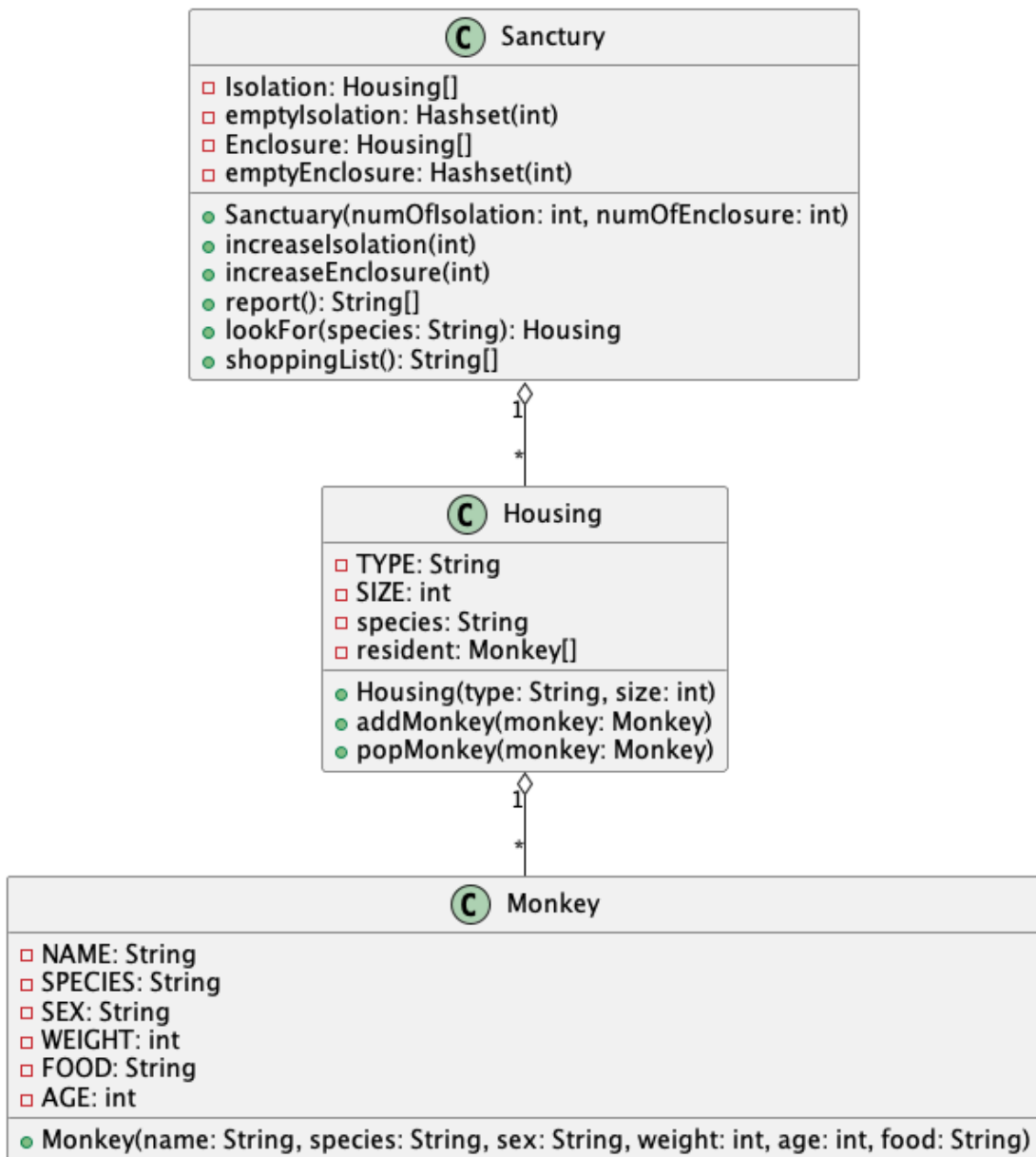


1. ORIGINAL DESIGN



2. MONKEY

Testing construction	Input	
Constructor with invalid name	Monkey("", "drill", "Male", 80, 12, "eggs")	IllegalArg
Constructor with invalid species	Monkey("good", "ok", "Male", 80, 12, "eggs")	IllegalArg
Constructor with invalid sex	Monkey("good", "drill", "ok", 80, 12, "eggs")	IllegalArg
Constructor with invalid weight	Monkey("good", "drill", "Male", -80, 12, "eggs")	IllegalArg
Constructor with invalid age	Monkey("good", "drill", "Male", 80, -12, "eggs")	IllegalArg
Constructor with invalid food	Monkey("good", "drill", "Male", 80, 12, "apples")	IllegalArg
Constructor with valid input	Monkey("good", "drill", "Male", 80, 12, "eggs")	

3. HOUSING

Testing construction	Input	Expected
Constructor with invalid type	Housing("ok", 10)	IllegalArgumentException
Constructor with invalid size	Housing("Isolation", -1)	IllegalArgumentException
Constructor with valid type and size	Housing("Isolation", 10)	None
Constructor with valid type and size	Housing("Enclosure", 10)	None

