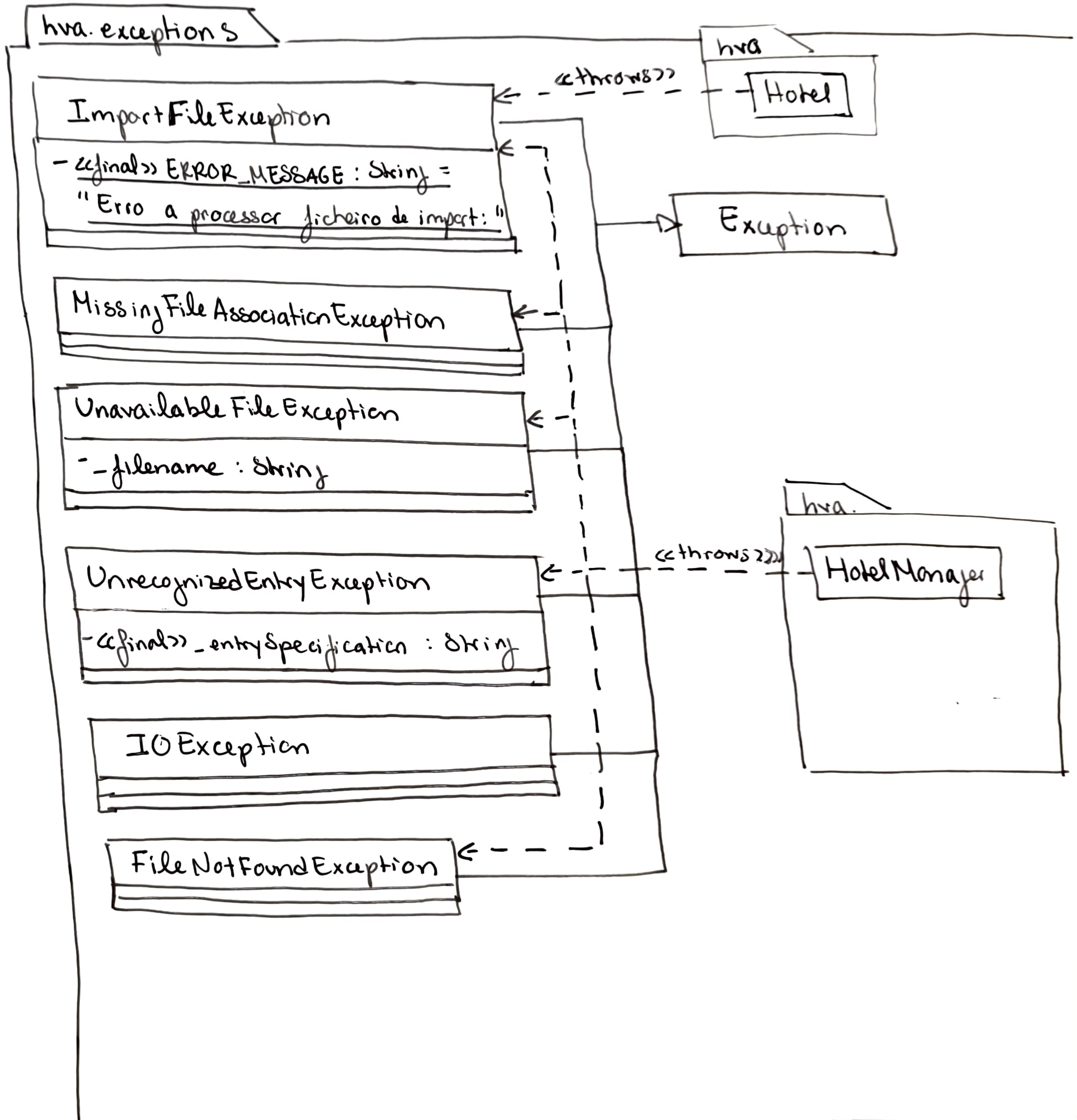
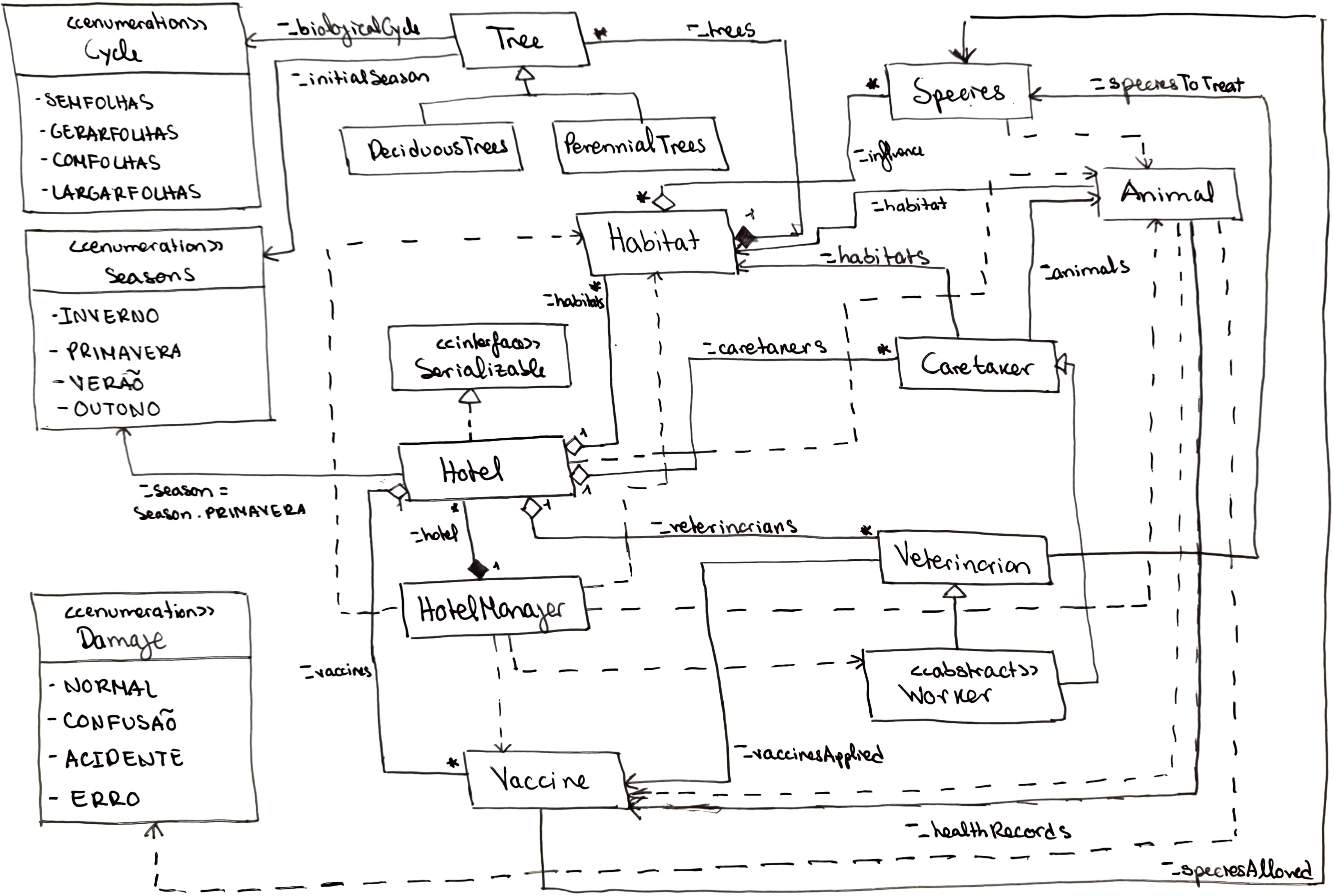
→Serial numbers for serialization have been omitted

