HarmonyMuse Project Class Diagram

Model

three.note.structures

MajorTriad
- root:Note
- third:Note
- fifth:Note
- quility:String
- inversion:String

+getRoot:Note
+getThird:Note
+getFifth:Note
+getQuality:String
+getInversion:String
+setRoot(Note):viod
+setThird(Note):void
+setFifth(Note):void
+setQuality(String):void
+setInversion(String):voic

file.handling

ReadFromFile

+<E extends Chord>
readFile(filename, chordSequence):ChordSequence

WriteToFile

+writeToFile(filename chordSequence):void

Database

gui

<<Controller>> ChordReadGUI

<<Controller>> ChordEntryGUI

ChordSequenceEntryPage
-controller:ChordEntryGUIController
-temp:ArrayList<Chord>
-filename:String
-tonalCenter:Note
-labels:Label

+start():void
+changed():void (for toggle buttons)
+anonymousEvent:submitFIle
+anonymoustEvent:submitChord
+anonymoustEvent:submitTonalCenter
+anonymoustEvent:submitSequence
+stop():void

ReadSequencePage
-desktop:Desktop
-controller:ChordReadGUIController
-fileChooserLabel:Label

-configureFileChooder():void
-openFile(String):String
-start():void
-anonymousEvent:chooses file and analyzes
-stop():void

general.containers

<<interface>> Triad

+getThird:Note
+getFifth:Note
+setRoot(Note)
+setThird(Note)
+setFifth(Note)

Chord
-rawData:Note[]
-intervals:ArrayList<Intervals>

+getNotes():Notes[]
+getIntervals():ArrayList<Interval>
+abstract getRoot(): void
+setQuality(String)
+replaceAccidental(Note)

builders

ChordSequence
-sequence:ArrayList<E>
-progression:ArrayList<Intervals>
-tonalCenter:Note
-size:int

+getTonalCenter:Note
+getChord:E
+getSize:int
+setProgression:void

TriadFactory<T extends Triad>
-distances:Integer[]

+buildTriad:<T>

ChordBuilder
-inversions:String[]
-quality:String

+getRoot():null
+getInversions():String[]
+getQuality():null
+setQuality(String):void
+classify(ChordBuilder):Chord

Note
-name:String
-intValue:int
-enharmonics:ArrayList<String>
-notesMap:Map<Integer, ArrayList<String>>

+buildNotesMap:void
+getName():String
+getIntValue():int
+getEnharmonics():ArrayList<String>

Interval
-intervalQualities:String[]
-enharmonicQualities:String[]
-rawInterval:Pair<Note,Note>
-intValue:Integer
-distance:int
-quality:String
-enharmonic:String

+getTopNote:Note
+getBottomNote:Note
+getIntValue:Integer
+getEnharmonic:String
+setQuality:void
+setDistance:void

classifiers

TriadClassifier
-triadQualities:String[]

+classifyTriad(ChordBuilder, Note, Note):Chord
+setRootPosClosed(Triad, ChordBuilder)
+setFirstInvClosed(Triad, Chorduilder)
+setSecondInvClosed(Triad, ChordBuilder)

utilities

CircularlyLinkedList

+add(E):void
+getSize():int
+distance(E, Integer):int

Node<E>