room

Room

-name : String
-topLeft_x : Int
-topeLeft_y : Int
-BottonRight_y : int
-bottomRight_x : Int
-listOfItems : List<Item>

+getName() : String
+getLocation() : int[4]
+getItems() : List<Items>

item

Item

-name : String
-damage : Int

+Item(name : String, damage : Int)
+getName() : String
+getDamage() : Int

0..*   1

1..*

mansion

MansionImpl

-numberOfRows : Int
-numberOfColumns : Int
-name : String
-numberOfRooms : Int
-numberOfItems : Int
-listOfRooms : List<Room>
-target : Target
-mansionMap : List<Map<Room[]>>

+MansionImpl(mansionBuilder : MansionBuilder)
+moveTarget()
+createGraphicalRespresentation() : BufferedImage
+getNeighborForRoom(roomName : String) : List<Room>
+getName() : String
+getDetailsOfRoom(roomName : String) : String

MansionBuilder

-numberOfRows : Int
-numberOfColumns : Int
-name : String
-numberOfRooms : Int
-numberOfItems : Int
-listOfRooms : List<Room>
-target : Target
-mansionMap : List<Map<Room[]>>

+MansionBuilder()
-setName(name : String)
-setNumberOfRows(numberOfRows : Int)
-setNumberOfColumns(numberOfColumns : Int)
-setNumberOfRooms(numberOfRooms : int)
-setNumberOfItems(numberOfItems : int)
-createRooms() : List<Room>
-createItems() : Map<List<Item>>
-setTarget(name : String, hp : int)
-createMap()
-addItemsToRoom()
+readConfig(FileReader fr) : Mansion

0..*   1

1   1

<<Interface>>
Mansion

+getDetailsOfRoom()
+moveTarget()
+createGraphicalRepresentation()
+getNeighborForRoom()

1

1

target

Target

-name : String
-location : int
-hp : Int

+Target(name : String, hp : Int)
+getName() : String
+getHP() : Int
+getLocation() : Int

Test Cases			
TestName	**Method name**	**Input**	**Expected Output**
BuilderTest			
Mansion with no name	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with invalid size	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with invalid target health	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with no target name	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with invalid room number	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with invalid room size	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with no room name	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with invalid items number	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with invalid item damage	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with invalid room location	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with no item name	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with no rooms	readConfig()	Mansion File text	IllegalStateException Should be thrown
Mansion with no items	readConfig()	Mansion File text	IllegalStateException Should be thrown
A correct Mansion file provided	readConfig()	Mansion File text	Mansion object is created
Mansion			
Move a target	moveTarget()	NA	Target location should change by 1
Create a Graphical Representation	createGraphicalRepresetation()	NA	Buffered Image represenatation of Mansion should be created
Get Neighbors for a Room with invalid name	getNeighborForRoom()	name of a room that does not exist	IllegalArgumentException should be thrown
Get Neighbors for a Room	getNeighborForRoom()	name of a room that exists	Neighbors of the specied room are returned in a list (Eg: 1,6,7)
Get Name of the Mansion	getName()	NA	Name of the mansion is returned (Eg: Lucky's Mansion)
Get Details of the Room with invalid room name	getDetailsOfRoom()	name of a room that does not exist	IllegalArgumentException should be thrown
Get Details of the Room	getDetailsOfRoom()	name of a room that exists	Details of the specied room are returned as a String
Room			
Get Name of Room	getName()	NA	The name of the Room (Eg: Armoury Room)
Get Location of a Room	getLocation()	NA	Location coordinates of a Room (Eg: 12,25,17,30)
Get Items in a Room	getItems()	NA	Items in the room as a list (Eg : Mace, Pen)
Item			
Get item name	getName()	NA	Name of Item (Eg: Mace)
Get item damage	getDamage()	NA	Damage done by the item (Eg: 10)
Target			
Get Target Name	getName()	NA	Name of the Target (Eg: Lucky)
Get Target Health	getHP()	NA	Current Health of the target (Eg: 50)
Get Target Location	getLocation()	NA	Current Location of the Target (Eg: 2)