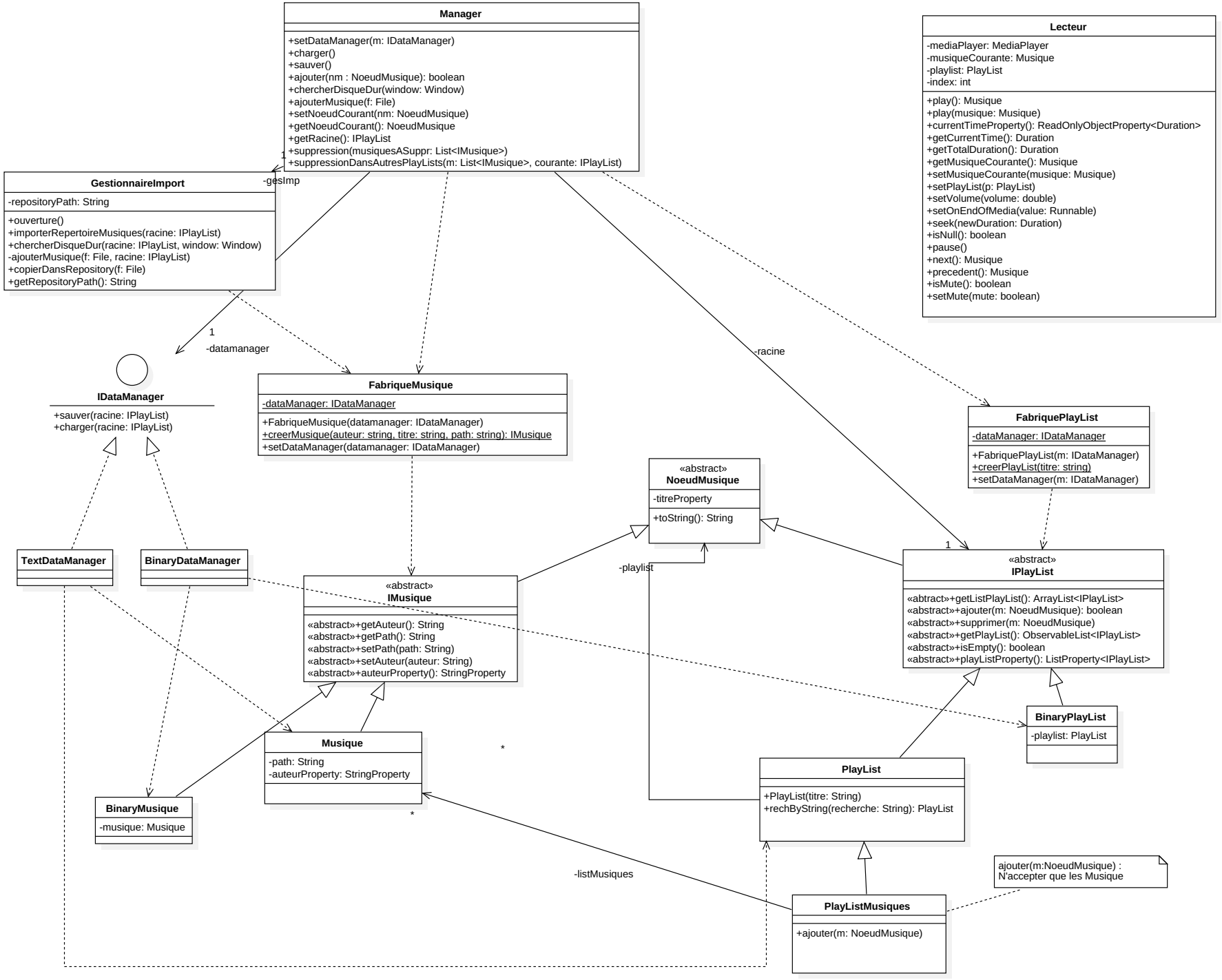
Model::Main


Manager
+setDataManager(m: IDataManager)
+charger()
+sauver()
+ajouter(nm : NoeudMusique): boolean
+chercherDisqueDur(window: Window)
+ajouterMusique(f: File)
+setNoeudCourant(nm: NoeudMusique)
+getNoeudCourant(): NoeudMusique
+getRacine(): IPlayList
+suppression(musiquesASuppr: List<IMusique>)
+suppressionDansAutresPlayLists(m: List<IMusique>, courante: IPlayList)
Lecteur
-mediaPlayer: MediaPlayer
-musiqueCourante: Musique
-playlist: PlayList
-index: int
+play(): Musique
+play(musique: Musique)
+currentTimeProperty(): ReadOnlyObjectProperty<Duration>
+getCurrentTime(): Duration
+getTotalDuration(): Duration
+getMusiqueCourante(): Musique
+setMusiqueCourante(musique: Musique)
+setPlayList(p: PlayList)
+setVolume(volume: double)
+setOnEndOfMedia(value: Runnable)
+seek(newDuration: Duration)
+isNull(): boolean
+pause()
+next(): Musique
+precedent(): Musique
+isMute(): boolean
+setMute(mute: boolean)
GestionnaireImport
-repositoryPath: String
+ouverture()
+importerRepertoireMusiques(racine: IPlayList)
+chercherDisqueDur(racine: IPlayList, window: Window)
-ajouterMusique(f: File, racine: IPlayList)
+copierDansRepository(f: File)
+getRepositoryPath(): String
1
-gesImp
1
-datamanager
-racine
IDataManager
+sauver(racine: IPlayList)
+charger(racine: IPlayList)
FabriqueMusique
-dataManager: IDataManager
+FabriqueMusique(datamanager: IDataManager)
+creerMusique(auteur: string, titre: string, path: string): IMusique
+setDataManager(datamanager: IDataManager)
FabriquePlayList
-dataManager: IDataManager
+FabriquePlayList(m: IDataManager)
+creerPlayList(titre: string)
+setDataManager(m: IDataManager)
«abstract»
NoeudMusique
-titreProperty
+toString(): String
1
TextDataManager
BinaryDataManager
-playlist
«abstract»
IMusique
«abstract»+getAuteur(): String
«abstract»+getPath(): String
«abstract»+setPath(path: String)
«abstract»+setAuteur(auteur: String)
«abstract»+auteurProperty(): StringProperty
«abstract»
IPlayList
«abtract»+getListPlayList(): ArrayList<IPlayList>
«abstract»+ajouter(m: NoeudMusique): boolean
«abstract»+supprimer(m: NoeudMusique)
«abstract»+getPlayList(): ObservableList<IPlayList>
«abstract»+isEmpty(): boolean
«abstract»+playListProperty(): ListProperty<IPlayList>
BinaryPlayList
-playlist: PlayList
Musique
-path: String
-auteurProperty: StringProperty
*
*
BinaryMusique
-musique: Musique
PlayList
+PlayList(titre: String)
+rechByString(recherche: String): PlayList
-listMusiques
ajouter(m:NoeudMusique) :
N'accepter que les Musique
PlayListMusiques
+ajouter(m: NoeudMusique)