VideoMetadataEditor (1.11)

Kcits970

February 6, 2022

Sections

Section 1. Description

The purpose of the program ‘VideoMetadataEditor’ is to simulate an online video database. It provides an event-driven interface to add, delete, edit, load, and save videos.

It should be noted that this program does not user real videos. Any object that can contain some fields of metadata is enough to satisfy my purpose.

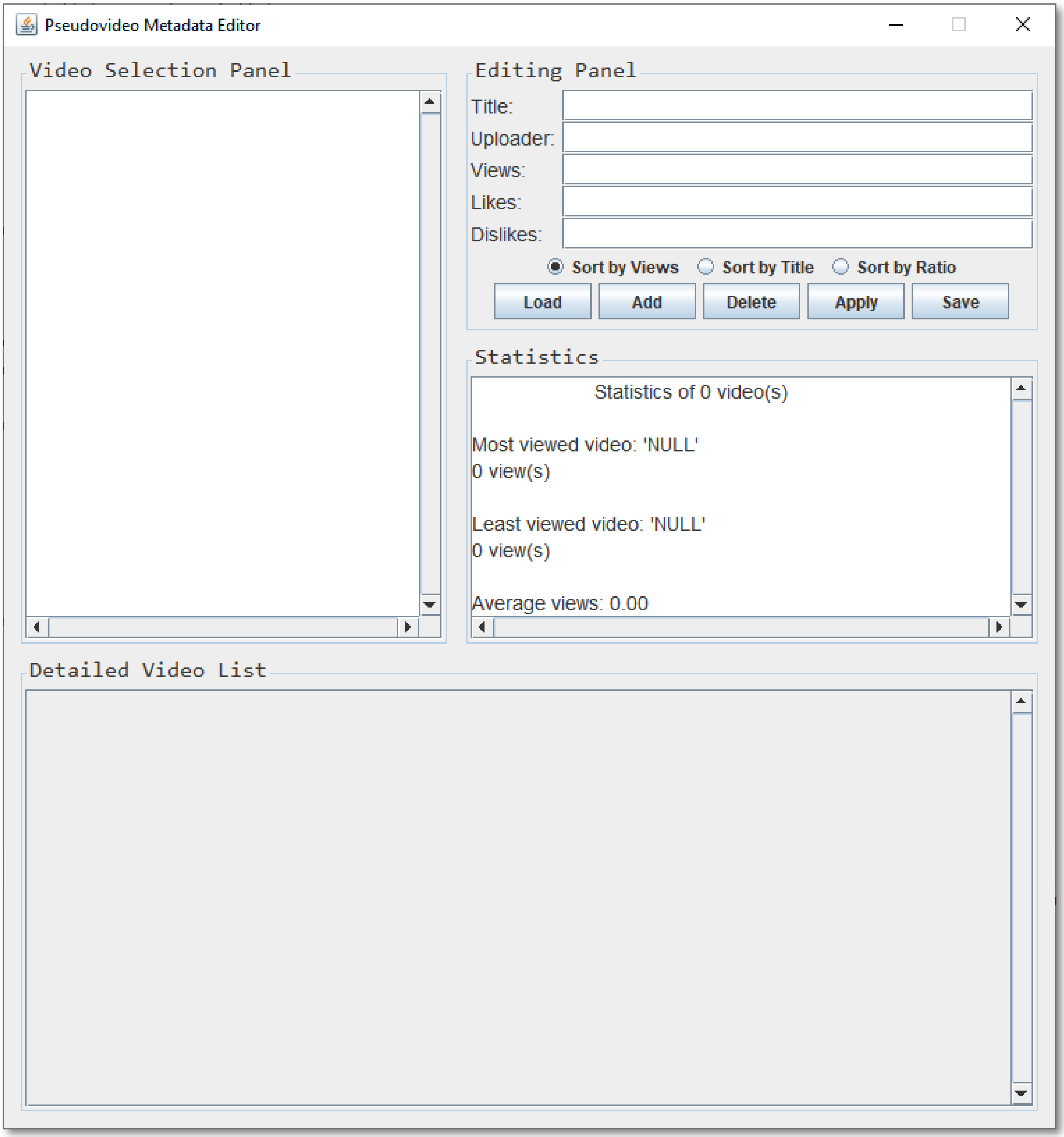
<Video>

```
String title
String uploader
int views
int likes
int dislikes
```

The program is written in Java language, and compiled using Java Compiler 15.0.1. Attempts to compile the given source code with a different compiler may result in unexpected or undefined behavior.

