2021

Movie Management System

GROUP PROJECT

TEAM 1

ADAM JOST | NEHA METLAPALLI

Movie Management System Group Project

Objective

The objective of the movie management system group project was to develop a menu-based Java program that maintains two lists of movies, "received" and "released", and allows the user to enter commands to perform operations to the lists and the movies contained within those lists.

Our Team

Our team consists of two Java developers:

- Adam Jost
- Neha Metlapalli

Team Member Contributions

Project report:

- Adam Jost
- Neha Metlapalli

UML Diagram:

- Adam Jost

Source Code:

- ManageMentSystem.java class was coded by Adam Jost
- MovieList.java class was coded by Neha Metlapalli
- ComparableType.java interface was coded by Adam Jost
- MovieStatus.java Enum was coded by Adam Jost
- Movie.java class was coded by Adam Jost

Debugging:

- Adam Jost
- Neha Metlapalli

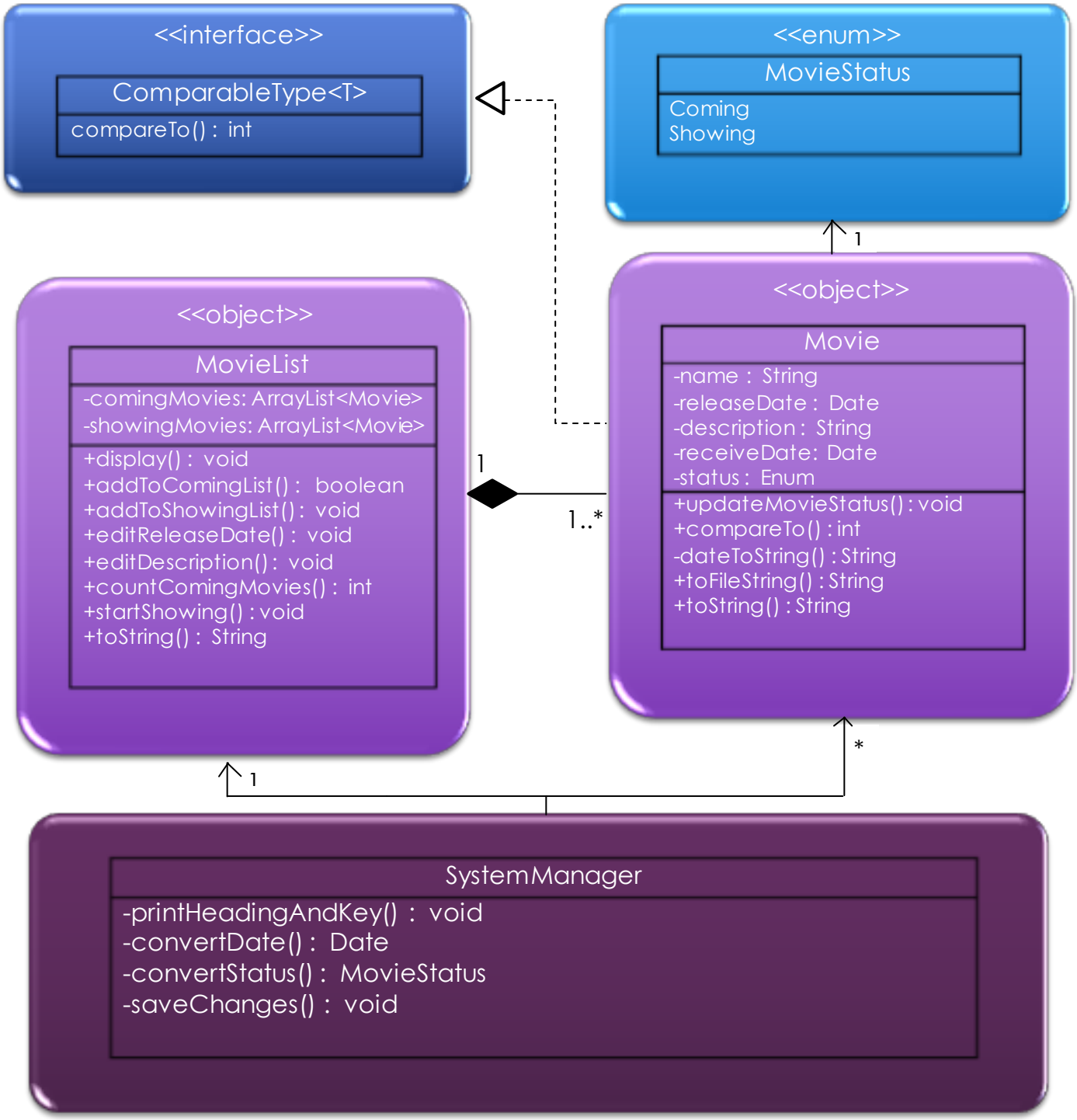
System Functionality Overview

Our team developed a system to read in and store data from an input file containing a list of movies and all of their data. A Movie object is created for each individual movie and is then stored into one of two lists, received movies or released movies. When populating the movie lists, the system automatically updates the status of all movies with a release date of today's date or prior to "released". The system then prints a simple console interface consisting of a heading and a command key which provides the user with all available commands at their disposal. The operations that the user can perform includes displaying the command key, displaying the movies from both of the movie lists, editing a movie's release date, editing a movie's description, changing a movie's status from "received" to "released", counting the number of movie's prior to a given date whose movie status is "received", canceling the current command in progress, saving all changes made, and exiting the program. The system gracefully handles invalid input from the user and responds with success or error messages after the completion of all operations that are performed.

The Interface

```
=======================================================
                MOVIE MANAGEMENT SYSTEM
=======================================================
                      Command Key
-------------------------------------------------------
KEY - Display the command key
DISPLAY - Display all movies
ADD - Add a movie to the received movies list
EDIT - Edit a movie's release date or description
START SHOWING - Start showing movies on a given date
COUNT - Number of received movies prior to a date
CANCEL - Cancels the current operation
SAVE - Save all changes
EXIT - Exit the program
=======================================================
          Type any above command to continue
=======================================================
```

UML Class Diagram



System Design

Physical Design

Physical design relates to the actual input and output processes of the system such as how data is entered into a system, verified, processed, and displayed.

Input Design

File Input

