COMP 303 Winter 2021

Assignment 2

Belle Pan 260839939

5th February, 2021

`Episode` Class

- This class implements the abstract methods in interfaces `Watchable`
    - An `Episode` object may be watched and has elements of a `Watchable` object (validity, a title, a language, a publishing studio, and custom information)
- Field `aPath` represents the file path
    - `aPath` is declared as a `final private` field, means that its value will be immutable once it has been declared
        - This ensures that the client cannot access and change its value once it has been declared (i.e. this helps prevent information leaking)
- Fields `aTitle, aLanguage, aStudio, aTag, aSeqNum` represent the title, language, publishing studio, custom information, and sequential number respectively
    - Note that these fields are all `private` fields, meaning that their values will be only be accessible through their respective getter and setter methods
        - This ensures that the client cannot access and change their values directly, and that these values may be controlled by pre-existing methods that were designed (i.e. this helps prevent information leaking)
        - Their respective getter methods are `getTitle(), getLanguage(), getStudio(), getTag(String pKey)`
        - There is only one setter method provided : `setTag(StringpKey, String pValue)`
        - It is unlikely that the title, publishing studio and language of an episode would be modified, but perhaps it would be better to provide the functionality to the client regardless to prevent massive changes in the future
- Field `aName` uses a `String` to represent the name of the watchlist
    - This is accessible using `getName()` and may be changed using `changeName()`

`TVShow` Class

- This class implements the abstract methods in interfaces `Watchable` and `Bingeable<Episode>`
    - A `TVShow` object may be watched and has elements of a `Watchable` object (validity, a title, a language, a publishing studio, and custom information)
    - A `TVShow` object is also a `Bingeable` object, meaning that we should be able to retrieve an iterator to easily access the elements in a `TVShow` object
- Field `aList` is a `LinkedList` of objects of class `Episode`, which represents the `Episodes` in the `TVShow` object
    - Users may see all movies in the watchlist using the method `getEpisodes()`, which returns an unmodifiable list of the episodes in the show

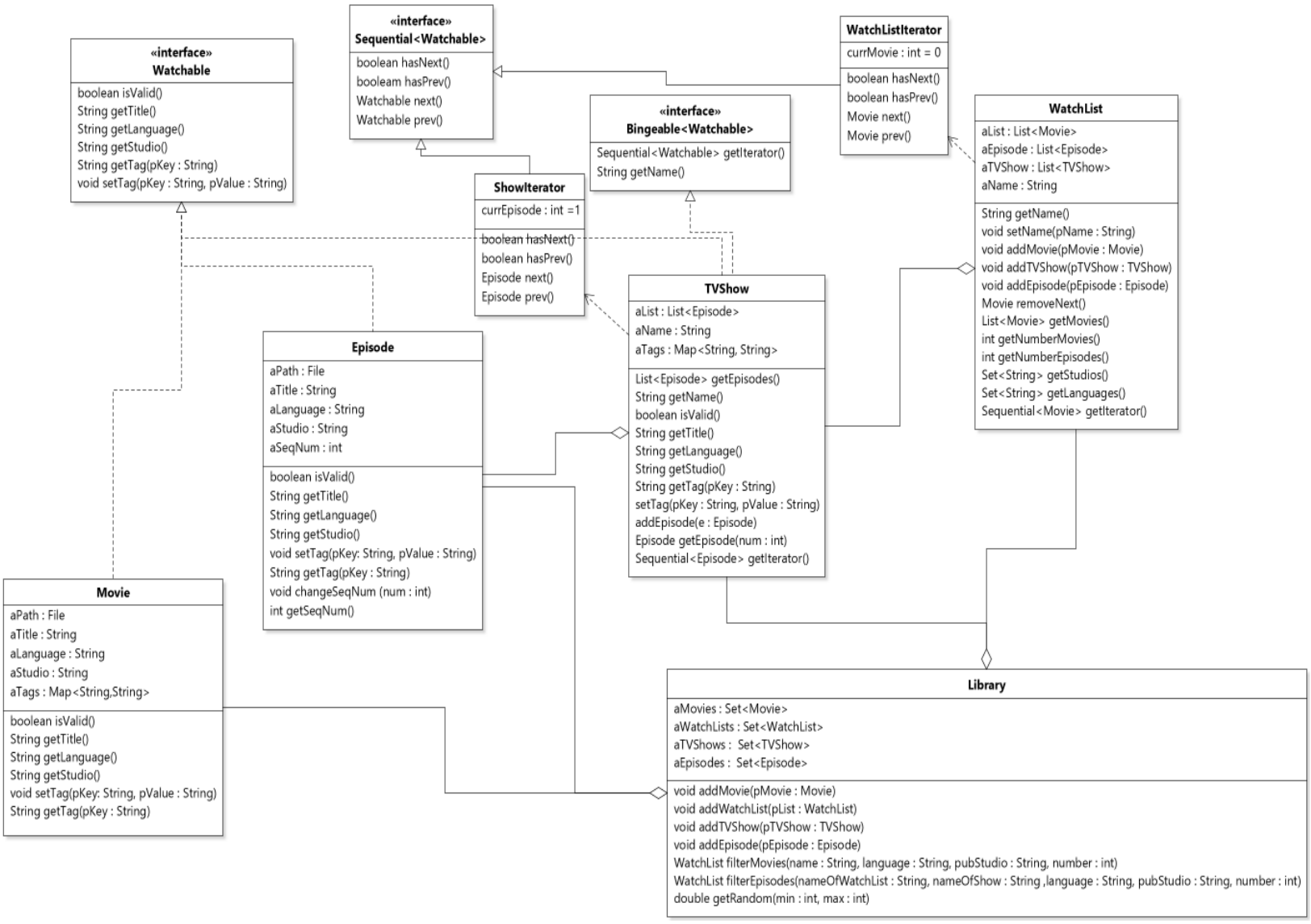
- Method `getIterator()` returns a `Sequential<Episode>` by calling the class `ShowIterator`, which is also defined within class `TVShow`
    - Class `ShowIterator` implements the abstract traversal methods declared in `Sequential<Watchable>` interface.
    - Using and declaring our own iterator makes the code more tailored to our needs (i.e. easily gives us the data we want to manipulate)

