

# UML

## 1. Class

AudioBooks
-author : String -langue : Langues -category : Categories
+AudioBooks(title : String, author : String, duration : int, contents : String, langue : Langues, category : Categories, id : String) +AudioBooks(title : String, author : String, duration : int, contents : String, langue : Langues, category : Categories) +getAuthor : String +getCategory : String +getLanguage : String +toString : String

Playlist
-name : String -id : UUID -audiobooksList : LinkedList<AudioBooks> -songsList : LinkedList<Song>
+Playlist(name : String, id : UUID, songsList : LinkedList<Song>, audiobooksList : LinkedList<AudioBooks>) +Playlist(name : String, songsList : LinkedList<Song>, audiobooksList : LinkedList<AudioBook>) +toString : String +getName : String +getId : String +getSongsList : LinkedList<Song> +getAudioBooksList : LinkedList<AudioBook> +showSongList : void +showAudioBookList : void

Album
-title : String -artist : String -duration : int -date : int -songList : LinkedList<Song> +idAlbum : UUID
+Album(title : String, date: int, duration : int, songs : LinkedList<Song>, artist : String, id : String) +Album(title : String, date : int, date : int, songs : LinkedList<Song>, artist : String) +getTitle : String +getArtist : String +getDuration : int +setDuration : void +getDate : int +getId : String +addSong(song : Song) : void +showAlbum : void +showAlbumInformations : void +showSongList : void +toString : String

Song
-artist : String -genre : Genres
+Song(title : String, artist : String, duration : int, contents : String, genre : Genres, id : String) +Song(title : String, artist : String, duration : int, contents : String, genre : Genres) +getGenre : String +getGenreForXML : String +getArtist : String +toString : String

MenuSelection
+menuData(sc : Scanner) : static int <i>throws NotANumberException</i> +selectAlbum5(sc : Scanner, util : Util) : static void +selectPlaylist6(util : Util, sc : Scanner) : static void +selectArtist7(util : Util, sc : Scanner) : static void +selectAuthor8(util : Util, sc : Scanner) : static void +changeContent9(util : Util, sc : Scanner, xmlfile : XmlFile) : static void +addASongToTheSongsList(util : Util, sc : Scanner) : static void +addAnAlbumToTheAlbumsList(util : Util, sc : Scanner) : static void +addASongToAnAlbum(util : Util, sc : Scanner) : static void +addAnAudioBookToTheAudioBooksList(util : Util, sc : Scanner) : static void +createAPlaylist(util : Util, sc : Scanner) : static void +deleteAPlaylist(util : Util, sc : Scanner) : static void +saveApplicationInXml(xmlfile : XmlFile, util : Util) : static void +chooseTheMusicToListen(util : Util, sc : Scanner) : static void

Music
<pre> +PlaySound(Sound : File, l : int) : static void +PlaylistOfSound(soundList : LinkedList&lt;File&gt;, l : int) : static void +listenToSomeMusic(musicFilepath : String, sc : Scanner) : static void +listenToAListOfMusics(listMusicFilepath : LinkedList&lt;String&gt;, sc : Scanner) : static void </pre>

Util
<pre> -audiobooksList : LinkedList&lt;AudioBooks&gt; -songsList : LinkedList&lt;Song&gt; -albumsList : LinkedList&lt;Album&gt; -playlistsList : LinkedList&lt;Playlist&gt;  +Util() +Util(au : LinkedList&lt;AudioBooks&gt; , so : LinkedList&lt;Song&gt;, al : LinkedList&lt;Album&gt;, pl : LinkedList&lt;Playlist&gt;) +fileTheAudioBooksList(xmlfile : XmlFile) : void +fileTheSongsList(xmlfile : XmlFile) : void +fileTheAlbumsList(xmlfile : XmlFile) : void +fileThePlaylistsList(xmlfile : XmlFile) : void +addAudiobookToTheList(audiobookToAdd : AudioBooks) : void +addSongToTheList(songToAdd : Song) : void +addAlbumToTheList(albumToAdd : Album) : void +addPlaylistToTheList(playlistToAdd : Playlist) : void +getAudioBooksListForXml : LinkedList&lt;AudioBook&gt; +getSongsListForXml : LinkedList&lt;Song&gt; +getAlbumsListForXml : LinkedList&lt;Album&gt; +getPlaylistsListForXml : LinkedList&lt;Playlist&gt; +showAudioBooksNamesOrdered(orderedOrNot : String) : void +getAudioBookIndex(thisAudioBook : String) : int <i>throws NotAnExistingAudioBook</i> +addAudioBookToNewPlaylist(thisAudioBook : String, listAudioBookPlaylist : LinkedList&lt;AudioBook&gt; ) : String +getAuthors : LinkedList&lt;String&gt; +showSongsNamesOrdered(orderedOrNot : String) : void +getSongIndex(thisSong : String) : int <i>throws NotAnExistingSong</i> +addSongToNewAlbumOrPlaylist(thisSong String, listSongAlbumOrPlaylist : LinkedList&lt;Song&gt;) : String +addSongToAlbum(thisSong : int, thisAlbum : int) : void +musicToListen(thisMusic : String, kindOfMusic : String) : String +showAlbumsNamesOrdered(orderedOrNot : String) : void +getAlbumIndex(thisAlbum : String) : int <i>throws NotAnExistingAlbum</i> +getAlbumSongsFilepath(albumWanted : String, sc : Scanner) : LinkedList&lt;String&gt; +showPlaylistsNamesOrdered(orderedOrNot : String) : void +getPlaylistIndex(thisPlaylist : String) : int <i>throws NotAnExistingPlaylist</i> +getPlaylistMusicsFilepath(playlistWanted : String, sc : Scanner) : LinkedList&lt;String&gt; </pre>

