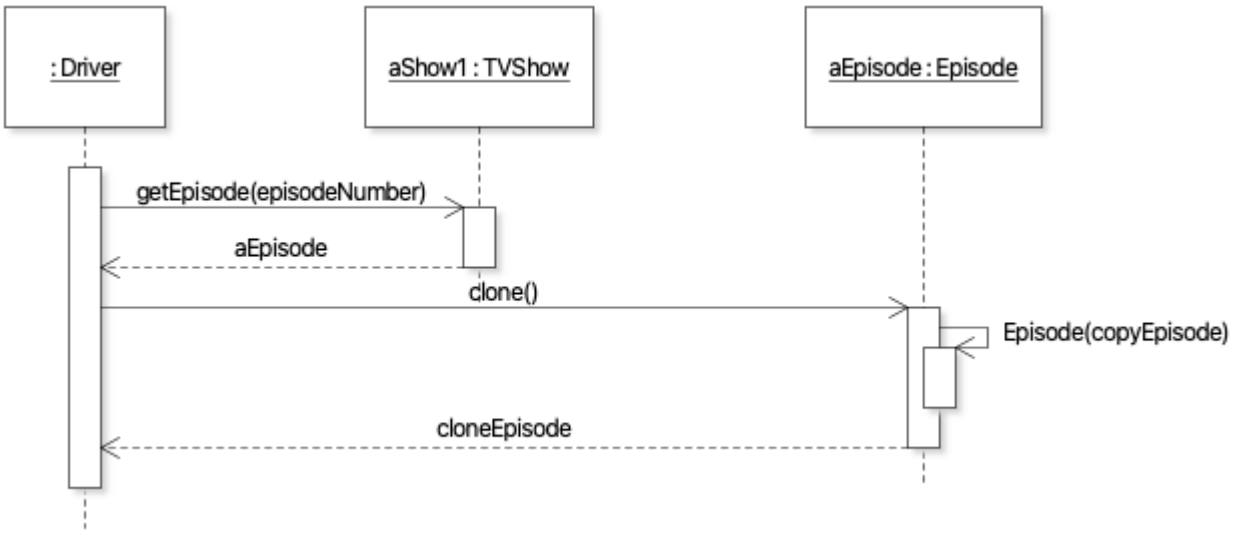
COMP 303 Neyir Erdeser
Assignment 4 260736004

Part 1:
First, I made a small modification of keeping all common elements of Episode, Movie and TVShow in the Watchable interface. This way I could keep the WatchListFilter interface cleaner. Aside from the existing LanguageFilterStrategy class, I created StudioFilterStrategy and FirstEpisode FilterStrategy classes, all implementing WatchListFilter. For the FirstEpisodeFilterStrategy, I decided not to 'punish' and input that are non-Episode, since the client may be filtering from a pile of different typed Watchables. On top of these I added the two classes AND and OR for the composition of filter strategies. Both of them by default require the strategies to compose, but after more can be added to create more complex compositions.

Part 2:
Since we aren't dealing with any polymorphic classes, I created a copy constructor for Episode with assigns all values from the original Episode object to the new one, except for the path, since that is the identifier. Then, since the client cannot (or at least should not) access the constructor directly, I added a method, clone() that calls the copy constructor and returns the new object. I thought this would be ideal since the client can already access episodes using the getEpisode(int) method from TVShow Class.



Part 3:
For part 1's testing, I tested each individual filter strategy and then the composition classes, using randomly entered values for Watchable objects.
For part 2's testing, I created a TV Show with a few episodes then selected one of them to clone. Each test method then checks each attribute to see if they were assigned properly to the new object.