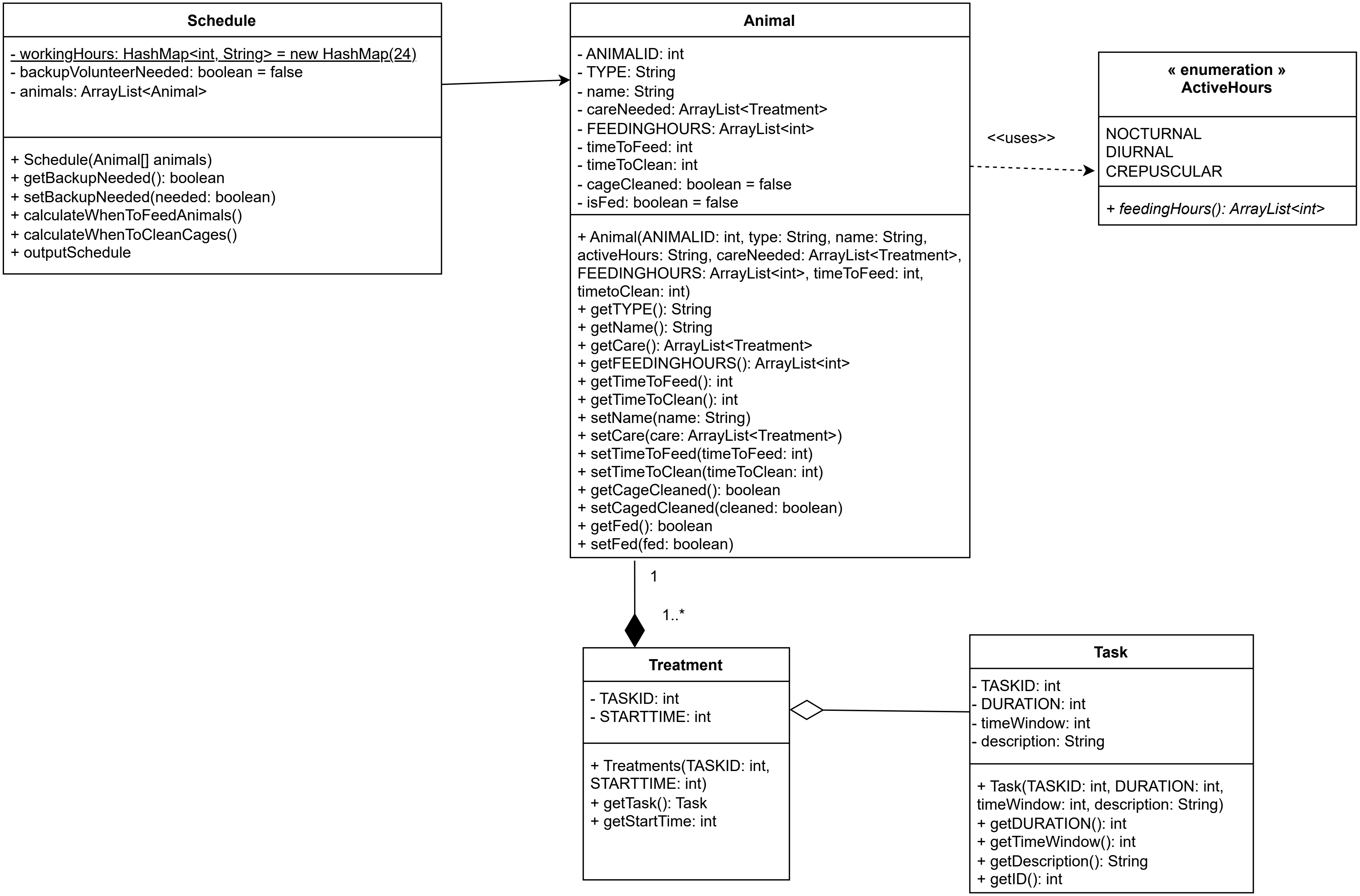

Schedule

- workingHours: HashMap<int, String> = new HashMap(24)
- backupVolunteerNeeded: boolean = false
- animals: ArrayList<Animal>

+ Schedule(Animal[] animals)
+ getBackupNeeded(): boolean
+ setBackupNeeded(needed: boolean)
+ calculateWhenToFeedAnimals()
+ calculateWhenToCleanCages()
+ outputSchedule

Animal

- ANIMALID: int
- TYPE: String
- name: String
- careNeeded: ArrayList<Treatment>
- FEEDINGHOURS: ArrayList<int>
- timeToFeed: int
- timeToClean: int
- cageCleaned: boolean = false
- isFed: boolean = false

+ Animal(ANIMALID: int, type: String, name: String, activeHours: String, careNeeded: ArrayList<Treatment>, FEEDINGHOURS: ArrayList<int>, timeToFeed: int, timetoClean: int)
+ getTYPE(): String
+ getName(): String
+ getCare(): ArrayList<Treatment>
+ getFEEDINGHOURS(): ArrayList<int>
+ getTimeToFeed(): int
+ getTimeToClean(): int
+ setName(name: String)
+ setCare(care: ArrayList<Treatment>)
+ setTimeToFeed(timeToFeed: int)
+ setTimeToClean(timeToClean: int)
+ getCageCleaned(): boolean
+ setCagedCleaned(cleaned: boolean)
+ getFed(): boolean
+ setFed(fed: boolean)

« enumeration »
ActiveHours

NOCTURNAL
DIURNAL
CREPUSCULAR

+ *feedingHours(): ArrayList<int>*

Treatment

- TASKID: int
- STARTTIME: int

+ Treatments(TASKID: int, STARTTIME: int)
+ getTask(): Task
+ getStartTime: int

Task

- TASKID: int
- DURATION: int
- timeWindow: int
- description: String

+ Task(TASKID: int, DURATION: int, timeWindow: int, description: String)
+ getDURATION(): int
+ getTimeWindow(): int
+ getDescription(): String
+ getID(): int