Primates | UML and Testing Plan

UML

Sanctuary

- IsolationList: Isolation
- EnclosureList: Enclosure

+ getSpeciesListInHousing(): ArrayList<Primate>
+ getEnclosureSign(): ArrayList<Primate>
+ LookUpSpeciesHousingDetails(): String
+ getAlphabeticalList(): String[]
+ getShoppingList(): HashMap<String, int>
+ enterInputValues(int, int)
+ newPrimateCreation()

<<Interface>> Housing

+ checkAvailability(): int
+ checkMedicalAttention: boolean
+ addAnimal()
+ remAnimal()

AbstractHousing

+ checkMedicalAttention(): boolean

Enclosure

- enclosureSize:int = m
- ListEnclosure: ArrayList<ArrayList<Primate>>

+ addAnimal()
+ checkAvailability(): int
+ remAnimal()
+ getEnclosureList(): ArrayList<ArrayList<Primate>>

Isolation

- isolationCage: int = n
- ListIsolation: ArrayList<Primate>

+ addAnimal()
+ checkAvailability(): int
+ remAnimal()
+ getIsolationDetail(): ArrayList<Primate>

Primate Interface

+ createNewPrimate()
+ getName(): String
+ getSize(): int
+ getSex(): String
+ getAge(): int
+ getWeight(): double
+ getFavFood(): String
+ getFoodQty(): int
+ getSpecies(): String
+ getSpaceReq(): int

<<Enumeration>> Food

EGGS
FRUITS
INSECTS
LEAVES
NUTS
SEEDS
TREE SAP

<<Enumeration>> SIZE

SMALL, 1
MEDIUM, 5
LARGE, 10

Monkey

- name: string
- size: int
- sex: String
- age: int
- weight: age
- faveFood: String
- spaceReq: int

+ createNewPrimate()
+ getName(): String
+ getSize(): int
+ getSex(): String
+ getAge(): int
+ getWeight(): double
+ getFavFood(): String
+ getFoodRequirement(): int
+ getSpaceReq(): int

Drill

species: String

+ getSpecies():

Howler

species: String

+ getSpecies()

Saki

species: String

+ getSpecies()

Tamarin

species: String

+ getSpecies()

Mangabey

species: String

+ getSpecies()

Guereza

species: String

+ getSpecies()

Spider

species: String

+ getSpecies()