At the start of the program, a one-time operation occurs where the system reads in a source document (.txt file) which contains a list of movies and their data in the following format:

```
name, release date, description, receive date, status
```

All of the above-mentioned data is initially accepted line-by-line as String values and then parsed accordingly before instantiating Movie objects.

Interface for Accepting User Input

The user interface is a simple command line interface and is constructed using a textual-based design utilizing "=" and "-" symbols as separators which is printed to the standard console. This text-based, menu-driven system provides the user with a pleasing appearance and precise instructions of how to proceed and accomplish the operations they may wish to perform.

User Input

The user input design consists of precise input instructions, which are easy, logical, and easy to follow. The system then evaluates the user input data and depending on whether it's in the expected format or not, the process either goes to the next step or rejects the input by outputting an error response and then once again requests the required information. If an operation is completed

successfully using only valid input the user will receive a success message signifying that the operation has been completed successfully.

Valid User Input

- Commands: Can be any combination of capital and non-capital letters of any of the acknowledged commands.
- Movie name: Must be a non-empty string.
- Date: Must be entered in the format 01/01/2021.
- Description: Must be a non-empty string.
- Status: Can be any combination of capital and non-capitol letters of the words "received" or "released".

Architectural Design

Architectural design focuses on the design of system architecture. It describes the structure and behavior of the system and defines the structure and relationship between various parts of the systems.

Major Classes

There is a total of three major classes:

- *SystemManager*
- *MovieList*
- *Movie*

Relationship Between the Major Classes

- *SystemManager* creates a *MovieList* object.
- *SystemManager* creates many *Movie* objects.
- *SystemManager* can modify a MovieList.
- *MovieList* stores 0…* *Movie* objects in one of two ArrayLists data structures.
- A *MovieList* can modify a *Movie* object's *description* or *releaseDate*.

- *SystemManager* can display the *MovieList* object.

Class and Interface Details

MovieList

Stores Movie objects in ArrayLists and performs operations to the lists and the Movie objects contained within those lists. Calls methods from the Movie class.

- Attributes:
  - *receivedMovies*: ArrayList<Movie>
  - *releasedMovies*: ArrayList<Movie>
- Created by SystemManager
- Data Structures:
  - ArrayList<Movie> receivedMovies
    - Contains *Movie* objects with the status of *RECEIVED*.
  - ArrayList<Movie> releasedMovies
    - Contains *Movie* objects with the status of *RELEASED*.
  - ArrayList<Integer> *removePos*
    - Contains integers that are the index positions of elements to be removed from the *receivedMovies* list in the *startShowing()* method.