Section 2. Features

When the program is launched, an empty frame will be shown. From here, the user can perform various actions such as adding new videos, deleting existing videos, writing to an external file, etc.



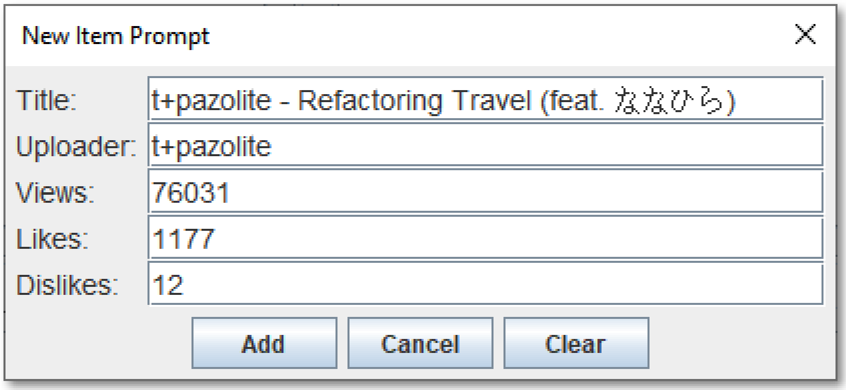
● Sort by Views ○ Sort by Title ○ Sort by Ratio

```
Statistics of 0 video(s)

Most viewed video: 'NULL'
0 view(s)

Least viewed video: 'NULL'
0 view(s)

Average views: 0.00
```

(The initial frame at program launch.)

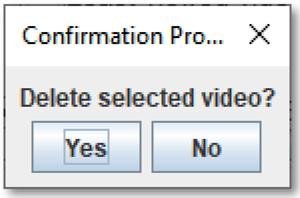
Adding/Deleting Videos

To add a new video to the list, click the [Add] button in the [Editing Panel]. A prompt containing 5 textfields will appear. Simply enter the video metadata[1] into the fields and click the [Add] button.


New Item Prompt

Title: t+pazolite - Refactoring Travel (feat. ななひら)
Uploader: t+pazolite
Views: 76031
Likes: 1177
Dislikes: 12

As opposed to adding, the video can also be removed from the list. To delete an existing video, select the desired video from the [Video Selection Panel], and click the [Delete] button. A prompt asking for confirmation will appear. Click the [Yes] button to confirm the deletion.


Confirmation Pro...