Squirrel

species: String

+ getSpecies()
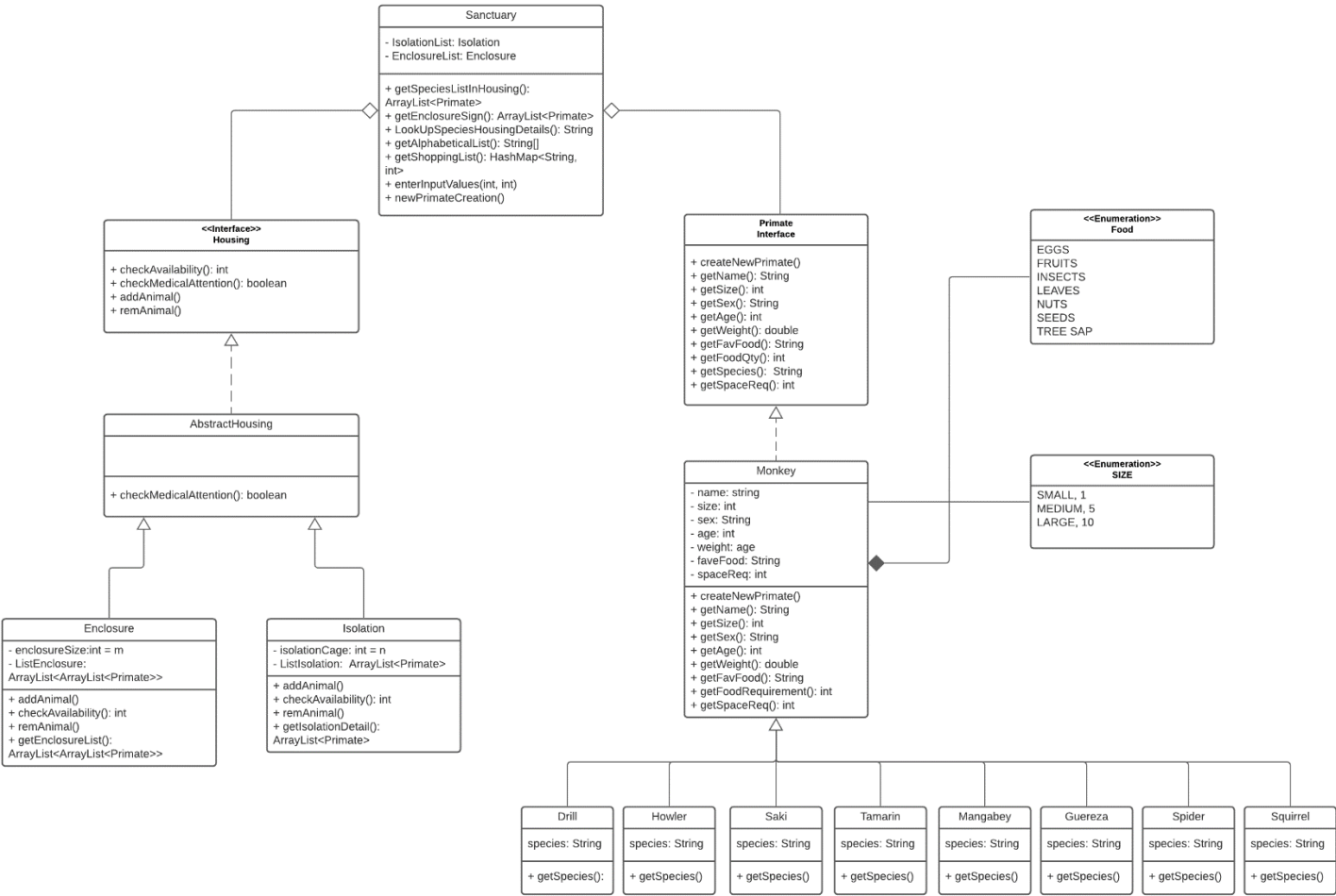
Testing Plan

Primate

1. Correct creation of new primate object along with species name
2. Testing getters of all attributes in Primate class to check if they return the correct passed value
3. Testing the Enumeration link between Primate and Food by checking if for a monkey the Primate returns correct favorite food
4. Testing the size and food requirement scenario, depending on the size of the monkey, correct food requirement should be passed
5. Testing if based on size correct area occupied by the monkey is calculated

Housing

1. If a monkey is added to the enclosure, it should be added to the enclosure of the same species. If the species has no enclosure yet, a new enclosure should be created
2. Testing to check if upon requirement of medical attention, a monkey is correctly shifted to isolation
3. It should be easy to add and remove a monkey from the enclosure or isolation and obtain total no of enclosure list
4. Testing to check if the checkAvailability() method correctly return the no of remaining enclosures and isolations respectively
5. Test to check if we can easily retrieve total no of enclosures and isolations
6. If a monkey is added to the isolation, it should be added to the isolation of the current list
7. If a monkey is currently in isolation but its health is fine now it should be shifted correctly to enclosure

Sanctuary

1. Sanctuary class should be able to create a new primate object, check the space in isolation and move it to isolation if required.
2. Sanctuary class should be able to provide the list of all the monkeys housing in isolation and enclosure
3. Sanctuary class should be able to provide the list of all the species housing in isolation and enclosure
4. Sanctuary class should be able to generate a shopping list according to all the different sizes of animals and their food requirements of the Monkeys currently housing in the isolation and enclosure
5. Test to see if it returns correct sign for any monkey in the enclosure along with its name, sex and favorite food
6. If a particular species is not housed, sanctuary should report the fact
7. It should return where a particular species is currently being housed
8. The total isolation and total enclosure should be able to be increased easily
9. Should correctly be able to get the no of enclosure and isolations from the user
10. Should throw exceptions wherever required (e.g., negative values for isolation and enclosures)