src

### musichub

### util

### 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)

# business

# Playlist

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

# Album

+getNom(): String

-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

vitesse:int

Option

- 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

# main

+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

 $+ verif Date (String \ date \_a\_verif) : boolean \\$ 

+ 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

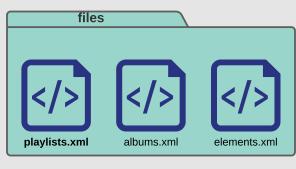
## +FileXML

+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>

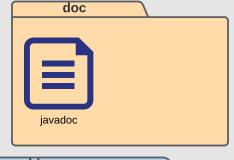
livres):void +createID():String











# bin

# musichub



touts es fichers .class correspondants à ceux .java

# **business**

tous les fichiers .class correspondant à ceux





correspondant à ceux .java