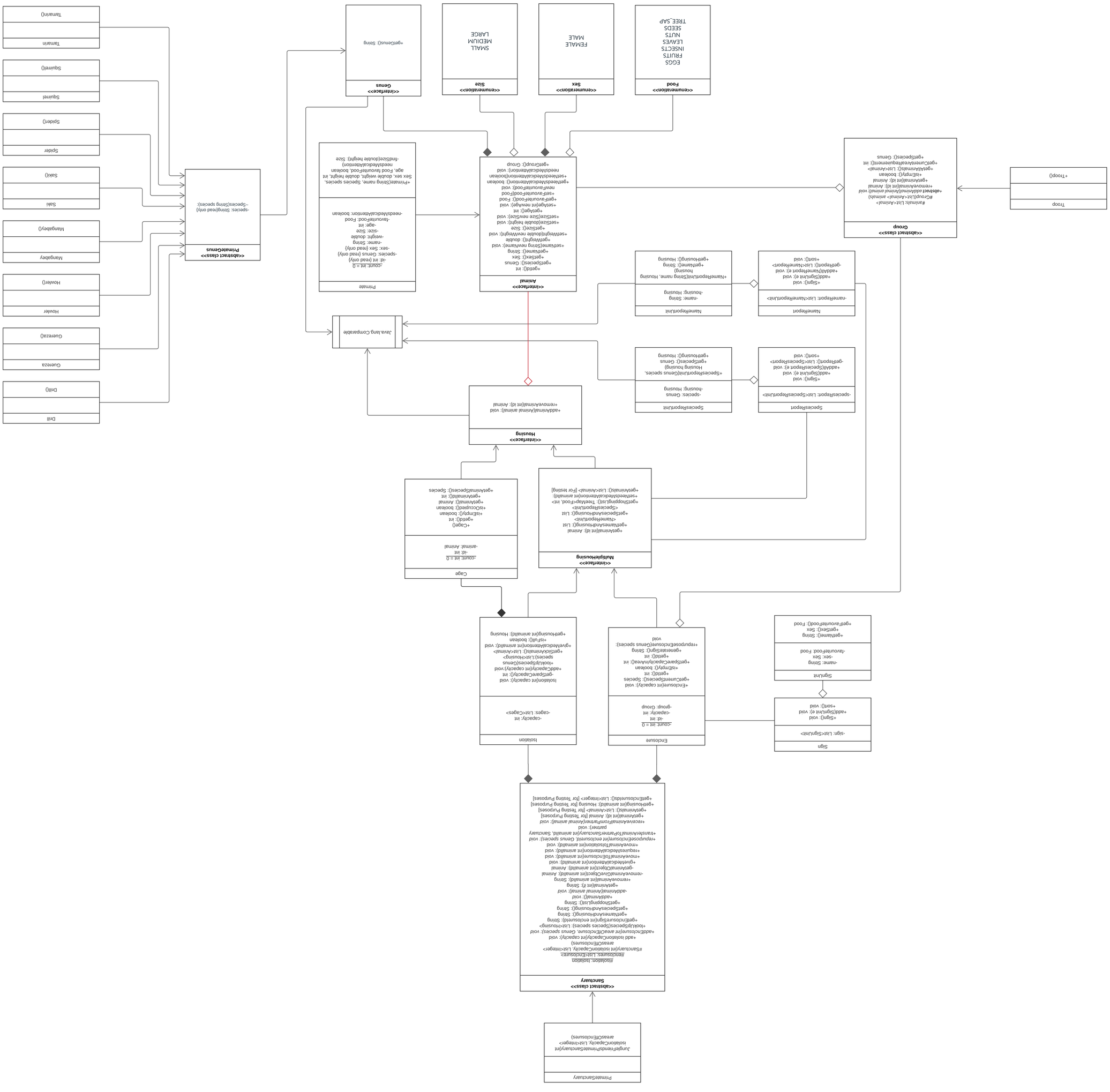
Drill
Drill()
Guereza
Guereza()
Howler
Howler()
Mangabey
Mangabey()
Saki
Saki()
Spider
Spider()
Squirrel
Squirrel()
Tamarin
Tamarin()
<<abstract class>>
PrimateGenus
-species: String{read only}
~Species(String species):
+getGenus(): String
<<interface>>
Genus
<<enumeration>>
Size
SMALL
MEDIUM
LARGE
<<enumeration>>
Sex
FEMALE
MALE
<<enumeration>>
Food
EGGS
FRUITS
INSECTS
LEAVES
NUTS
SEEDS
TREE_SAP
Primate
-count: int = 0
-id: int{read only}
-sex: Sex{read only}
-name: String
-weight: double
-size: Size
-age: int
-favouriteFood: Food
-needsMedicalAttention: boolean
+Primate(String name, Species species,
Sex sex, double weight, double height, int
age, Food favouriteFood, boolean
needsMedicalAttention)
-findSize(double height): Size
<<interface>>
Animal
+getId(): int
+getSpecies(): Genus
+getName(): String
+getSex(): Sex
+getWeight(): double
+setName(String newName): void
+setWeight(double newWeight): void
+getSize(): Size
+setSize(double height): void
+setSize(Size newSize): void
+getAge(): int
+setAge(int newAge): void
+getFavouriteFood(): Food
+setFavouriteFood(Food
newFavouriteFood): void
+getNeedsMedicalAttention(): boolean
+setNeedsMedicalAttention(boolean
needsMedicalAttention): void
+getGroupId(): Group
<<abstract class>>
Group
#animals: List<Animal>
#Group(List<Animal> animals)
+abstract addAnimal(Animal animal): void
+removeAnimal(int id): Animal
+getAnimal(int id): Animal
+getAnimals(): List<Animal>
+getAnimalsAt(int): Animal
+isEmpty(): boolean
+getCurrentLiveRequirement(): int
+getSpecies(): Genus
Troop
+Troop()
<<interface>>
Housing
+addAnimal(Animal animal): void
+removeAnimal(int id): Animal
java.lang.Comparable
SpeciesReport
-speciesReport: List<SpeciesReportUnit>
+Sign(): void
+addAll(SpeciesReport e): void
+sort(): void
SpeciesReportUnit
-species: Genus
-housing: Housing
+SpeciesReportUnit(Genus species,
Housing housing)
+getSpecies(): Genus
+getHousing(): Housing
NameReport
-nameReport: List<NameReportUnit>
+Sign(): void
+addAll(NameReport e): void
+getReport(): List<NameReport>
+sort(): void
NameReportUnit
-name: String
-housing: Housing
+NameReportUnit(String name, Housing
housing)
+getName(): String
+getHousing(): Housing
<<abstract class>>
Sanctuary
#isolation: Isolation
#enclosures: List<Enclosure>
#Sanctuary(int isolationCapacity, List<Integer>
areasOfEnclosures)