`Library` Class

- Method `filterMovies(String name, String language, String pubStudio, int number)` may be used by the client to filter `Movies` into a `WatchList`
    - The client may specify what they want to filter by in the function input
    - However, this has many drawbacks, as the code will be difficult to alter for more specific types of parameters (ex. Get 10 random movies)
- Method `filterEpisode(String nameOfWatchList, String nameOfShow, String languages, String pubStudio, int number)` is used to filter a `TVShow's Episodes`
    - The client may specify which `TVShow` they want to retrieve `Episodes` from, the language of the show, the publishing studio of the show in the function input, and how many `Episodes` they want in the `WatchList` created
    - Again, this code is quite convoluted and difficult to alter for other types of filtering as it will need to be hardcoded into the program

Using interfaces in this implementation is a beneficial choice here because interfaces allow us to have a polymorphic structure that can represent many of our objects at the same time (ex. A `Movie` object has very similar fields as an `Episode` object, and having an interface to represent these two classes, we can see their similarities very easily and avoid declaring multiple methods that behave similarly). Utilizing the Interface Segregation Principle, all classes that behave similarly utilize the same interface, but they also have their own methods that are more specific to the type of object they are.

A diagram below shows the relationships between the interfaces and classes:


«interface»
Watchable
boolean isValid()
String getTitle()
String getLanguage()
String getStudio()
String getTag(pKey : String)
void setTag(pKey : String, pValue : String)
«interface»
Sequential<Watchable>
boolean hasNext()
booleam hasPrev()
Watchable next()
Watchable prev()
WatchListIterator
currMovie : int = 0
boolean hasNext()
boolean hasPrev()
Movie next()
Movie prev()
«interface»
Bingeable<Watchable>
Sequential<Watchable> getIterator()
String getName()
WatchList
aList : List<Movie>
aEpisode : List<Episode>
aTVShow : List<TVShow>
aName : String
String getName()
void setName(pName : String)
void addMovie(pMovie : Movie)
void addTVShow(pTVShow : TVShow)
void addEpisode(pEpisode : Episode)
Movie removeNext()
List<Movie> getMovies()
int getNumberMovies()
int getNumberEpisodes()
Set<String> getStudios()
Set<String> getLanguages()
Sequential<Movie> getIterator()
ShowIterator
currEpisode : int =1
boolean hasNext()
boolean hasPrev()
Episode next()
Episode prev()
TVShow
aList : List<Episode>
aName : String
aTags : Map<String, String>
List<Episode> getEpisodes()
String getName()
boolean isValid()
String getTitle()
String getLanguage()
String getStudio()
String getTag(pKey : String)
setTag(pKey : String, pValue : String)
addEpisode(e : Episode)
Episode getEpisode(num : int)
Sequential<Episode> getIterator()
Episode
aPath : File
aTitle : String
aLanguage : String
aStudio : String
aSeqNum : int
boolean isValid()
String getTitle()
String getLanguage()
String getStudio()
void setTag(pKey: String, pValue : String)
String getTag(pKey : String)
void changeSeqNum (num : int)
int getSeqNum()
Movie
aPath : File
aTitle : String
aLanguage : String
aStudio : String
aTags : Map<String,String>
boolean isValid()
String getTitle()
String getLanguage()
String getStudio()
void setTag(pKey: String, pValue : String)
String getTag(pKey : String)
Library
aMovies : Set<Movie>
aWatchLists : Set<WatchList>
aTVShows : Set<TVShow>
aEpisodes : Set<Episode>
void addMovie(pMovie : Movie)
void addWatchList(pList : WatchList)
void addTVShow(pTVShow : TVShow)
void addEpisode(pEpisode : Episode)
WatchList filterMovies(name : String, language : String, pubStudio : String, number : int)
WatchList filterEpisodes(nameOfWatchList : String, nameOfShow : String ,language : String, pubStudio : String, number : int)
double getRandom(min : int, max : int)