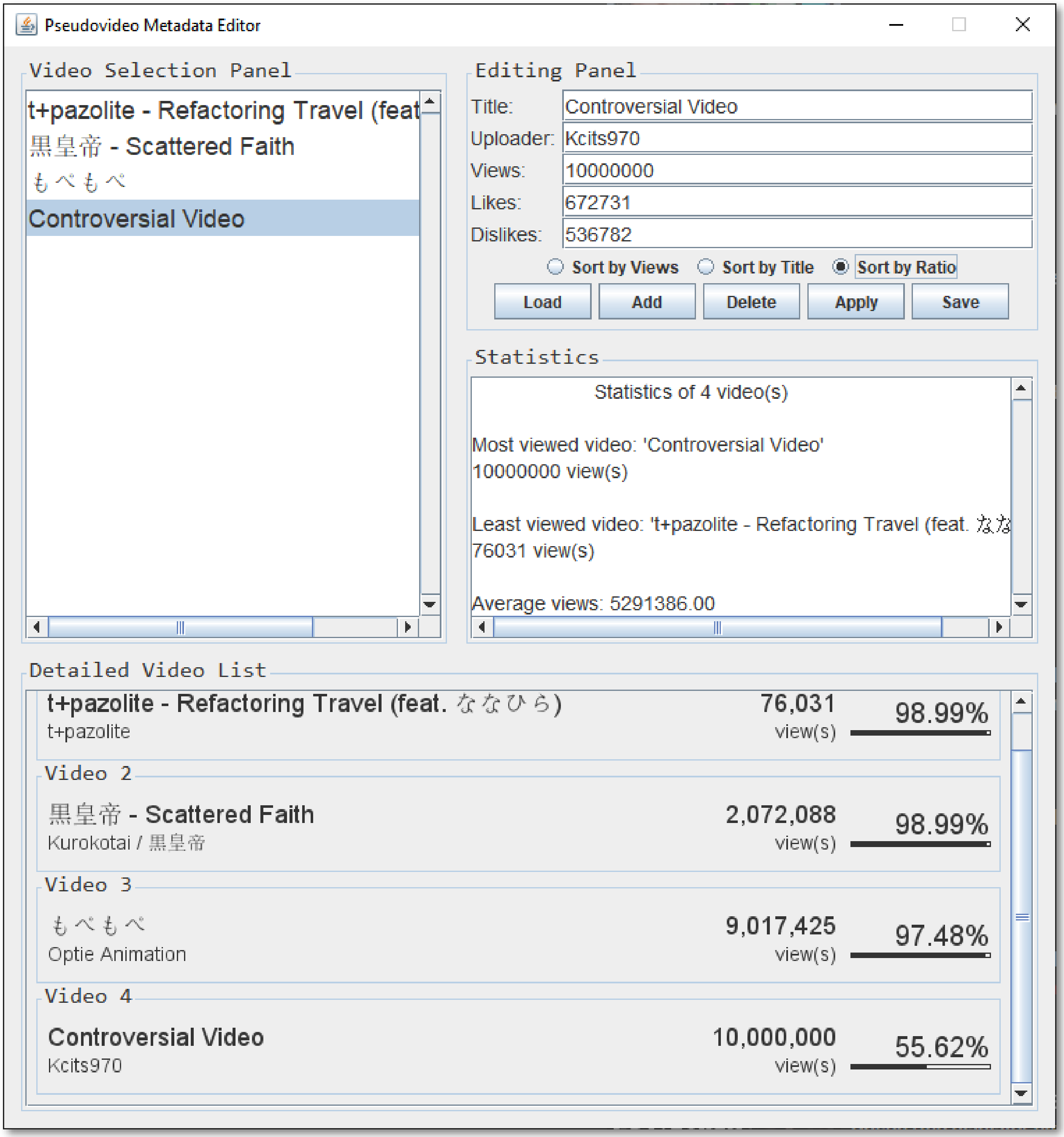
Delete selected video?

Editing/Sorting Videos

The process of editing or sorting is quite straightforward. To edit a video, select any video from the [Video Selection Panel] and modify the values of the textfields in the [Editing Panel]. The changes will only apply if the user clicks the [Apply] button. If any other action is performed without applying changes, the entered information will be lost.

[1] The metadata doesn't necessarily have to be from a real video. This program is for simulation purposes only, so any inputs are completely fine.

The user may also want to sort the list in a different order. Simply click any one of the radio buttons to sort the list in a specific order.



t+pazolite - Refactoring Travel (feat
黒皇帝 - Scattered Faith
もぺもぺ
Controversial Video

Title: Controversial Video
Uploader: Kcits970
Views: 10000000
Likes: 672731
Dislikes: 536782

○ Sort by Views ○ Sort by Title ● Sort by Ratio

Statistics of 4 video(s)

Most viewed video: 'Controversial Video'
10000000 view(s)

Least viewed video: 't+pazolite - Refactoring Travel (feat. なな
76031 view(s)

Average views: 5291386.00

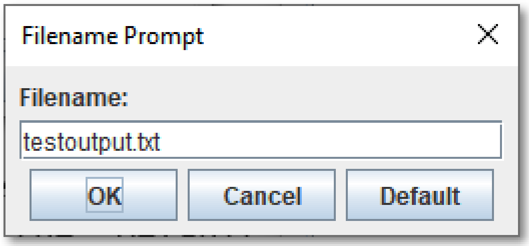
t+pazolite - Refactoring Travel (feat. ななひら) t+pazolite	76,031 view(s)	98.99%
Video 2		
黒皇帝 - Scattered Faith Kurokotai / 黒皇帝	2,072,088 view(s)	98.99%
Video 3		
もぺもぺ Optie Animation	9,017,425 view(s)	97.48%
Video 4		
Controversial Video Kcits970	10,000,000 view(s)	55.62%

(The frame after adding 4 videos and sorting them by ratio[2].)