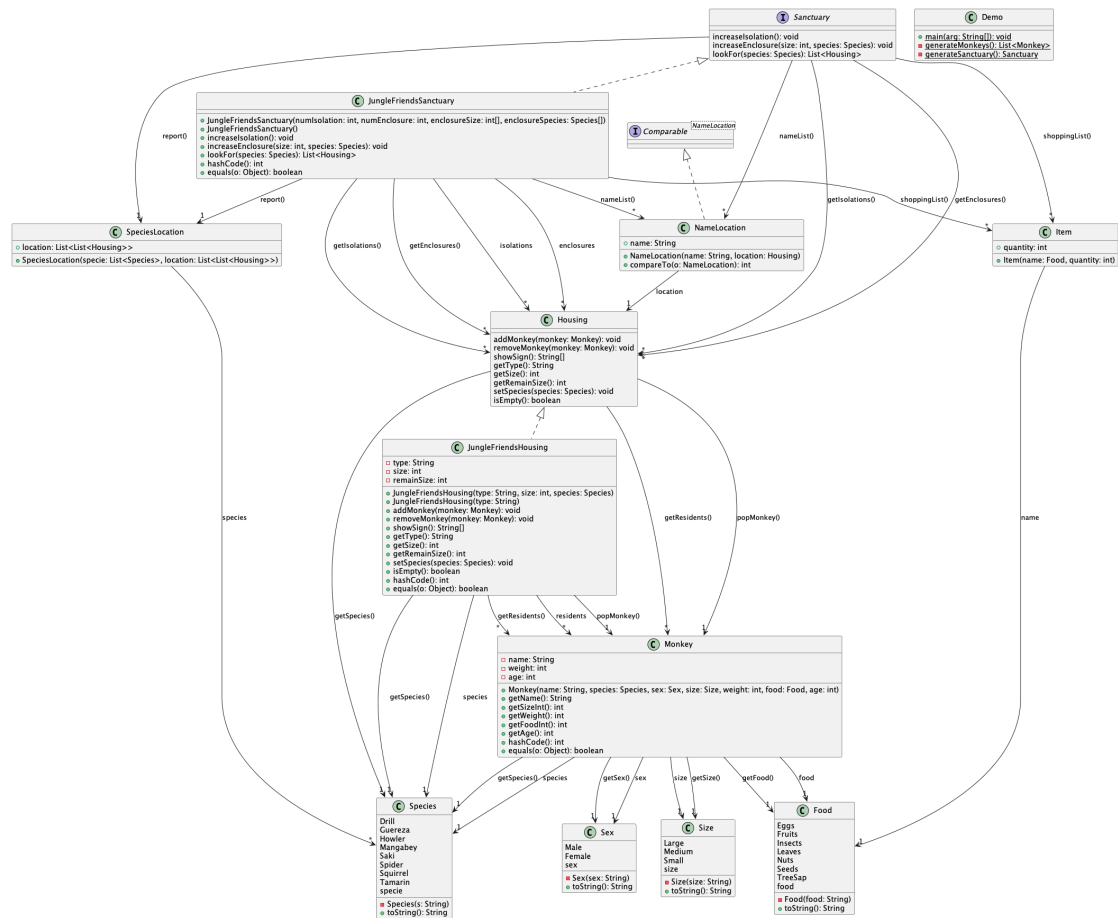
Testing addMonkey()	Input	Expected
Add a monkey	addMonkey(monkeyA)	None

Testing popMonkey()	Input	Expected
Pop a monkey	popMonkey(monkeyA)	None

4. SANCTUARY

Testing construction	Input	Expected
Constructor with invalid input	Sanctuary(-1, 1)	IllegalArgumentException
Constructor with invalid input	Sanctuary(1, -1)	IllegalArgumentException
Constructor with valid input	Sanctuary(1, 1)	None
Testing increaseIsolation()	Input	Expected
increase one each time	increaseIsolation()	None
Testing increaseEnclosure()	Input	Expected
increase one each time	increaseEnclosure()	None
Testing report()	Input	Expected
report species housed	report()	String[] in alphabetical order
Testing lookFor()	Input	Expected
look up for a special species	lookFor("species name")	a Housing object
Testing shoppingList()	Input	Expected
print a list for favorite food	shoppingList()	String[]

5. FINAL DESIGN



6. MONKEY

Testing construction

Constructor with invalid name
 Constructor with invalid species
 Constructor with invalid sex

Input

Monkey("", Species.Drill, Sex.Female, Size.Large, 10, Food.
 Monkey("name", Species.Drill, Sex.Female, Size.Large, 0, Food.
 Monkey("name", Species.Drill, Sex.Female, Size.Large, 10, Food.

7. HOUSING

Method Tested	Input
Constructor with invalid type	JungleFriendsHousing("any", 1, Species.Drill);
Constructor with incompatible type and species	JungleFriendsHousing("Isolation", 1, Species.Drill);
Constructor with incompatible type	JungleFriendsHousing("Enclosure");
Constructor with incompatible type and species	JungleFriendsHousing("Enclosure", -1, Species.Drill)
testAddMonkey	tmp.addMonkey(this.monkey);
testRemoveMonkey	this.iso.removeMonkey(this.monkey);
testPopMonkey	if the popped is equal to the entered
testPopMonkey	It's empty after pop
testShowSign	Test on empty house
testShowSign	Test on nonempty house

8. SANCTUARY

Method Tested	Input	
Constructor with invalid input	Negative number of isolation	IllegalArgument
Constructor with invalid input	Negative number of enclosure	IllegalArgument
Constructor with invalid input	Incompatible enclosure size species length	IllegalArgument
testIncreaseIsolation	tmp.increaseIsolation();	
testIncreaseEnclosure	tmp.increaseEnclosure(5, Species.Drill);	
testIncreaseEnclosure	tmp.increaseEnclosure(-1, Species.Drill);	IllegalArgument
testReport	SpeciesLocation report = tmp.report();	A SpeciesLocati
testLookFor	tmp.lookFor(this.monkey.getSpecies()).get(0)	List of
testShoppingList	List<Item> shopList = tmp.shoppingList();	List of It