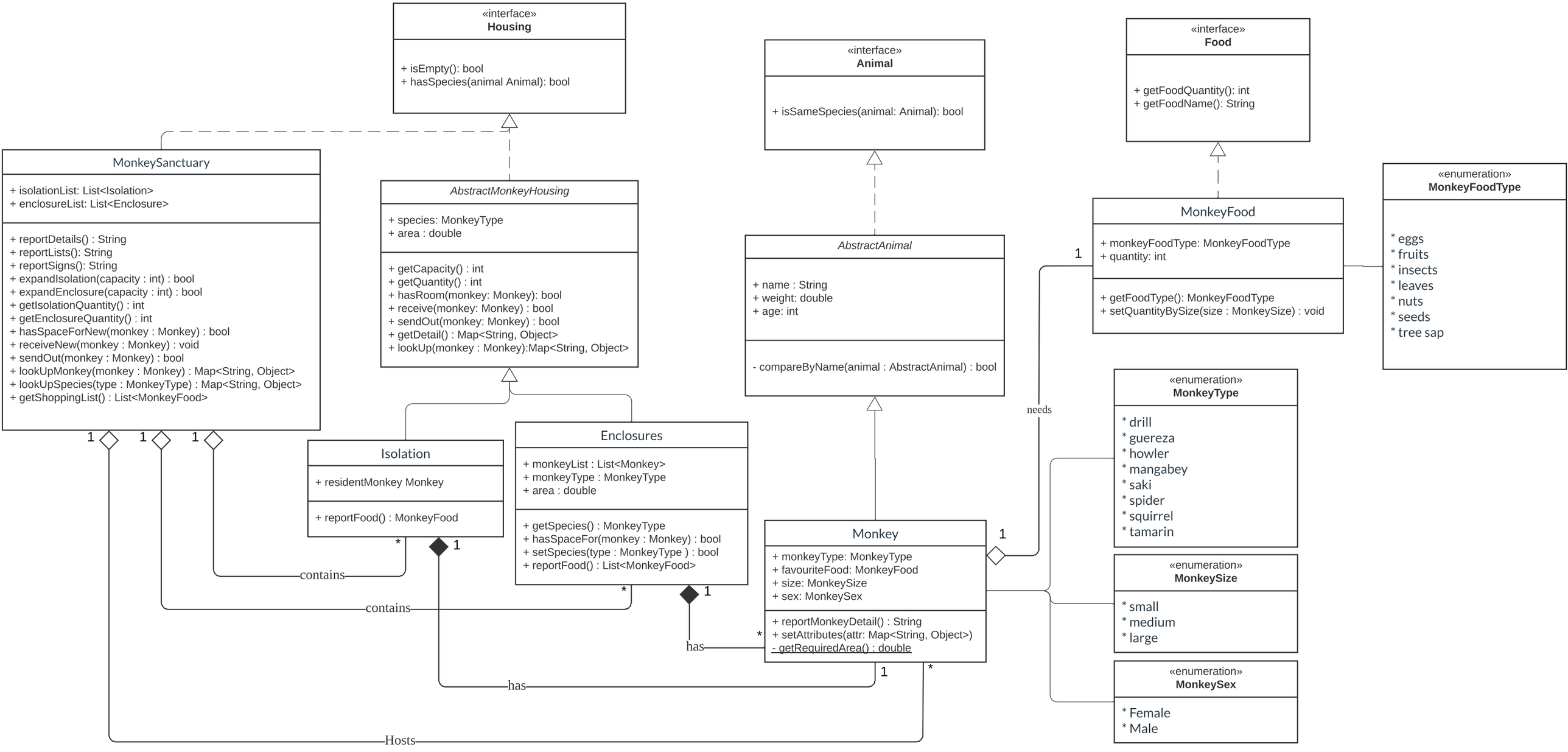
Sanctuary UML class Template

Towya | September 26, 2022



List of essential tests, omitted some trivial tests				
Class	**Method**	**Testing For**	**Test Scenario / Input**	**Expected Scenario / Output**
MonkeyFood	Constructor	construct with illegal enum	value not in FoodType	IllegalArgumentException
		construct with legal enum (compare to getFoodType())	FoodType.eggs	0
Monkey	Getter	getter for all attributes	valid inputs	no exception thrown, and equals to the initiated variables
	Setter	setter for illegal inputs	value not in MonkeyType	IllegalArgumentException
			value not in MonkeySex	IllegalArgumentException
			value not in MonkeySize	IllegalArgumentException
	compareByName	compare with monkey of same name	Monkey({same name})	TRUE
		compare with monkey of different name	Monkey({other name})	FALSE
Isolation	getCapacity	should always be 1		1
	hasRoom	return false when it's occupied	has a monkey (not null)	FALSE
		return true when it's not occupied	monkey is NULL	TRUE
	receive	no exception thrown when has room for new monkey	monkey is NULL	no exception thrown. The monkey is set to be the received one
		throw exception thrown when occupied	monkey is not NULL	IllegalStateException
	sendOut	monkey set to NULL after it's sent out		monkey became NULL
	getDetail	get correct details of monkey in this isolation	monkey is not NULL	print correct details of the resident monkey's attributes
	hasSpecies	when exists monkey of same species, return true	monkey of contained species	TRUE
		when monkey of non-existent species, return false	monkey of not contained species	FALSE
Enclosures	getCapacity	should be the size of rooms		#{size of rooms}
	hasRoom	return false when all rooms are occupied, or monkey type unmatched	#{monkey} = #{room} or other type of monkey	FALSE
		return true when still has room	#{monkey} < #{room} and same monkey type	TRUE
	receive	no exception thrown when has room for new monkey	has room for new and same type	no exception thrown. The monkey is set to empty room

		throw exception when no rooms	no more room, or unmatched type	IllegalStateException
	sendOut	size changes		monkey dropped from list
	getDetail	get correct details of monkeys in this isolation		print correct details of the resident monkeys' attributes
	lookUp	find the desired monkey's location	has this monkey	print its details
			does not have this monkey	print 'NOT found' message
MonkeySanctuary	expandIsolation	enlarge the isolation quantity	larger number of current size	List of isolation set to the new size
			smaller number than current size	IllegalArgumentException
	expandEnclosure	enlarge the enclosure quantity	larger number of current size	List of Enclosure set to the new size
			smaller number than current size	IllegalArgumentException
	getShoppingList	report the desired foods in a list		Match the monkeys' favourites, from both the enclosure and the isolation
	hasSpaceForNew	search for the isolation to find empty room	monkey of different types	TRUE
		when rooms unavailable, return false		FALSE
	receiveNew	when rooms unavailable, throw exception	rooms are full, new monkey coming in	IllegalStateException
	lookUpMonkey	iterate through enclosures and isolations to find the desired monkey	has desired monkey in both enclosures and isolations	the returned detail map contains information of all those desired monkeys, no one missed
		when there is no such monkey, return not found message	monkey that does not exist in the sancturary	not found message
	sendOut	when the input monkey exists, drop it from the housing	monkey that exists in the sancturary	dropped from its housing, leaving a new available room
		when the input monkey does not exist, thorw exception	non-existent monkey	IllegalArgumentException

	getShoppingList	return the list of desired favourite foods	when the sancturary is not empty and the resident monkeys have desired food	return the List containing food from both enclosures and isolations, no one left out