



List of essential tests, omitted some trivial tests				
Class	Method	Testing For	Test Scenario / Input	Expected Scenario / Output
RandomHelper	getInt	can generate random number of certain range	loop for large amount of times, test if the output falls within range	Numbers of the desired range
	getFourDicesValue	can generate value of four dices	loop for many times to see if value falls in range	
	assignItems	can shuffle the list return certain number of items	get a list of items and capacity	
Player	Getter	getter for all attributes	valid inputs	no exception thrown, and equals to the initiated variables
	Setter	setter for illegal inputs	value not in Ability	IllegalArgumentException
			value Negative / out of bound	IllegalArgumentException
			value of NULL	IllegalArgumentException
	isAlive	test if get correct value for health	health <= 0	FALSE
			health > 0	TRUE
	getAbility	test if get value of all abilities in list	random assign values for abilities	list of correct values for different abilities
			after we change the ability	
	attack	can compute correct damage and update	damage influence both players	both players receive damage
			damage unsuccessful	player health didn't change
			damage successful	player health updated
	pickUp	can add weapons to List	Pair swords	num = 2
Equipment	getAffect	can work with RandomHelper to set attributes	init udner randomHelper	can set affect for respective ability, others set to 0
			during running of model	can be updated when game goes on
			have negative effect on holder	can decrease the ability of user when equipment is used
			be dropped	when dropped, no effect on users any more
	getCapacity	can return value of max num	for different size of equipments	can return corresponding value
Weapon	getDamage	can calculate the damage correctly	weapon influenced by abilities	can take account the abilities
			weapon damage both players	can return values for both players
			weapon unable to damage	can set damage to 0 for player
			normal attack	can get the correct damage with randomness in account
BattleGame	getTurn	can tell next player	when game is not done	can return correct player
			when game is done	NULL
	isOver	can tell if game ends	when one of player dies	TRUE
			when no one dies	FALSE

	getWinner	can tell result	when ties	NULL
			when someone wins	can return the player
	assignValues	can init players' abilities	when game starts	can set value of desired ranges with randomHelper
	geneEquipBag	can generate equipment bags for players		can generate bag of equipments of correct state
	nextAttack	can operate players' attack and check if game ends	when game continues	can operate next attack from correct players
	toString	can print the description of the game's state	when game continues	can print correct descriptions and tell results if ended