Movie

- Attributes:
  - *name*: String
  - *releaseDate*: Date
  - *description*: String
  - *receiveDate*: Date
  - *status:* MovieStatus
- Created by the *SystemManager*

- Stored in a *MovieList* object
- Implements the *ComparableType* interface

ComparableType <<Interface>>

- Interface implemented by the Movie class.
- Includes the compareTo() method which allows for comparison of Movie objects by their *releaseDate* attribute.

MovieStatus <<enum>>

- *RECEIVED* or *RELEASED*.
- Data field of the *Movie* class.

Test Cases

We will be using the following input file for both test cases:

```
1 The Departed, 07/01/2021, Drama/Mystery, 06/28/2021, RECEIVED
2 The Tomorrow War, 07/02/2021, Sci-Fi/Action, 07/01/2021, RECEIVED
3 Let Us In, 07/02/2021, Sci-Fi/Drama, 07/01/2021, RECEIVED
4 Lawless, 07/10/2021, Drama, 07/05/2021, RECEIVED
5 Hacker, 07/12/2021, Action/Drama/Suspense, 07/10/2021, RECEIVED
```

Test Case 1

Expected output

KEY: This command should print the heading and the command key containing a list of all commands that the user can enter to the console.

DISPLAY: This command should print the coming and showing movies to the console showing the state of the movie lists before the following operations are performed. The received movies should be ordered by release date.

ADD: Using the inputs "Avatar", "05/05/2021", "Sci-Fi", "04/10/2021", and "Received", this command should add a movie with the name "Avatar", release date of May 5, 2021, description of "Sci-Fi", and receive date of April 10, 2021, with status of "Received."

EDIT: Using the inputs "Avatar" and "Action", this command should edit the description of the movie named "Avatar" to "Action".

START SHOWING: Using the input "05/05/2021", this command will move all movies with a release date of 05/05/ 2021 from the received list to the released list and update their status to "RELEASED".

COUNT: Using the input "01/01/2019" this command will count the number of received movies before January 1, 2019, which should be 0.

CANCEL: This command will cancel any operation that's already in process. For example, if the user entered CANCEL when the program asks the user to enter the date to count the received movies, the program will promptly end the count operation and prompt the user to enter a new command.

SAVE: This command should overwrite the movies.txt file to reflect all of the above changes made by the user. The movies.txt file should now contain the added Avatar movie with all of its data.

EXIT: This command should end and successfully exit the program.

Actual output

KEY:

As shown below, the program successfully reprinted the command key to the console as expected.

```
key
=====================================================
                MOVIE MANAGEMENT SYSTEM
=====================================================
                      Command Key
-----------------------------------------------------
KEY - Display the command key
DISPLAY - Display all movies
ADD - Add a movie to the received movies list
EDIT - Edit a movie's release date or description
START SHOWING - Start showing movies on a given date
COUNT - Number of received movies prior to a date
CANCEL - Cancels the current operation
SAVE - Save all changes
EXIT - Exit the program
=====================================================
          Type any above command to continue
=====================================================

Enter another command or "EXIT" to exit:
```

DISPLAY:

As shown below, the program successfully prints the movie list to the console as expected.

```
Enter another command or "EXIT" to exit:
DISPLAY
================
Received Movies:
================

The Departed
Genre: Drama/Mystery
Status: RECEIVED
Receive Date: 06/28/2021
Release Date: 07/01/2021


The Tomorrow War
Genre: Sci-Fi/Action
Status: RECEIVED
Receive Date: 07/01/2021
Release Date: 07/02/2021


Let Us In
Genre: Sci-Fi/Drama
Status: RECEIVED
Receive Date: 07/01/2021
Release Date: 07/02/2021
```

```
Lawless
Genre: Drama
Status: RECEIVED
Receive Date: 07/05/2021
Release Date: 07/10/2021


Hacker
Genre: Action/Drama/Suspense
Status: RECEIVED
Receive Date: 07/10/2021
Release Date: 07/12/2021

--------------------------------------------------------

================
Released Movies:
================
```

