Team 68

Phillip Hrinko, philh97
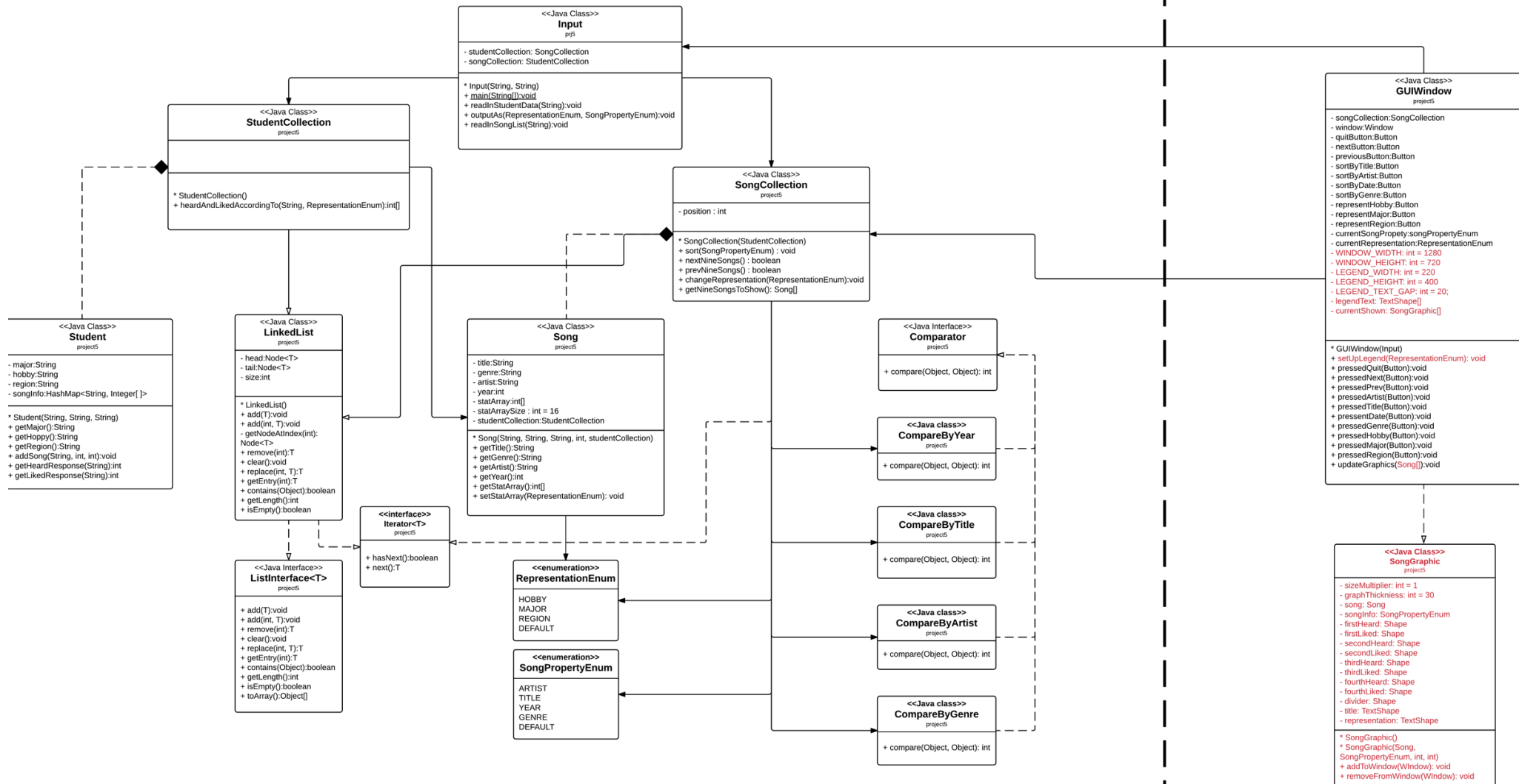
Julian Nguyen, julian18

Jooyoung Whang, joo918

UML Change Prose

- GUIWindow
    - Added static fields for the window height, the window width, the legend height, the legend width, and the legend text gap.
    - Added field for a text shape that represents the text in the legend (the selected representation and options), TextShape[] legendText.
    - Added field that represents the current songs being displayed, SongGraphic[] currentShown.
    - Added a method that creates and updates the legend, setupLegend(RepresentationEnum)
    - Added a parameter to updateGraphics(Song[]) which is the current collection of songs being displayed.
    - Created an inner class called SongGraphic that represents the glyph for each song that is being displayed.