→All the methods were defined separately in order to better understand the connections established between classes.

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


hva.exceptions
ImportFileException
- <<final>> ERROR_MESSAGE : String = "Erro a processar ficheiro de import:"
MissingFileAssociationException
UnavailableFileException
- _filename : String
UnrecognizedEntryException
- <<final>> _entrySpecification : String
IOException
FileNotFoundException
Exception
hva
Hotel
<<throws>>
hva.
HotelManager
<<throws>>



<<enumeration>>
Cycle
-SEMFOLHAS
-GERARFOLHAS
-COMFOLHAS
-LARGARFOLHAS
<<enumeration>>
Seasons
-INVERNO
-PRIMAVERA
-VERÃO
-OUTONO
<<enumeration>>
Damage
-NORMAL
-CONFUSÃO
-ACIDENTE
-ERRO
Tree
DeciduousTrees
PerennialTrees
-biologicalCycle
-initialSeason
-trees
*
Species
*
-speciesToTreat
-influence
Animal
-habitat
-animals
Habitat
*
1
-habitats
-habitats
*
<<interface>>
Serializable
-caretakers
*
Caretaker
Hotel
1
1
1
*
-season = Season.PRIMAVERA
-hotel
1
-veterinarians
*
Veterinarian
HotelManager
-vaccines
<<abstract>>
Worker
*
Vaccine
-vaccinesApplied
-healthRecords
-speciesAllowed

```
Vaccine
------------------------------------------------
-_id : String
-_name : String
-_speciesAllowed : Map <String, Species>
-_rejistryUsage : Map <Animal, Veterinarian>
------------------------------------------------
+ addSpecies (s : Species) : void
+ updateRejistry (a : Animal, v : Veterinarian):
                                          void
```

```
Veterinarian
------------------------------------------------
-_speciesToTreat : Map <String, Species>
-_vaccinesApplied : Vaccine[]
------------------------------------------------
+ addSpecies (s : Species) : void
+ calculateWorkloadEachSpecies () : double
+ applyVaccine (v : Vaccine, a : Animal) : void
+ calculateDamage (v:Vaccine, a : Animal) :
                                          void
+ viewVaccinations () : String
```

```
Hotel
------------------------------------------------
-_veterinarians : Map <String,
                  Veterinarian >
-_caretakers : Map <String,
                  CareTaker >
-_habitats : Map <String, Habitat>
-_vaccines : Map <String, Vaccine>
-_season : Seasons = Seasons.
                          PRIMAVERA
------------------------------------------------
+ addWorker (worker : Worker) : void
+ addAnimal (animal : Animal) : void
+ addHabitat (habitat : Habitat) : void
+ addVaccine (v : Vaccine) : void
+ transferAnimalHabitats (animalId :
    String ; h1 : String, h2 : String) : void
+ applyVaccine (v : Vaccine, vet :
    Veterinarian, a : Animal ) : void
+ calculateSatisfactionAnimal () : int
+ calculateSatisfactionVeterinarian () :
                                   double
+ calculateSatisfactionCaretaker () :
                                   double
+ treeCleaningEffort (treeId : String) :
                                   double
# importFile (filename : String) :
                                   void
```

```
HotelManager
------------------------------------------------
-_hotel : Hotel = new Hotel()
------------------------------------------------
+ save () : void
+ saveAs (filename : String) : void
+ load (filename : string) : void
+ importFile (filename : String) : void
+ addAnimalToHotel (animal : Animal) : void
+ addWorkerToHotel (worker : Worker) : void
+ addHabitatToHotel (habitat : Habitat) : void
+ addVaccineToHotel (vaccine : Vaccine) : void
+ addResponsabilityVet (workerId : String,
                   speciesId : String) : void
+ addResponsabilityCareT (workerId :
        String, habitatId : String) : void
+ removeResponsabilityVet (workerId :
        String, speciesId : String) : void
+ removeResponsabilityCareT (workerId :
        String, habitatId : String) : void
+ calculateSatisfactionHotel () : int
```

Animal

-_id : String
-_name : String
-_species : Species
-_healthRecords : Map <Vaccine, String>
-_habitat : Habitat

+ countSameSpecies () : int
+ countDiffSpecies () : int
+ calculateSatisfaction () : double
+ updateHealthRecords (v: Vaccine,
damage: Damage) : void
+ viewVaccinesApplied () : String
+ getInfluence () : int

CareTaker

-_habitats : Map < String, Habitat>
-_animals : Map < String, Animal>

+ calculateWorkloadInHabitat(): int
+ calculateWorkload () : double
+ updateHabitats(h: Habitat) : void
+ updateAnimals (a : Animal) : void

DeciduousTrees

- <<final>> _type : String = "CADUCA"
- _cleaningDifficulty : int
- _seasonEffect : int

+ calculateTotalCostCleaning () : int

Habitat

-_id : String
-_name : String
-_dimension : int
-_trees : Map <String, Tree>
-_animals : Map <String, Animal>
-_influence : Map <Species, Integer>

+ getPopulation () : int
+ calculateCorrespondence (animal:
Animal)
+ addTree (tree : Tree) : void
+ removeTree (tree : Tree) : void
+ getInfluence (species : Species) : int
+ viewTrees () : String
+ addAnimal (a : Animal) : void
+ addInfluence (i : int) : void

Perennial Trees

- <<final>> _type : String = "PERENE"
- _cleaningDifficulty : int
- _seasonEffect : int

+ calculateTotalCostCleaning () : int

Species

-_id : String
-_name : String
-_animals : String[]

+ printAnimals() : String
+ addAnimal (a : Animal) : void

Tree

-_id : String
-_name : String
-_age : int
-_initialSeason : Seasons
-_biologicalCycle : Cycle

+ updateAge () : void

<<abstract>> Worker

-_id : String
-_name : String

+ <<abstract>> calculateSatisfaction () : int