WildlifeRescue

- animalsInCare: List<Animal>
- tasks: List<Task>
- backupVolunteerNeeded: boolean = false
- schedule: String

+ WildlifeRescue(animals: List<Animal>, tasks: List<Task>)
+ getTasks(): List<Task>
+ getAnimalsInCare(): List<Animal>
+ getBackupNeeded(): boolean
+ addTask(task: Task)
+ addAnimal(animal: Animal)
+ setBackupVolunteerNeeded(needed: boolean)
+ outputScheduleFIle(): void

1   1..*

Animal

- TYPE: String

- name: String

- careNeeded: List<Task>

- FEEDINGHOURS: List<int>

- timeToFeed: int
- timeToClean: int

+ Animal(type: String, name: String,
activeHours: String, careNeeded: List<Task>,
timeToFeed: int, timetoClean: int)

+ getTYPE(): String
+ getName(): String
+ getCare(): List<Task>
+ getFEEDINGHOURS(): List<int>
+ getTimeToFeed(): int
+ getTimeToClean(): int
+ setName(name: String)
+ setCare(care: List<Task>)
+ setTimeToFeed(timeToFeed: int)
+ setTimeToClean(timeToClean: int)

------Use----->

« enumeration »
ActiveHours

NOCTURNAL
DIURNAL
CREPUSCULAR

+ *feedingHours(): List<int>*

Task

- DURATION: int

- timeSlot: int

- possibleTimeSlots: List<int>

- FLEXIBLE: boolean

- description: String

+ Task(DURATION: int, timeSlot: int,
possibleTimeSlots: List<int>,
FLEXIBLE: boolean, description:
String)
+ Task(DURATION: int,
possibleTimeSlots: List<int>,
FLEXIBLE: boolean, description:
String)
+ setTimeSlot(timeSlot: int)

```
+ getDURATION(): int
+ getTimeSlot(): int
+ getPossibleTimeSlots(): List<int>
+ getFLEXIBLE(): boolean
+ getDescription(): String
```