ADD:

```
Enter another command or "EXIT" to exit:
ADD
Enter the movie name:
Avatar
Enter the movie's release date (e.g. 01/22/2021):
05/05/2021
Enter the movie's description (e.g. Sci-Fi/Adventure):
Sci-Fi
Enter the movie's receive date (e.g. 01/22/2021):
04/10/2021
Enter the movie's status (e.g. "Received" or "Released"):
Received
Movie has been successfully added.
```

EDIT:

```
Enter another command or "EXIT" to exit:
EDIT
-----------------------------------------------------
                * Important Notice *
 Movies can only be edited if their status is "Coming"
-----------------------------------------------------

Enter the movie name:
Avatar
Edit "DESCRIPTION" or "RELEASE DATE"?
DESCRIPTION
Enter the movie's description (e.g. Sci-Fi/Adventure):
Action
Edit was successful.
```

START SHOWING:

```
Enter another command or "EXIT" to exit:
START SHOWING
Enter the release date of the movies you would like
to start showing (e.g. 01/22/2021):
05/05/2021
Avatar started showing successfully.
```

COUNT:

This command successfully returned the count of zero as expected.

```
Enter another command or "EXIT" to exit:
COUNT
Enter the date to count up to (e.g. 01/22/2021):
01/01/2019
Count of movie's coming before the given date is 0.
```

CANCEL:

This command successfully cancelled the current operation as expected.

```
Enter another command or "EXIT" to exit:
COUNT
Enter the date to count up to (e.g. 01/22/2021):
CANCEL

Enter another command or "EXIT" to exit:
```

SAVE:

The Avatar movie and all of the changes made to its data is now reflected in the movies.txt file as expected.

Command being entered with program's response.

```
Enter another command or "EXIT" to exit:
SAVE
Save completed successfully.
```

Movies.txt after the "SAVE" operation:

```
The Departed, 07/01/2021, Drama/Mystery, 06/28/2021, RECEIVED
The Tomorrow War, 07/02/2021, Sci-Fi/Action, 07/01/2021, RECEIVED
Let Us In, 07/02/2021, Sci-Fi/Drama, 07/01/2021, RECEIVED
Lawless, 07/10/2021, Drama, 07/05/2021, RECEIVED
Hacker, 07/12/2021, Action/Drama/Suspense, 07/10/2021, RECEIVED
Avatar, 05/05/2021, Action, 04/10/2021, RELEASED
```

EXIT: This command successfully stopped the program's execution as expected.

```
==========
exit
Application has been successfuly exited.
```

Test Case 2

Expected output

KEY: This command should print the heading and the command key containing a list of all commands that the user can enter to the console.

DISPLAY: This command should print all of the received and released movies to the console. This includes the "Avatar" movie we added in test case 1. The received movies will be ordered by release date.

ADD: For this operation, the user will add a movie with the name "Black Panther", release date of 05/07/2021, description of "Action", and receive date of 05/10/2021. However, this should show an error message that says that the receive date has to be before the release date. Then, the user will enter the receive date to be 05/01/ 2021, which should work. Finally, the user will enter the status to be "Received".

EDIT: This command should edit the release date of the movie named "Black Panther" to 06/20/2021.

START SHOWING: This operation will move all movies with release date of 06/21/2021 from the received list to the released list, but no such movies exist with that release date, so this operation won't move any new movies to the released list.

COUNT: This operation should count the number of received movies before 06/20/2021, which should be zero.

CANCEL: This command should cancel any operation that's already in process. For example, if the user entered CANCEL when the program asks the user to enter the date to count the coming movies, the program will promptly end the current count operation and prompt the user to enter a new command.

SAVE: This operation will overwrite whatever is currently in the movies.txt file to reflect the changes the user made, so it should add the "Black Panther" movie to the file.

EXIT: This command should promptly end the program.

