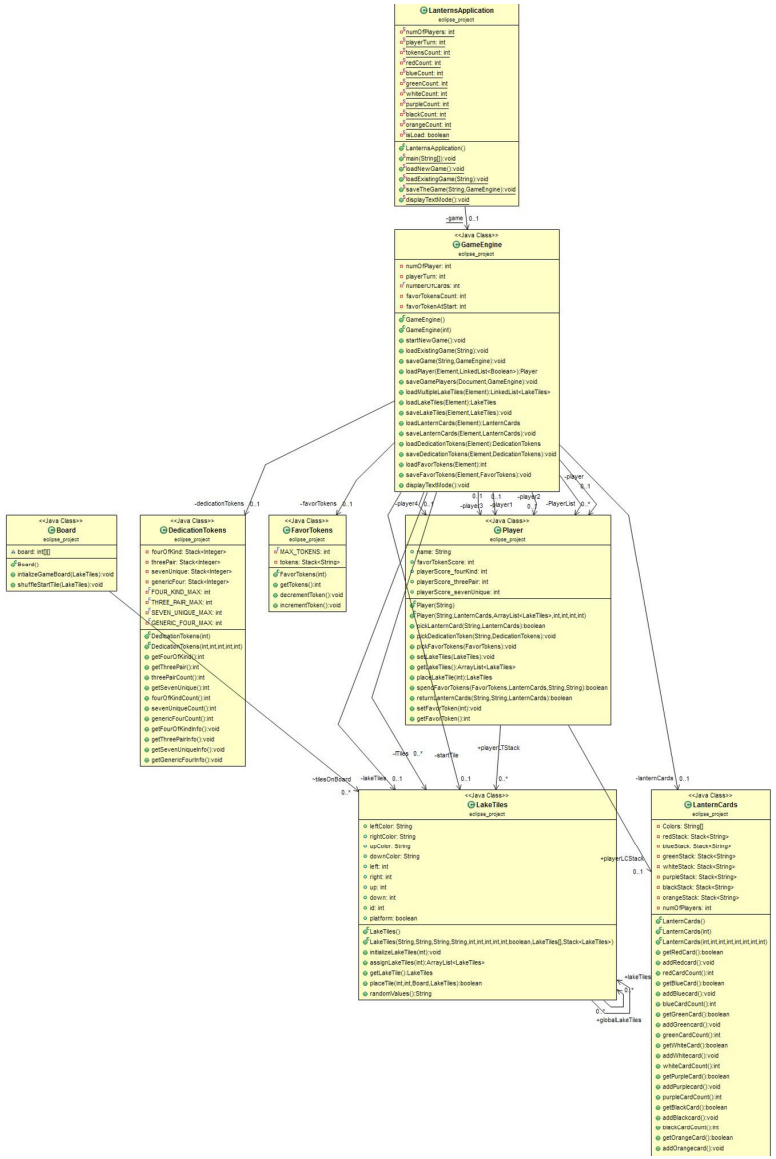
LanternsApplication
eclipse_project
numOfPlayers: int
playerTurn: int
tokensCount: int
redCount: int
blueCount: int
greenCount: int
whiteCount: int
purpleCount: int
blackCount: int
orangeCount: int
isLoad: boolean
LanternsApplication()
main(String[]):void
loadNewGame():void
loadExistingGame(String):void
saveTheGame(String,GameEngine):void
displayTextMode():void
-game
0..1
<<Java Class>>
GameEngine
eclipse_project
numOfPlayer: int
playerTurn: int
numberOfCards: int
favorTokensCount: int
favorTokensAtStart: int
GameEngine()
GameEngine(int)
startNewGame():void
loadExistingGame(String):void
saveGame(String,GameEngine):void
loadPlayer(Element,LinkedList<Boolean>):Player
saveGamePlayers(Document,GameEngine):void
loadMultipleLakeTiles(Element):LinkedList<LakeTiles>
loadLakeTiles(Element):LakeTiles
saveLakeTiles(Element,LakeTiles):void
loadLanternCards(Element):LanternCards
saveLanternCards(Element,LanternCards):void
loadDedicationTokens(Element):DedicationTokens
saveDedicationTokens(Element,DedicationTokens):void
loadFavorTokens(Element):int
saveFavorTokens(Element,FavorTokens):void
displayTextMode():void
-dedicationTokens
0..1
-favorTokens
0..1
-player
0..1
-player4
0..1
-player3
0..1
-player1
0..1
-player2
0..1
-PlayerList
0..*
<<Java Class>>
Board
eclipse_project
board: int[][]
Board()
initializeGameBoard(LakeTiles):void
shuffleStartTile(LakeTiles):void
<<Java Class>>
DedicationTokens
eclipse_project
fourOfKind: Stack<Integer>
threePair: Stack<Integer>
sevenUnique: Stack<Integer>
genericFour: Stack<Integer>
FOUR_KIND_MAX: int
THREE_PAIR_MAX: int
SEVEN_UNIQUE_MAX: int
GENERIC_FOUR_MAX: int
DedicationTokens(int)
DedicationTokens(int,int,int,int,int)
getFourOfKind():int
getThreePair():int
threePairCount():int
getSevenUnique():int
fourOfKindCount():int
sevenUniqueCount():int
genericFourCount():int
getFourOfKindInfo():void
getThreePairInfo():void
getSevenUniqueInfo():void
getGenericFourInfo():void
<<Java Class>>
FavorTokens
eclipse_project
MAX_TOKENS: int
tokens: Stack<String>
FavorTokens(int)