<b>XmlFile</b>
Root : Element -filepathPlaylist = "files/playlist.xml" : static String -filepathAlbum = "files/albums.xml" : static String -filepathMedias= "files/elements.xml" : static String -transformerFactory : TransformerFactory -transformer : Transformer -docFactory : DocumentBuilderFactory -docBuilder : DocumentBuilder -docPlaylist : static Document -docAlbum : static Document -docMedias : static Document
+XmlFile() +showXML(xmlWanted : String) : void +getContent(n : Node, tab : String) : String -getAudioBooks(n : Node) : AudioBook <i>throws NotACategoryException, NotALanguageException</i> +getListAudioBooks : LinkedList<AudioBook> -getSong(n : Node) : Song <i>throws NotAGenreException</i> +getListSongs : LinkedList<Song> -getAlbum(n : Node) : Album +getListAlbum : LinkedList<Album> -getPlaylist(n : Node) : Playlist +getListPlaylist : LinkedList<Playlist> +setXmlWithUtilList(util : Util) : void +writeXml(whereToWrite : String, util : Util) : void

## 2.Enum

<<enum>> <b>Categories</b>
=JEUNESSE, NOVEL, THEATRE, SPEECH, DOCUMENTARY -category : String
-Categories : (category : String) +getCategory : String

<<enum>> <b>Genres</b>
=JAZZ, CLASSIC, HIPHOP, ROCK, POP, RAP, KPOP -genre : String
-Genres : (genre : String) +getGenre : String

<<enum>> <b>Languages</b>
=FRENCH, ENGLISH, ITALIAN, SPANISH, GERMAN, COREEN, KOREAN, CHINESE, RUSSIAN -language : String
-Language : (language : String) +getLanguage : String

### 3.Abstract

<b>&lt;&lt;abstract&gt;&gt; Media</b>
#titre : String #int : Duration #id : UUID #contents : String +toString : abstract String
+getTitle : String +getContent : String +getId : String +getDuration : int

### 4.Exception

<b>&lt;&lt;Exception&gt;&gt; NotACategoryException</b>
+NotACategoryException(titleSong : String)

<b>&lt;&lt;Exception&gt;&gt; NotAGenreException</b>
+NotAGenreException(titleSong : String)

<b>&lt;&lt;Exception&gt;&gt; NotALanguageException</b>
+NotALanguageException(titleSong : String)

<b>&lt;&lt;Exception&gt;&gt; NotANumberException</b>
+NotANumberException()

<b>&lt;&lt;Exception&gt;&gt; NotAnExistingAlbum</b>
+NotAnExistingAlbum()

<b>&lt;&lt;Exception&gt;&gt;</b> <b>NotAnExistingAudioBook</b>
+NotAnExistingAudioBook()

<b>&lt;&lt;Exception&gt;&gt;</b> <b>NotAnExistingPlaylist</b>
+NotAnExistingPlaylist()

<b>&lt;&lt;Exception&gt;&gt;</b> <b>NotAnExistingSong</b>
+NotAnExistingSong()

## 5.Interface

<b>&lt;&lt;Interface&gt;&gt;</b> <b>InterfaceDuration</b>
+getDuration : int

<b>&lt;&lt;Interface&gt;&gt;</b> <b>InterfaceGenre</b>
+getGenre : String

<b>&lt;&lt;Interface&gt;&gt;</b> <b>InterfaceString</b>
+toString : String

<b>&lt;&lt;Interface&gt;&gt;</b> <b>InterfaceTitle</b>
+getTitle : String