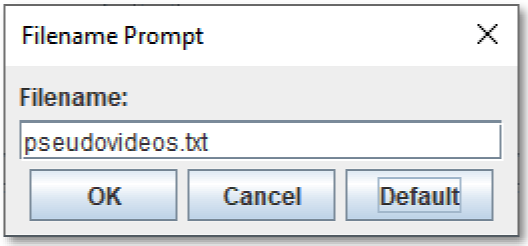
[2] ‘Ratio’ refers to the like/dislike ratio in terms of likes.

Saving/Loading Videos

The program also supports loading and saving videos to an external file. To save the current status, click the [Save] button in the [Editing Panel], which the program responds with a filename prompt. From here, the user can enter a desired filename, or use the default filename by clicking the [Default] button[3].



Filename: testoutput.txt

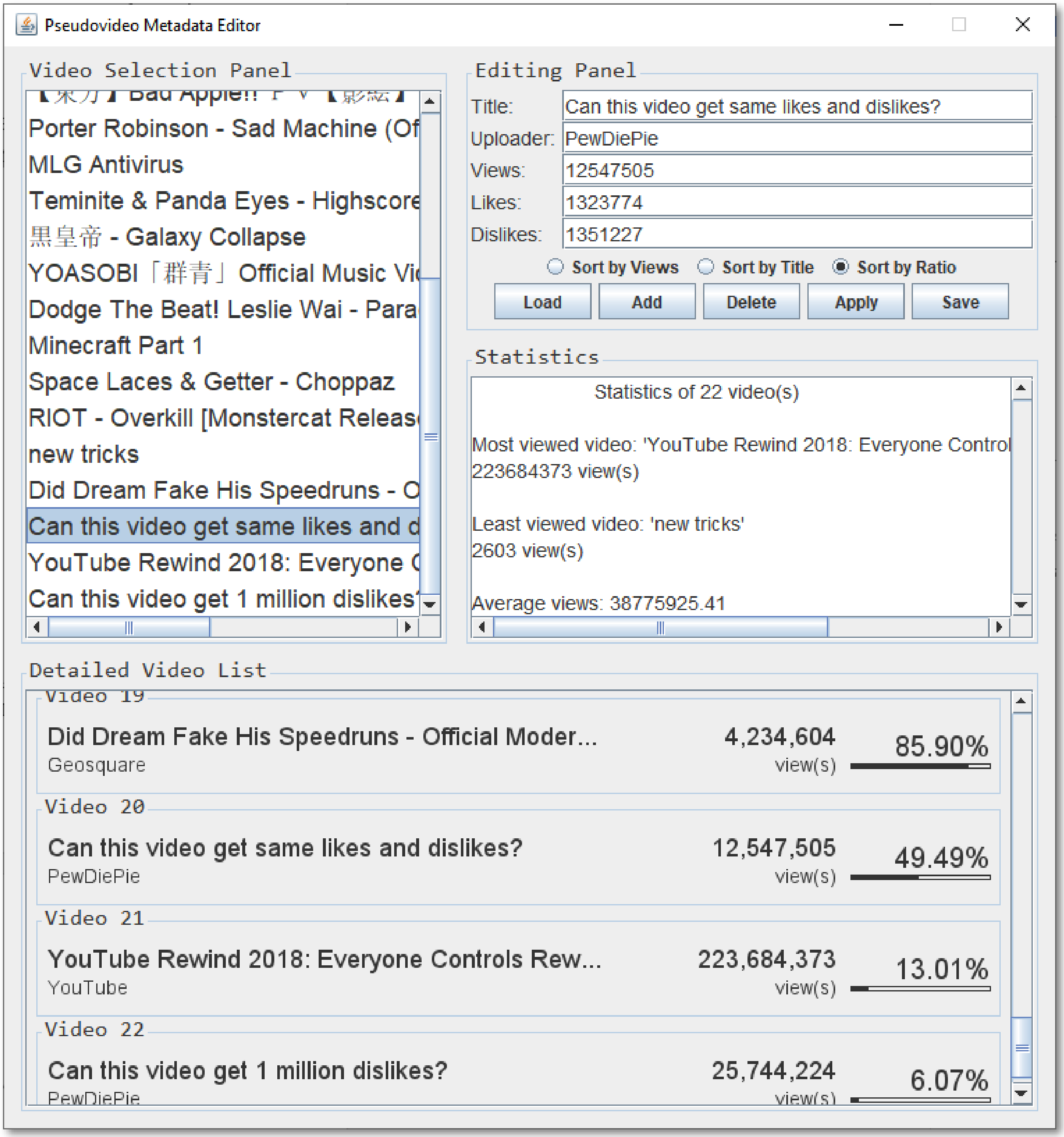


Filename: pseudovideos.txt

Click the [OK] button to save the file. If successfully saved, the file with the inputted name will appear in the directory of launch.

