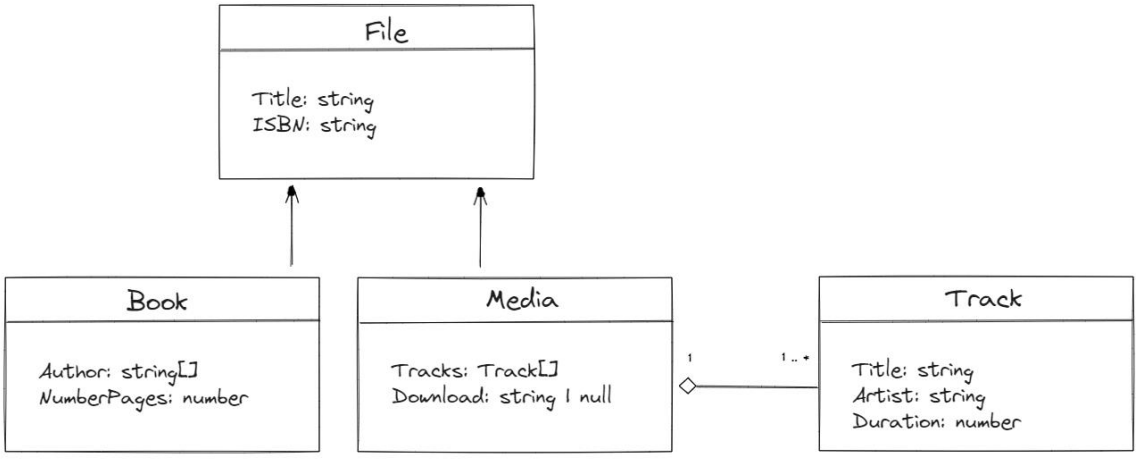
# Assignment

Natalia Rodas

## Exercise 1.1




I created a superclass File with the fields that all the classes should contain. The class Media inherits from File and has the field tracks that would contain an array of the class Track that has the title, artist and duration of each track, and the field download that can be null or contain a string for the link where the media can be downloaded (assuming is on the cloud). This Media class is for CD, DVD and Blue-ray. Finally, the class Book also inherits from File and has the fields for number of pages as an integer and the authors are saved into an array or list of strings.

## Exercise 1.2

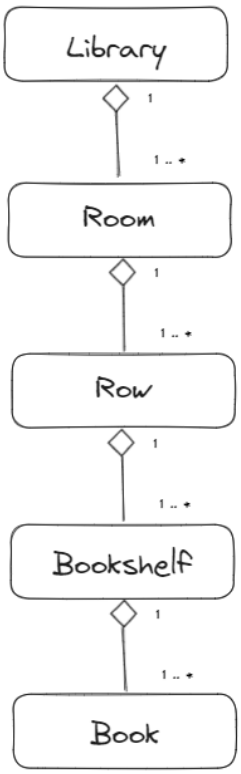
Extending from the initial types for the classes, the class Book now has the field publisher and published. I created an additional class BookManager that reads an input and parses it into creating a new instance for each Book. It also has a method for searching books that accepts a strict string, the * key that allows the string to have more characters next to it. This key * could be and the beginning of the string, at the end, or both. To make these distinctions I helped myself with regex. I tried to make all my methods reusable and tested with the test data provided.

Up until this part, it took me two hours to finish.

To evaluate several queries with the &, I created a split for this key and for each query I reused the method that looks for individual queries. Then I took the intersection of each list returned so I only kept the books that matches all queries.

Lastly, I added some tests on BookManagerTest to assert that the search methods where working according to the requirements.

# Exercise 1.3




For this part, I created a class for Library that contains a List of Rooms. The class Room in itself contains a unique id and multiple Rows. This happens again with Row, which has several instances of Bookshelf, and Bookshelf has several books. To keep track of the several Bookshelf in a Row, I created a Map<id, Bookshelf> so it is easy to find. I did the same with the Rows in Room. To look for a Book by its ISBN in the library, I keep a record of the whole Book inventory in Library as Map<ISBN, (roomId,rowId,bookshelfId)>, so the search would be more efficient. I also created methods to get the books by roomId, rowId, and bookshelfId.

Lastly, I created a fake input data and several tests to assert that these methods are working correctly.

In total, this project took me approximately 6 hours.