

Categorie

- categorie : String

- Categorie(String)
+ toString() : String

Chanson

- artiste : String
- contenu : String
- genre : Genre
- duree : int

+ Chanson(String,String,int,String,String,Genre)
+ toString() : String
+ getArtiste() : String
+ getDuree() : int
+ getContenu() : String
+ getGenre() : Genre
+ getChansonGenre() : String
+ play() : void
+ stop() : void

Genre

- genre : String

- Genre(String)
+ toString() : String

Langue

- langue : String

- Langue(String)
+ toString() : String

LivreAudio

- auteur : String
- contenu : String
- langue : Langue
- categorie : Categorie
- duree : int

- LivreAudio(String,String,int,String,Langue,Categorie)
+ toString() : String
+ getAuteur() : String
+ getDuree() : int
+ getContenu() : String
+ getLivreLangue() : String
+ getLangue() : Langue
+ getCategorie() : Categorie
+ play() : void
+ stop() : void

EmptyFichException

+ EmptyFichException(String)

Playlist

+Playlist(String nom, String ID)
+toString() : String
+getNom() : String

Album

-artiste: String
-duree:int
-date:Date

+Album(String titre, String artiste, int
duree, Date date, String ID)
+toString() : String
+getArtiste(): String
+getDuree():int
+getDate(); Date
+play():void
+stop():void

<<abstract>> Gestion

+bestQuality():void
+lessQuality():void
+moreSpeed() :void
+lessSpeed():void

Option

- vitesse:int
- quality : String

+Options (int vitesse, String quality)
+bestQuality():void
+lessQuality():void
+moreSpeed() :void
+lessSpeed():void

SortByDate

+ compare(Album,Album) : int

SortByGenre

+ compare(Chanson,Chanson) : int

SortByAuthor

+ compare(LivreAudio,LivreAudio) : int

<<interface>>
Audio

+play():void
+stop():void

Son

titre: String
#ID : String

+getID():String
+getTitre():String
+getAlbum(): String
+getPlaylist():String
+getPartID():String

MusicHubConsole

+MusicHubConsole()
+\$ main(String args):void

MusicHub

scanner:Scanner = Scanner
albums:LinkedList<Album>
playlists: LinkedList<Playlist>
chansons:LinkedList<Chanson>
livres: LinkedList<LivreAudio>

+MusicHub()
+recupPlaylists() : void
+ recupElements() : void
+ recupAlbums() :void
+ ajoutChansons() : void
+verifDate(String date_a_verif) : boolean
+ ajoutAlbum(): void
+ajoutChansonAlbum() : void
+ajoutLivreAudio():void
+ajoutPlaylist() : void
+suppressPlaylist():void
+sauvegarde():void
+affchanson_album():void
+afftitre_date():void
+afftitre_genre():void
+affplaylist():void
+afflivre_auteur():void

FileXML

-transformer:Transformer
-documentFactory:DocumentBuilderFactory
-documentBuilder:DocumentBuilder
-XML_ELEMENTS:String
-XML_ALBUMS:String
-XML_PLAYLISTS:String
-XML_ELEMENTS_TEST:String
-XML_ALBUMS_TEST:String
-XML_PLAYLISTS_TEST:String

+FileXML
+createXMLFile(Document document, String filePath) : void
+createXMLDocument() : Document
+parseXMLFile (String filePath): NodeList
+readPlaylists(LinkedList<Playlist> playlists):LinkedList<Playlist>
+writePlaylists(LinkedList<Playlist> playlists):void
+readAlbums(LinkedList<Album> albums):LinkedList<Album>
+writeAlbums(LinkedList<Album> albums):void
+readElementsChansons(LinkedList<Chanson>
chansons):LinkedList<Chanson>
+readElementsLivres(LinkedList<LivreAudio> livres): LinkedList<LivreAudio>
+writeElements(LinkedList<Chanson> chansons, LinkedList<LivreAudio>
livres):void
+createlD():String



playlists.xml



albums.xml



elements.xml



compile.sh



run.sh



cx



javadoc



touts es fichiers .class
correspondants à
ceux .java



tous les fichiers .class
correspondant à ceux
.java



tous les fichiers .class
correspondant à ceux
.java