|  |  |
| --- | --- |
| **Type** | **Description** |
| **Enum** | * **Categories** (action, horror etc.): created through enum. * **Media Types** (book, video, game): created through enum (as well as class) for easier references. |
| **Classes** | * **Loader**: Main/helper class with the static main method and the NewLine used throughout the code. * **Subclasses**: Each media type was made into subclasses, to hold their unique attributes. * **Superclass (Media)**: Parent of all media type subclasses to hold the common attributes. * **MediaStorage**: Rest of code and logic placed within the “main” class. * **Database:** Class with the methods related to database management. |
| **Methods** | * **Menu**: Each menu has been placed within a method instead of nesting within switch/if statement to make it easier to navigate between. * **Input**: Two methods were created (one for int and one for String) to return the userinput value.   + **Int** was given extra logic to parse and check that it was a valid value returned. * **List**: Three methods were created to work with the media list: Add, Remove, and Print. |
| **Attributes** | * **mediaList**: Created an ArrayList to store the data from the .txt in memory as a list while working with a specific media type.   + **Media** (superclass) was used as the ArrayList type to only have one list, and then casting the correct subclass when adding the unique attributes to the element. * **Private:** All attributes were set to private, and getters were used for attributes needed to be reached outside the class.   + **Setters** were not needed in the objects as they only needed their initial value set through the constructor. * **mediaType**: An enum variable was created to always show the methods which list was being used. |
| **External Elements** | * **Txt**: All saved data was stored within a text document for later availability.   + **mediaList** ArrayListwas used by most methods, as all data was inserted when a media type was chosen.   + **Updating** the text document only happened when changes were made to the list (through adding or removing objects). |