Quick tips

Comment

Record video

UML class diagrams

Read more about structuring UML class diagrams here. And for additional help, watch this tutorial about basic UML class diagrams.

Click the Quick Action to easily record a video before you share!

Lucidchart basics

Abstract Class

+ operation1(params):returnType
- operation2(params)

1..*

Wrote

1..*

«interface»
Interface

«enumeration»
AccountState

«use»

«entity»
Interface

+ operation1(params):returnType
- operation2(params)

0..12

0..3

Class

+ attribute1:type = defaultValue
+ attribute2:type

+ operation1(params):returnType
- operation2(params)

0..*

1

Library

Add multiplicities and change the line direction by clicking the gear icon.

0..*

1

Class

Board

+ Pieces:vector<ChessPiece> = null
+ attribute2:type
- attribute3:type

+ InitialiseBoard(params):board
+ MovePiece(PieceLocation, PieceLocation):bool
+SetPieceStatus(PieceLocation):bool
+ SaveBoard(filepath):bool
+LoadBoard(filepath):bool
+SetGameInfo(GameInfo):void
+GetBoard(): vector<ChessPiece>
-LookupObj(PieceLocation):ChessPiece

1

1..*

ChessPiece

- color:enum = black
-peicetype:enum=pawn
- position:array = [e,4]
- status:enum = in_play
- ID:integer = 1
-SupportedMoves:array = [1]

+ ChessPiece(color, peicetype,poisition, status):void
+ SetPosition(params):void
+ SetStatus(params):void
+SetID():void
+GetID():integer
+GetStatus():enum
+GetPosition():array
+GetSupportedMoves():array

Game

+ gameInfo:GameInfo = null
+ attribute2:type
- attribute3:type

+ AddPlayer(params):Player
+ MovePiece(PieceLocation)
- StartTime()
-StopTimer()
-ProcessTimeout()
-UpdateGameStatus()
- UpdateGameOutcome()

«use»

Rules

+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type

+EvaluateMove(PieceLocation,PieceLocation, vector<ChessPiece>):bool