<<Java Class>>
Input
prj5

- studentCollection: SongCollection
- songCollection: StudentCollection

* Input(String, String)
+ main(String[]):void
+ readInStudentData(String):void
+ outputAs(RepresentationEnum, SongPropertyEnum):void
+ readInSongList(String):void

<<Java Class>>
StudentCollection
project5

* StudentCollection()
+ heardAndLikedAccordingTo(String, RepresentationEnum):int[]

<<Java Class>>
SongCollection
project5

- position : int

* SongCollection(StudentCollection)
+ sort(SongPropertyEnum) : void
+ nextNineSongs() : boolean
+ prevNineSongs() : boolean
+ changeRepresentation(RepresentationEnum):void
+ getNineSongsToShow(): Song[]

<<Java Class>>
GUIWindow
project5

- songCollection:SongCollection
- window:Window
- quitButton:Button
- nextButton:Button
- previousButton:Button
- sortByTitle:Button
- sortByArtist:Button
- sortByDate:Button
- sortByGenre:Button
- representHobby:Button
- representMajor:Button
- representRegion:Button
- currentSongPropety:songPropertyEnum
- currentRepresentation:RepresentationEnum
- WINDOW_WIDTH: int = 1280
- WINDOW_HEIGHT: int = 720
- LEGEND_WIDTH: int = 220
- LEGEND_HEIGHT: int = 400
- LEGEND_TEXT_GAP: int = 20;
- legendText: TextShape[]
- currentShown: SongGraphic[]

* GUIWindow(Input)
+ setUpLegend(RepresentationEnum): void
+ pressedQuit(Button):void
+ pressedNext(Button):void
+ pressedPrev(Button):void
+ pressedArtist(Button):void
+ pressedTitle(Button):void
+ pressedDate(Button):void
+ pressedGenre(Button):void
+ pressedHobby(Button):void
+ pressedMajor(Button):void
+ pressedRegion(Button):void
+ updateGraphics(Song[]):void

<<Java Class>>
Student
project5

- major:String
- hobby:String
- region:String
- songInfo:HashMap<String, Integer[ ]>

* Student(String, String, String)
+ getMajor():String
+ getHobby():String
+ getRegion():String
+ addSong(String, int, int):void
+ getHeardResponse(String):int
+ getLikedResponse(String):int

<<Java Class>>
LinkedList
project5

- head:Node<T>
- tail:Node<T>
- size:int

* LinkedList()
+ add(T):void
+ add(int, T):void
- getNodeAtIndex(int):Node<T>
+ remove(int):T
+ clear():void
+ replace(int, T):T
+ getEntry(int):T
+ contains(Object):boolean
+ getLength():int
+ isEmpty():boolean

<<Java Class>>
Song
project5

- title:String
- genre:String
- artist:String
- year:int
- statArray:int[]
- statArraySize : int = 16
- studentCollection:StudentCollection

* Song(String, String, String, int, studentCollection)
+ getTitle():String
+ getGenre():String
+ getArtist():String
+ getYear():int
+ getStatArray():int[]
+ setStatArray(RepresentationEnum): void

<<Java Interface>>
Comparator
project5

+ compare(Object, Object): int

<<Java class>>
CompareByYear
project5

+ compare(Object, Object): int

<<Java class>>
CompareByTitle
project5

+ compare(Object, Object): int

<<Java class>>
CompareByArtist
project5

+ compare(Object, Object): int

<<Java class>>
CompareByGenre
project5

+ compare(Object, Object): int

<<interface>>
Iterator<T>
project5

+ hasNext():boolean
+ next():T

<<Java Interface>>
ListInterface<T>
project5

+ add(T):void
+ add(int, T):void
+ remove(int):T
+ clear():void
+ replace(int, T):T
+ getEntry(int):T
+ contains(Object):boolean
+ getLength():int
+ isEmpty():boolean
+ toArray():Object[]

<<enumeration>>
RepresentationEnum

HOBBY
MAJOR
REGION
DEFAULT

<<enumeration>>
SongPropertyEnum

ARTIST
TITLE
YEAR
GENRE
DEFAULT

<<Java Class>>
SongGraphic
project5

- sizeMultiplier: int = 1
- graphThickniess: int = 30
- song: Song
- songInfo: SongPropertyEnum
- firstHeard: Shape
- firstLiked: Shape
- secondHeard: Shape
- secondLiked: Shape
- thirdHeard: Shape
- thirdLiked: Shape
- fourthHeard: Shape
- fourthLiked: Shape
- divider: Shape
- title: TextShape
- representation: TextShape

* SongGraphic()
* SongGraphic(Song, SongPropertyEnum, int, int)
+ addToWindow(WIndow): void
+ removeFromWindow(WIndow): void

BACK END

FRONT END