







## Species

- \_speciesID: String
- \_name: String
- \_animals: List<Animal>

+ addAnimal(Animal: Animal): void  
+ removeAnimal(AnimalID: String): void

## <<abstract>> Employee

- \_employeeID: String
- \_employeeName: String
- \_satisfaction: double

## Vaccine

- \_vaccineID: String
- \_name: String
- \_applicableSpecies: List<Species>
- \_applicationHistory: Map<Animal, Veterinarian>

+ calculateDamage(Animal: Animal): double

## Animal

- \_animalID: String
- \_name: String
- \_species: Species
- \_healthState: List<String>

- \_satisfaction: double  
+ calculateSatisfaction(Habitat: Habitat): double

## Caretakers

- \_habitats: List<Habitat>

+ calculateSatisfaction(Habitat: Habitat): double

## Veterinarian

- \_speciesHandled: List<Species>

+ calculateSatisfaction(Species: Species): double

## <<interface>>

### SatisfactionCalculator

+ calculateSatisfaction(): double

## Habitat

- \_habitatID: String
- \_habitatName: String
- \_area: int
- \_animals: List<Animal>
- \_adequacy: Map<Species, String>
- \_trees: List<Tree>

## Tree

- \_treeID: String
- \_name: String
- \_age: int
- \_type: TreeType
- \_season: Seasons
- \_cleaningDifficulty: int
- \_cleaningEffort: double

+ calculateCleaningEffort(): double

## <<enumeration>>

### TreeType

- + DECIDUOUS
- + EVERGREEN

<< enumeration >>
Season
SPRING
SUMMER
AUTUMN
WINTER
+ calculateBearingEffort(tree: Tree): double { override }
+ calculateBearingEffort(tree: Tree): double { override }
+ calculateBearingEffort(tree: Tree): double { override }
+ calculateBearingEffort(tree: Tree): double { override }

HotelManager
= _hotel: Hotel
= _filename: String
+ save(): void
+ saveAs(filename: String): void
+ load(filename: String): void
+ importFile(filename: String): void

Hotel
- _allSpecies: List<Species>
- _allAnimals: List<Animal>
- _allHabitats: List<Habitat>
- _allEmployees: List<Employee>
- _allVaccines: List<Vaccine>
- _allTrees: List<Tree>
+ createSpecies(id: String, name: String): void
+ deleteSpecies(id: String): void
+ createAnimal(id: String, name: String, SID: String, hID: String): void
+ deleteAnimal(id: String): void
+ createHabitat(id: String, name: String, area: int): void
+ deleteHabitat(id: String): void
+ addTreesToHabitat(id: String): void
+ createEmployee(id: String, name: String): void
+ deleteEmployee(id: String): void
+ createVaccine(id: String, name: String): void
+ deleteVaccine(id: String): void
+ addSpeciesToVaccine(id: String): void
+ createTree(id: String, name: String, age: int, difficulty: int, type: String): void
+ deleteTree(id: String): void
+ importFile(filename: String): void



João Teixeira (97226)

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do projeto

Afonso Calinas (102373)

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do projeto.