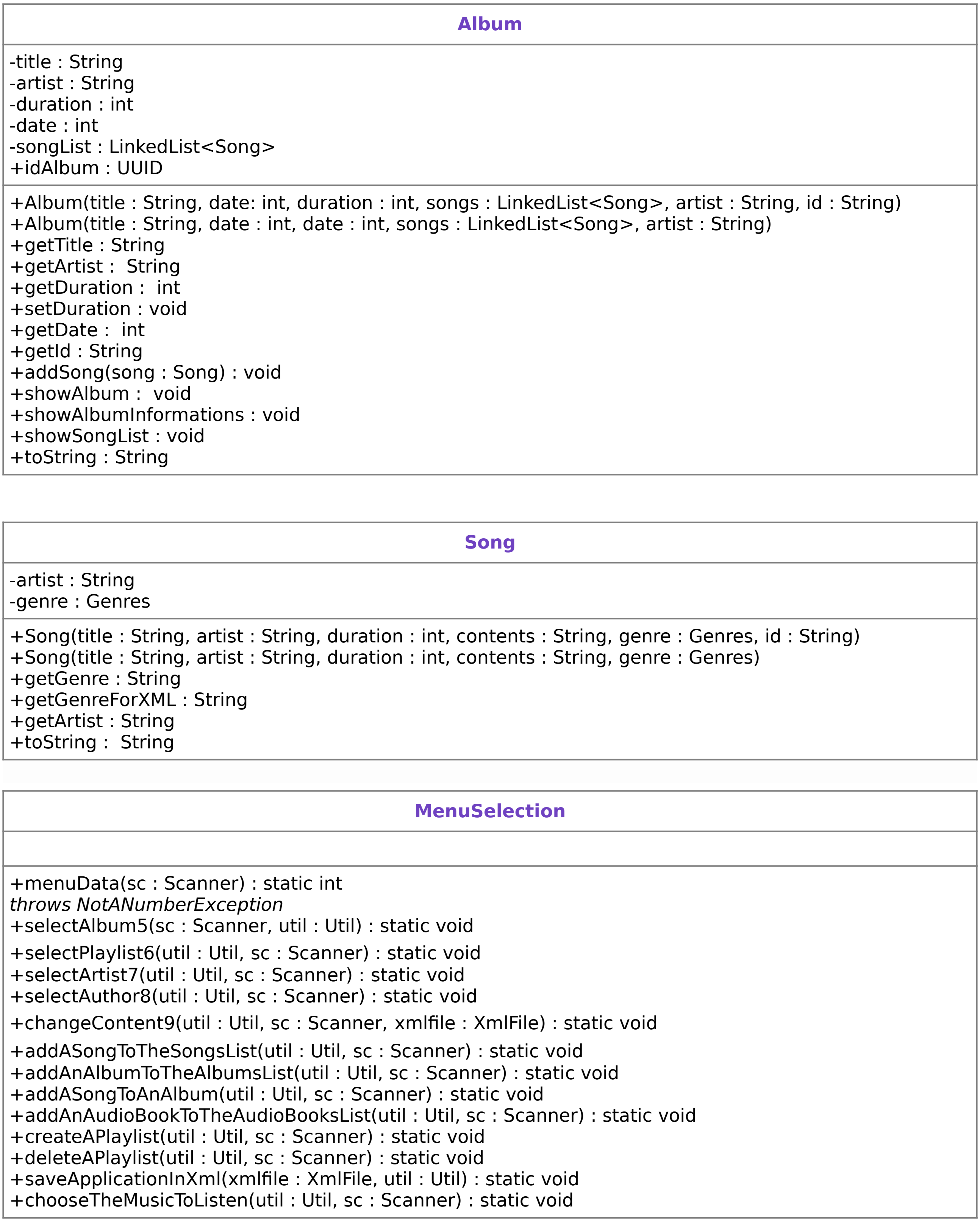
UML

1.Class

AudioBooks
-author : String -language : Languages -category : Categories
+AudioBooks(title : String, author : String, duration : int, contents : String, language : Languages, category : Categories, id : String) +AudioBooks(title : String, author : String, duration : int, contents : String, language : Languages, 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
throws NotANumberException
+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
+PlaySound(Sound : File, I : int) : static void +PlayListOfSound(soundList : LinkedList<File>, I : int) : static void +listenToSomeMusic(musicFilepath : String, sc : Scanner) : static void +listenToAListOfMusics(listMusicFilepath : LinkedList<String>, sc : Scanner) : static void

Util
-audiobooksList : LinkedList<AudioBooks> -songsList : LinkedList<Song> -albumsList : LinkedList<Album> -playlistsList : LinkedList<Playlist>
+Util() +Util(au : LinkedList<AudioBooks> , so : LinkedList<Song>, al : LinkedList<Album>, pl : LinkedList<Playlist>) +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<AudioBook> +getSongsListForXml : LinkedList<Song> +getAlbumsListForXml : LinkedList<Album> +getPlaylistsListForXml : LinkedList<Playlist> +showAudioBooksNamesOrdered(orderedOrNot : String) : void +getAudioBookIndex(thisAudioBook : String) : int *throws NotAnExistingAudioBook* +addAudioBookToNewPlaylist(thisAudioBook : String, listAudioBookPlaylist : LinkedList<AudioBook>) : String +getAuthors : LinkedList<String> +showSongsNamesOrdered(orderedOrNot : String) : void +getSongIndex(thisSong : String) : int *throws NotAnExistingSong* +addSongToNewAlbumOrPlaylist(thisSong String, listSongAlbumOrPlaylist : LinkedList<Song>) : String +addSongToAlbum(thisSong : int, thisAlbum : int) : void +musicToListen(thisMusic : String, kindOfMusic : String) : String +showAlbumsNamesOrdered(orderedOrNot : String) : void +getAlbumIndex(thisAlbum : String) : int *throws NotAnExistingAlbum* +getAlbumSongsFilepath(albumWanted : String, sc : Scanner) : LinkedList<String> +showPlaylistsNamesOrdered(orderedOrNot : String) : void +getPlaylistIndex(thisPlaylist : String) : int *throws NotAnExistingPlaylist* +getPlaylistMusicsFilepath(playlistWanted : String, sc : Scanner) : LinkedList<String>

XmlFile
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 *throws NotACategoryException, NotALanguageException* +getListAudioBooks : LinkedList<AudioBook> -getSong(n : Node) : Song *throws NotAGenreException* +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>> **Categories**
=JEUNESSE, NOVEL, THEATRE, SPEECH, DOCUMENTARY -category : String
-Categories : (category : String) +getCategory : String

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

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

3.Abstract

<<abstract>> **Media**
#titre : String #int : Duration #id : UUID #contents : String +toString : abstract String
+getTitle : String +getContent : String +getId : String +getDuration : int

4.Exception

<<Exception>> **NotACategoryException**
+NotACategoryException(titleSong : String)

<<Exception>> **NotAGenreException**
+NotAGenreException(titleSong : String)

<<Exception>> **NotALanguageException**
+NotALanguageException(titleSong : String)

<<Exception>> **NotANumberException**
+NotANumberException()

<<Exception>> **NotAnExistingAlbum**
+NotAnExistingAlbum()

<<Exception>> **NotAnExistingAudioBook**
+NotAnExistingAudioBook()

<<Exception>> **NotAnExistingPlaylist**
+NotAnExistingPlaylist()

<<Exception>> **NotAnExistingSong**
+NotAnExistingSong()

5.Interface

<<Interface>> **InterfaceDuration**
+getDuration : int

<<Interface>> **InterfaceGenre**
+getGenre : String

<<Interface>> **InterfaceString**
+toString : String

<<Interface>> **InterfaceTitle**
+getTitle : String