getTokens():int
decrementToken():void
incrementToken():void
<<Java Class>>
Player
eclipse_project
name: String
favorTokenScore: int
playerScore_fourKind: int
playerScore_threePair: int
playerScore_sevenUnique: int
Player(String)
Player(String,LanternCards,ArrayList<LakeTiles>,int,int,int,int)
pickLanternCard(String,LanternCards):boolean
pickDedicationToken(String,DedicationTokens):void
pickFavorTokens(FavorTokens):void
setLakeTiles(LakeTiles):void
getLakeTiles():ArrayList<LakeTiles>
pickLakeTile(int):LakeTiles
spendFavorTokens(FavorTokens,LanternCards,String,String):boolean
returnLanternCards(String,String,LanternCards):boolean
setFavorToken(int):void
getFavorToken():int
-Tiles
0..*
-startTile
0..1
+playerTStack
0..*
-tilesOnBoard
0..*
-lakeTiles
0..1
-lanternCards
0..1
<<Java Class>>
LakeTiles
eclipse_project
leftColor: String
rightColor: String
upColor: String
downColor: String
left: int
right: int
up: int
down: int
id: int
platform: boolean
LakeTiles()
LakeTiles(String,String,String,String,int,int,int,int,int,boolean,LakeTiles[],Stack<LakeTiles>)
initializeLakeTiles(int):void
assignLakeTiles(int):ArrayList<LakeTiles>
getLakeTile():LakeTiles
placeTile(int,int,Board,LakeTiles):boolean
randomValues():String
+playerLCStack
0..1
+lakeTiles
0..*
0..*
+globalLakeTiles
<<Java Class>>
LanternCards
eclipse_project
Colors: String[]
redStack: Stack<String>
blueStack: Stack<String>
greenStack: Stack<String>
whiteStack: Stack<String>
purpleStack: Stack<String>
blackStack: Stack<String>
orangeStack: Stack<String>
numOfPlayers: int
LanternCards()
LanternCards(int)
LanternCards(int,int,int,int,int,int,int,int)
getRedCard():boolean
addRedcard():void
redCardCount():int
getBlueCard():boolean
addBluecard():void
blueCardCount():int
getGreenCard():boolean
addGreencard():void
greenCardCount():int
getWhiteCard():boolean
addWhitecard():void
whiteCardCount():int
getPurpleCard():boolean
addPurplecard():void
purpleCardCount():int
getBlackCard():boolean
addBlackcard():void
blackCardCount():int
getOrangeCard():boolean
addOrangecard():void