«Enum» Type	«Enum» Food	Bird	«Enum» Food

Bird
- id: int
- name: String
- food: Food[]
- foodQuantity: int
- type: Type
- characteristic: String
- extinct: boolean
- wings: int

+ getId(): int
+ setId(): int
+ getName(): String
+ setName(): String
+ getFood(): Food[]
+ setFood(): Food[]
+getFoodQuantity(): int
+setFoodQuantity(): int
+ getType(): Type
+ setType(): Type
+ getCharacteristic(): String
+ setCharacteristic(): String
+ getExtinct(): boolean
+ setExtinct(): boolean
+ getWings(): int
+ setWings(): int

«Enum» Type
Hawks
Eagles
Osprey
Emus
Kiwis
Moas
Owls
Rose_ring_parakeet
Gray_parrot
Sulfur_crested_cockatoo
Pigeons
Great_auk
Horned_puffin
African_jacana
Ducks
Swans
Geese

«Enum» Food
Berries
seeds
fruit
insects
other_birds
eggs
small_mammals
fish
buds
larvae
aquatic_invertebrates
nuts
vegetation

«Enum» Food
wetlands
freshwater
saltwater_shorelands
ocean

Prey
- id: int
- name: String
- food: Food[]
- foodQuantity: int
- type: Type
- characteristic: String
- extinct: boolean
- wings: int

FlightlessBird
- id: int
- name: String
- food: Food[]
- foodQuantity: int
- type: Type
- characteristic: String
- extinct: boolean
- wings: int

Owl
- id: int
- name: String
- food: Food[]
- foodQuantity: int
- type: Type
- characteristic: String
- extinct: boolean
- wings: int

Parrot
- id: int
- name: String
- food: Food[]
- foodQuantity: int
- type: Type
- characteristic: String
- extinct: boolean
- wings: int

Pigeon
- id: int
- name: String
- food: Food[]
- foodQuantity: int
- type: Type
- characteristic: String
- extinct: boolean
- wings: int

WaterBird
- id: int
- name: String
- food: Food[]
- foodQuantity: int
- type: Type
- characteristic: String
- extinct: boolean
- wings: int
-waterBody: WaterBody

+ getWaterBody(): WaterBody

WaterFowl

ShoreBird

Testing Plan

Parent class: Bird
testGetType() Eg: initialize a new bird object with type parrot, and check whether the getType() returns the parrot.
testSetType() Eg: set a bird object's type to be prey, and check whether the setType() returns the prey.
testGetChar() Eg: initialize a new bird object with characteristic "friendly", and check whether the getChar() returns "friendly"
testSetChar() Eg: set a bird object's characteristic to be "not friendly", and check whether the setChar() returns "not friendly".
testGetExtinct() Eg: initialize a new bird object with not extinct, and check whether the getExtinct() returns false.
testSetExtinct() Eg: set a bird object's type to be extinct, and check whether the setExtinct() returns true.
testGetWings() Eg: initialize a new bird object with 2 wings, and check whether the getWings() returns 2.
 testSetWings() Eg: set a bird object's type to have 4 wings, and check whether the setWings() returns 4.
testWingException() Eg: initialize a new bird object with -2 wings, and check whether the exception is thrown.

testGetFood() Eg: initialize a new bird object with food [nuts, seeds], and check whether the getFood() returns [nuts, seeds].

testSetFood() Eg: set a bird object's food to be [nuts, fruits], and check whether the setFood() returns [nuts, fruits]

Child class: preyBird extends Bird
Same.

Child class: flightlessBird extends Bird
Same.

Child class: owl extends Bird
Same.

Child class: parrot extends Bird
Same.
Test: testGetWords() Eg: initialize a new parrot object with 2 words, and check whether the getWords() returns 2.
testSetWords() Eg: set a parrot object's type to have 4 words, and check whether the setWords() returns 4.
testWordException() Eg: initialize a new parrot object with -2 words, and check whether the exception is thrown.
testGetFav() Eg: initialize a new parrot object with saying "hi", and check whether the getFav() returns "hi".
testSetFav() Eg: set a parrot object's saying to be "hello", and check whether the setFav() returns "hello".

Child class: pigeons extends Bird
same

Child class: waterbird extends Bird
Test: testGetWaterBody(); testSetWaterBody()

Child class: shorebird extends waterbird
same

Child class: waterfowl extends waterbird
same