<<class>> Monkey

- name: String
- species: String
- sex: char
- weight: double
- age: int
- favFood: enum
- receivedMedicalAttension: boolean

+ addMonkeyToIsolation(): void

<<enumerration>> favFood

EGGS
FRUITS
INSECTS
LEAVES
NUTS
SEEDS
TREE_SAP

<>AbstractSanctuary

- n: int
- m: int
- isolation[]: Isolation
- enclosure[]: Enclosure

+ allMonkeys(): name, location
+ speciesHoused(): species, location
+ lookupSpecies():location
+ createShoppinglist(): array

<class>>Isolation

- isolationId: int (static)
- monkey: Monkey

+ allMonkeys(): name, location
+ speciesHoused(): species, location
+ lookupSpecies():location
+ createShoppinglist(): array

+ updateReceievedMedicalAttension(): void
+ getReceived MedicalAttension(): void
+ addMonkeytoEnclosure(): void

<<class>>Enclosure

- enclosureId: int (static)
- species: String
- area: double
- consumedArea: double
- animalCount: int
- enclosureList[]: Monkey

+ allMonkeys(): name, location
+ speciesHoused(): species, location
+ lookupSpecies():location
+ createShoppinglist(): array

+ checkCapacityToAccomodate(): int

TESTING PLAN:

1. Monkey class test: use constructor to validate the inputs and create an object only when all entries are valid.
2. Sanctuary class: creates an array of objects for Isolation & Enclosure. This can be tested by comparing it with the object created by passing the values to the constructors of these classes.
3. Expected functionalities like the species housed, looking up species and creating the shopping list can be implemented separately for Isolation and Enclosures and their output can be checked in the abstract class which will add the results of both the outputs.