Actual output

KEY:

As shown below, the program successfully reprinted the command key to the console as expected.

```
key
=======================================================
                MOVIE MANAGEMENT SYSTEM
=======================================================
                      Command Key
-------------------------------------------------------
KEY - Display the command key
DISPLAY - Display all movies
ADD - Add a movie to the received movies list
EDIT - Edit a movie's release date or description
START SHOWING - Start showing movies on a given date
COUNT - Number of received movies prior to a date
CANCEL - Cancels the current operation
SAVE - Save all changes
EXIT - Exit the program
=======================================================
          Type any above command to continue
=======================================================

Enter another command or "EXIT" to exit:
```

DISPLAY:

As shown below, the program successfully prints the movie list to the console as expected.

```
Enter another command or "EXIT" to exit:
DISPLAY

================
Received Movies:
================

The Departed
Genre: Drama/Mystery
Status: RECEIVED
Receive Date: 06/28/2021
Release Date: 07/01/2021


The Tomorrow War
Genre: Sci-Fi/Action
Status: RECEIVED
Receive Date: 07/01/2021
Release Date: 07/02/2021


Let Us In
Genre: Sci-Fi/Drama
Status: RECEIVED
Receive Date: 07/01/2021
Release Date: 07/02/2021
```

```
Lawless
Genre: Drama
Status: RECEIVED
Receive Date: 07/05/2021
Release Date: 07/10/2021


Hacker
Genre: Action/Drama/Suspense
Status: RECEIVED
Receive Date: 07/10/2021
Release Date: 07/12/2021

-----------------------------------------------------

================
Released Movies:
================

Avatar
Genre: Action
Status: RELEASED
Receive Date: 04/10/2021
Release Date: 05/05/2021
```

ADD:

```
Enter another command or "EXIT" to exit:
ADD
Enter the movie name:
Black Panther
Enter the movie's release date (e.g. 01/22/2021):
05/07/2021
Enter the movie's description (e.g. Sci-Fi/Adventure):
Action
Enter the movie's receive date (e.g. 01/22/2021):
05/10/2021
Error: Receive date must be before release date.
Enter the movie's receive date (e.g. 01/22/2021):
05/01/2021
Enter the movie's status (e.g. "Received" or "Released"):
Received
Movie has been successfully added.
```

EDIT:

```
Enter another command or "EXIT" to exit:
EDIT
-------------------------------------------------------
                * Important Notice *
 Movies can only be edited if their status is "Coming"
-------------------------------------------------------

Enter the movie name:
Black Panther
Edit "DESCRIPTION" or "RELEASE DATE"?
RELEASE DATE
Enter the movie's release date (e.g. 01/22/21):
06/20/2021
Edit was successful.
```

START SHOWING:

As expected, a message is returned informing the user that "No movies with that release date have the status of "Received".

```
Enter another command or "EXIT" to exit:
START SHOWING
Enter the release date of the movies you would like
to start showing (e.g. 01/22/2021):
06/21/2021
No movies with that release date have the status "Received".
```

COUNT:

As shown below, the program returned the expected count of zero.

```
Enter another command or "EXIT" to exit:
COUNT
Enter the date to count up to (e.g. 01/22/2021):
06/20/2021
Count of movie's coming before the given date is 0.
```

CANCEL:

As shown below, the program successfully canceled the in-progress operation as expected.

```
Enter another command or "EXIT" to exit:
COUNT
Enter the date to count up to (e.g. 01/22/2021):
CANCEL

Enter another command or "EXIT" to exit:
```

SAVE: The Black Panther movie and all of the changes made to its data is now reflected in the movies.txt file.

Command being entered with program's response.

```
Enter another command or "EXIT" to exit:
SAVE
Save completed successfully.
```

The movies.txt after the "SAVE" command has been entered:

```
Black Panther, 06/20/2021, Action, 05/01/2021, RECEIVED
The Departed, 07/01/2021, Drama/Mystery, 06/28/2021, RECEIVED
The Tomorrow War, 07/02/2021, Sci-Fi/Action, 07/01/2021, RECEIVED
Let Us In, 07/02/2021, Sci-Fi/Drama, 07/01/2021, RECEIVED
Lawless, 07/10/2021, Drama, 07/05/2021, RECEIVED
Hacker, 07/12/2021, Action/Drama/Suspense, 07/10/2021, RECEIVED
Avatar, 05/05/2021, Action, 04/10/2021, RELEASED
```

EXIT:

This command successfully stopped the program's execution as expected.

```
==========================================================
exit
Application has been successfuly exited.
```

Future Improvements

An improvement that could have been made would be to use *BufferedReader* when reading the input file because it is faster than using a *Scanner* object. Another improvement that could have been made was using binary search in the *addToReceivedList()* method to search for where to insert the input Movie object because binary search has a time complexity of $O(log(n))$, while linear search, which is what is used in the *addToReceivedList()* method, has a time complexity of $O(n)$. These two improvements would make the program run faster, especially for large amounts of input.