Loading videos from a file is very similar to saving. Simply click the [Load] button, and follow the same process. If the file-IO succeeds, the list should update with the contents of the specified file.

[3] Clicking the [Default] button will simply put "videos.xml" into the textfield. It does not automatically close the prompt and save to the default location.



【東方】Bad Apple!! ＰＶ【影絵】
Porter Robinson - Sad Machine (Of
MLG Antivirus
Teminite & Panda Eyes - Highscore
黒皇帝 - Galaxy Collapse
YOASOBI「群青」Official Music Vi
Dodge The Beat! Leslie Wai - Para
Minecraft Part 1
Space Laces & Getter - Choppaz
RIOT - Overkill [Monstercat Releas
new tricks
Did Dream Fake His Speedruns - O
Can this video get same likes and d
YouTube Rewind 2018: Everyone C
Can this video get 1 million dislikes

Title: Can this video get same likes and dislikes?
Uploader: PewDiePie
Views: 12547505
Likes: 1323774
Dislikes: 1351227

○ Sort by Views ○ Sort by Title ● Sort by Ratio

Statistics of 22 video(s)

Most viewed video: 'YouTube Rewind 2018: Everyone Control
223684373 view(s)

Least viewed video: 'new tricks'
2603 view(s)

Average views: 38775925.41

Video 19
Did Dream Fake His Speedruns - Official Moder... 4,234,604 view(s) 85.90%
Geosquare

Video 20
Can this video get same likes and dislikes? 12,547,505 view(s) 49.49%
PewDiePie

Video 21
YouTube Rewind 2018: Everyone Controls Rew... 223,684,373 view(s) 13.01%
YouTube

Video 22
Can this video get 1 million dislikes? 25,744,224 view(s) 6.07%
PewDiePie

(The frame after loading videos from the default location and sorting them by ratio.)

Section 3. File Format

The files that are used to save/load videos follow the XML syntax. Here’s an example of one.

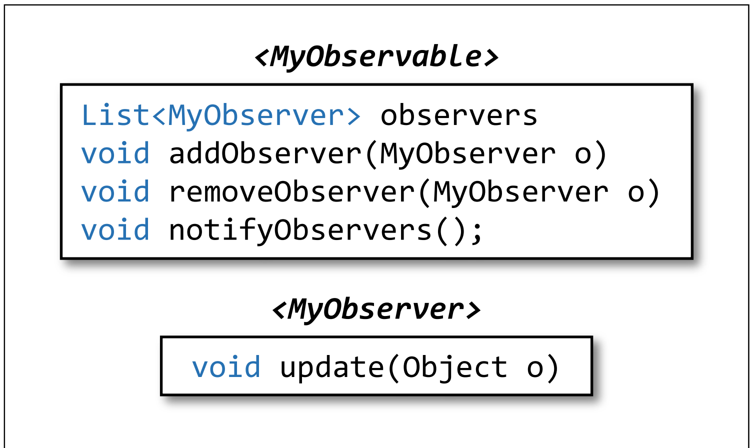
```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<videos>
    <video>
        <title>Koxx - A Fool Moon Night</title>
        <uploader>Everoid</uploader>
        <views>3841426</views>
        <likes>73616</likes>
        <dislikes>542</dislikes>
    </video>
    <video>
        <title>Space Laces &amp; Getter - Choppaz</title>
        <uploader>UKF Dubstep</uploader>
        <views>502969</views>
        <likes>12228</likes>
        <dislikes>248</dislikes>
    </video>
    <video>
        <title>Dodge The Beat! Leslie Wai - Paradigm [a void and ESC ape] 0.00%
1x100</title>
        <uploader>R eg</uploader>
        <views>100470</views>
        <likes>2247</likes>
        <dislikes>38</dislikes>
    </video>
</videos>
```

The elements should be self-explanatory, but I will leave a brief explanation of each element below.

videos	The root element of the tree.
video	Indicates a video object.
title	The title field of a video.
uploader	The uploader field of a video.
views	The number of views of a video.
likes	The number of likes of a video.
dislikes	The number of dislikes of a video.

