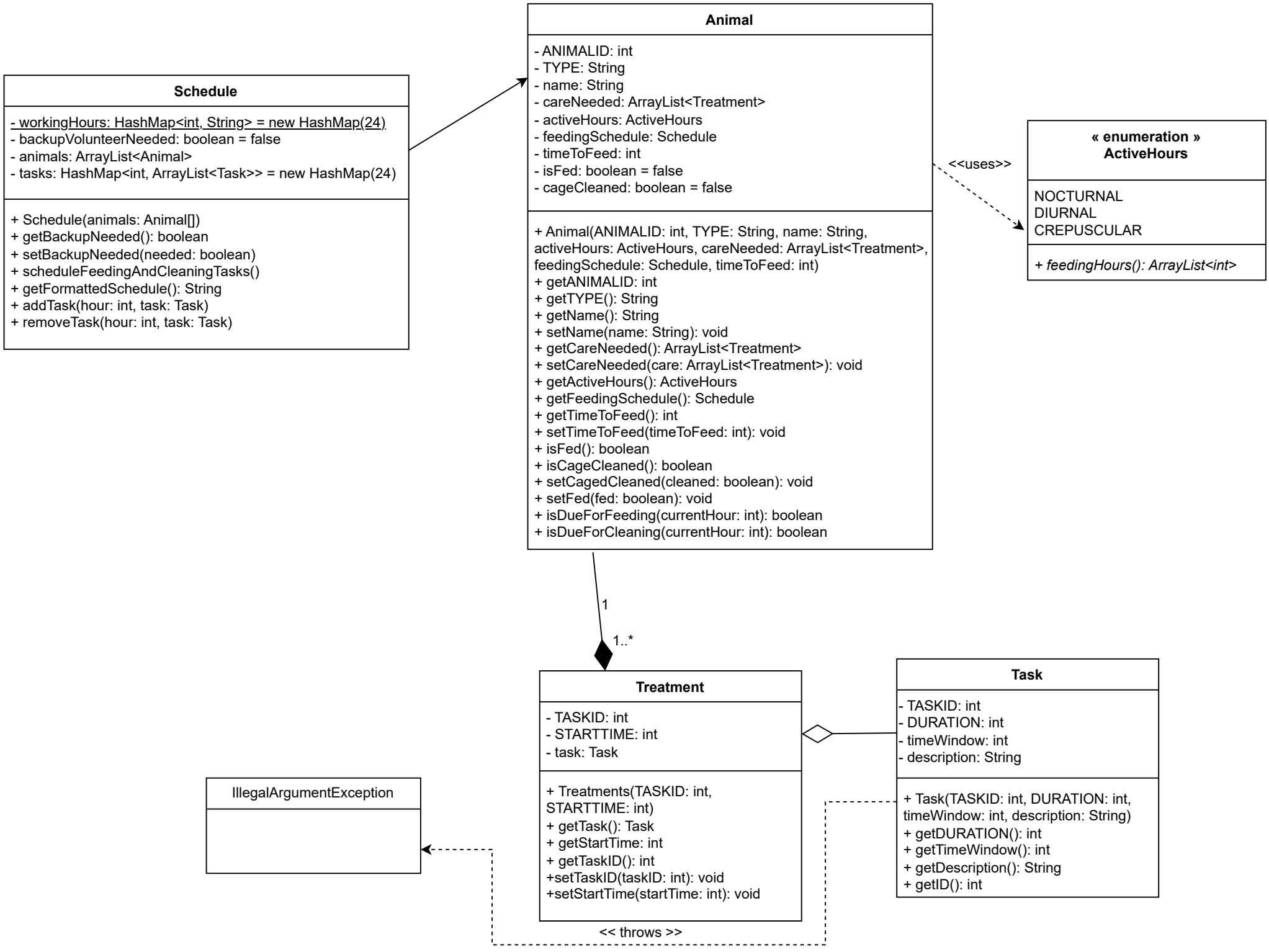
Schedule
- workingHours: HashMap<int, String> = new HashMap(24)
- backupVolunteerNeeded: boolean = false
- animals: ArrayList<Animal>
- tasks: HashMap<int, ArrayList<Task>> = new HashMap(24)
+ Schedule(animals: Animal[])
+ getBackupNeeded(): boolean
+ setBackupNeeded(needed: boolean)
+ scheduleFeedingAndCleaningTasks()
+ getFormattedSchedule(): String
+ addTask(hour: int, task: Task)
+ removeTask(hour: int, task: Task)
Animal
- ANIMALID: int
- TYPE: String
- name: String
- careNeeded: ArrayList<Treatment>
- activeHours: ActiveHours
- feedingSchedule: Schedule
- timeToFeed: int
- isFed: boolean = false
- cageCleaned: boolean = false
+ Animal(ANIMALID: int, TYPE: String, name: String, activeHours: ActiveHours, careNeeded: ArrayList<Treatment>, feedingSchedule: Schedule, timeToFeed: int)
+ getANIMALID: int
+ getTYPE(): String
+ getName(): String
+ setName(name: String): void
+ getCareNeeded(): ArrayList<Treatment>
+ setCareNeeded(care: ArrayList<Treatment>): void
+ getActiveHours(): ActiveHours
+ getFeedingSchedule(): Schedule
+ getTimeToFeed(): int
+ setTimeToFeed(timeToFeed: int): void
+ isFed(): boolean
+ isCageCleaned(): boolean
+ setCagedCleaned(cleaned: boolean): void
+ setFed(fed: boolean): void
+ isDueForFeeding(currentHour: int): boolean
+ isDueForCleaning(currentHour: int): boolean
<<uses>>
« enumeration »
ActiveHours
NOCTURNAL
DIURNAL
CREPUSCULAR
+ feedingHours(): ArrayList<int>
1
1..*
Treatment
- TASKID: int
- STARTTIME: int
- task: Task
+ Treatments(TASKID: int, STARTTIME: int)
+ getTask(): Task
+ getStartTime: int
+ getTaskID(): int
+setTaskID(taskID: int): void
+setStartTime(startTime: int): void
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
IllegalArgumentException
<< throws >>