IllegalArgumentException

<<throws>>
<<throws>>
<<throws>>
<<throws>>
<<throws>>
<<throws>>

MedicalTask

- TASK_ID: int

+ MedicalTask(description: String, duration: int, maxWindow:int, "Medical Task")
+ getTaskID(): int

Extends

Animal

- ANIMAL_SPECIES: String
- animalNickname: String
- ANIMAL_ID: int
- ACTIVITY_PATTERN: String

+ Animal(animalID : int, animalNickname : String, animalSpecies : String)
+ getAnimalSpecies(): String
+ getAnimalNickname(): String
+ setAnimalNickname(AnimalNickname : String) : void
+ getAnimalID(): int
+ getActivityPattern(): String

1...*   1

<<Uses>>

Treatment

- treatedAnimal: Animal
- task: Task

+ Treatment(task : MedicalTask, treatedAnimal : Animal)
+ getTreatedAnimal(): Animal
+ getTask(): Task

<< enumeration >>
AnimalInfo

COYOTE
PORCUPINE
FOX
RACCOON
BEAVER

Task

- backupVolunteerNeeded: boolean
- description: String
- duration: int
- maxWindow: int
- TASK_TYPE: String

+ Task(description : String, duration : int, maxWindow : int, taskType : String)
+ getDescription(): String
+ setDescription(description : String) : void
+ getDuration(): int
+ setDuration(duration : int) : void
+ getMaxWindow(): int
+ setMaxWindow(maxWindow : int) : void

0...*

1

<< Uses >>

Schedule

- treatmentTasks: ArrayList <Treatment>
- normalTasks: ArrayList <NormalTask>
- date: LocalDate
- scheduleResult: String
- day: int
- month: int
- year: int

+ Schedule(day : int, month : int, year : int)
+ getTreatmentList(): ArrayList<Treatment>
+ getDate() : LocalDate
+ setDate(day : int, month : int, year : int) : void
+ calculateEfficient(): String
+ getScheduleResult(): String
+ getDay(): int
+ setDay(day : int) : void
+ getMonth(): int
+ setMonth(month : int) : void
+ getYear(): int
+ setYear(year : int) : void
+ readSql(): void
- instantiateNormalTasks(): void

Extends

NormalTask

+ NormalTask(description: String, duration: int, maxWindow:int, "Normal Task")

0...*   1