Section 4. Design Pattern

This program follows the observer pattern. I’ve implemented two classes to follow this design pattern[4].



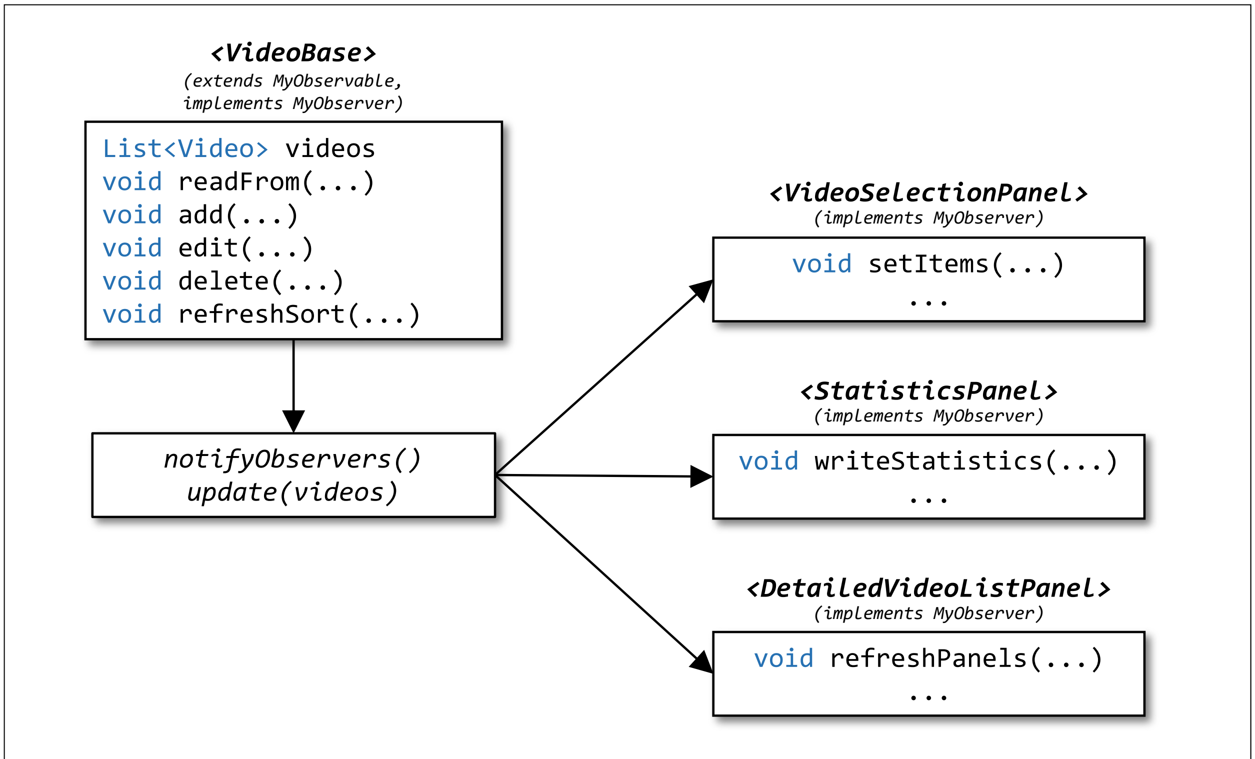
```
List<MyObserver> observers
void addObserver(MyObserver o)
void removeObserver(MyObserver o)
void notifyObservers();
```

```
void update(Object o)
```

Observer Classes

In this program, the frame classes need to refresh themselves every time VideoBase changes. The changes include: adding/deleting videos, changing the sorting order of videos, and changing the metadata of a video. The diagram in the next page illustrates how this pattern applies to the program.

[4] Since the release of Java 9, the observer classes from java.util have been deprecated. To go around this issue, I simply rewrote the classes on my own. MyObservable replaces java.util.Observable, and MyObserver replaces java.util.Observer.



Observable-Observer Relation Diagram

The frame classes[5] are registered as the observer of VideoBase. The call to any method that modify the contents of the base consequently invokes notifyObservers(), which allows the frame classes to correctly display the latest data.

[5] VideoSelectionPanel, StatisticsPanel, DetailedVideoListPanel