;
	private String imageName;
	private Tag tags;

	Image(String imageInfo){
		String[] imageByString=(imageInfo+" ").split(";");
		if(imageByString.length!=4) {
			throw new IllegalStateException
			("Image \""+ imageInfo +"\" has something wrong with
\";\".");
		}
		imageId=imageByString[0].trim();
		imageFileName=imageByString[1].trim();
		imageName=imageByString[2].trim();
		tags=new Tag(imageByString[3]);
	}

	@Override
	public String toString() {
```

```
                return imageId + "; " + imageFileName + "; " + imageName + "; " +
tags;
        }

        @Override
        public boolean equals(Object obj) {
                if (this == obj)
                        return true;
                if (obj == null)
                        return false;
                if (getClass() != obj.getClass())
                        return false;
                Image other = (Image) obj;
                return
imageId.equals(other.imageId)&&imageFileName.equals(other.imageFileName)

        &&imageName.equals(other.imageName)&&tags.equals(other.tags);
        }

        String getImageFileName() {
                return imageFileName;
        }
}
```

평가표 (즉, 리포트 작성시 체크 사항)

평가 항목	학생 자체 평가 (리포트 해당 부분 표시 및 간단한 의견)	평가 (빈칸)	점수 (빈칸)
완성도 (동작 여부) - GUI 화면 완성도 - "ADD" 클릭하면, picture 정보 입력하는 새로운 창이 떠야 함. - "SEARCH" 클릭하면, picture 정보 입력하는 새로운 창이 떠야 함.	GUI 화면 완성도: 완벽하지는 않으나 비슷하게 완성 ADD 버튼 event: 완료 SEARCH 버튼 event: 완료		
설계 노트 - GUI 설계 시 착안 사항 - 클래스 구조 - 프로젝트 3의 클래스와의 관계 - 시행착오 내용 및 해결 방안	GUI 설계 시 착안 사항: Layout 설정은 중간중간 설명, 클래스 분리 기준은 가장 앞 설명(자주 쓸 만한 것 위주) 클래스 구조: event는 별도로 빼지 않음, Panel은 사진과 Stuff 2개(Stuff도 정보를 담은 것과 빈 것이 같은 클래스에 따로 있음) 프로젝트 3과의 관계: 연동함(PictureList에 저장 후 그 정보를 가져와 그림) 시행착오: 중간중간 작성(창 크기, StuffPanel 등등)		
리포트 - 평가자 시각으로 리포트 검토 - 위의 평가 요소들이 명확하게 기술되었는가? - 기타 건의 사항	평가자 시각으로 검토: 보고서 설명이 복잡하기는 하나 과제 설명서와 같은 이유일 뿐, 읽기는 어렵지 않아 보임 평가 요소 기술: "착안 사항"이라는 게 내가 이해한 게 맞으면 문제없음		
총평/계			

* 학생 자체 평가는 점수에 반영되지 않음.
* 학생 스스로 자신의 보고서를 평가하면서, 체계적으로 프로젝트를 마무리하도록 유도하는 것이 목적임.