+addIsolationCapacity(int capacity): void
+addEnclosure(int areaOfEnclosure, Genus species): void
+getIsolationCapacity(): int
+getEnclosureSign(int enclosureId): String
+getNamesAndHousing(): String
+getSingleHousing(): String
+getSpecialsAndHousing(): String
+getAnimal(): void
+addAnimal(): void
-addAnimal(Animal animal): void
+getAnimal(int id): String
+removeAnimal(int animalId): String
-getAnimalObject(int animalId): Animal
-removeAnimalGiveObject(int animalId): Animal
+giveMedicalAttention(int animalId): void
+requestMedicalAttention(int animalId): void
+moveToEnclosure(int animalId): void
+moveAnimalToIsolation(int animalId): void
+repurposeEnclosure(int enclosureId, Genus species): void
+transferAnimalToPartnerSanctuary(int animalId, Sanctuary
partner): void
+receiveAnimalFromPartner(Animal animal): void
+getAnimal(int id): Animal [for Testing Purposes]
+getAnimals(): List<Animal> [for Testing Purposes]
+getHousing(int animalId): housing [for Testing Purposes]
+getEnclosureIds(): List<Integer> [for Testing Purposes]
Enclosure
-count: int = 0
-id: int
-capacity: int
-group: Group
+Enclosure(int capacity): void
+getCurrentSpecies(): Species
+isEmpty(): boolean
+getSpareCapacity/nAreal(): int
+getGenerateSign(): String
+repurposeEnclosure(Genus species):
void
Sign
-sign: List<SignUnit>
+Sign(): void
+addAll(SignUnit e): void
+sort(): void
SignUnit
-name: String
-sex: Sex
-favouriteFood: Food
+getName(): String
+getSex(): Sex
+getFavouriteFood(): Food
Isolation
-capacity: int
-cages: List<Cages>
isolation(int capacity): void
-getSpareCapacity(): int
+addCapacity(int capacity):void
+bookUpSpecies(Genus
species):List<Housing>
+getSickAnimals(): List<Animal>
+giveMedicalAttention(int animalId): void
+getHousing(int animalId): Housing
+isFull(): boolean
Cage
-count: int = 0
-id: int
-animal: Animal
+Cage()
+getId(): int
+isEmpty(): boolean
+isOccupied(): boolean
+getAnimal(): Animal
+getAnimalId(): int
+getAnimalsSpecies(): Species
<<interface>>
MultipleHousing
+getAnimal(int id): Animal
+getNamesAndHousing(): List
+getSpecieesAndHousing(): List
<SpeciesReportUnit>
<NameReportUnit>
+setNeedsMedicalAttention(int animalId):
void
+getSingleHousing(): TreeMap<Food, int>
+getAnimals(): List<Animal> [For testing]
JungleFriendsPrimateSanctuary(int
isolationCapacity, List<Integer>
areasOfEnclosures)
PrimateSanctuary