LAB 1 - BIRDS

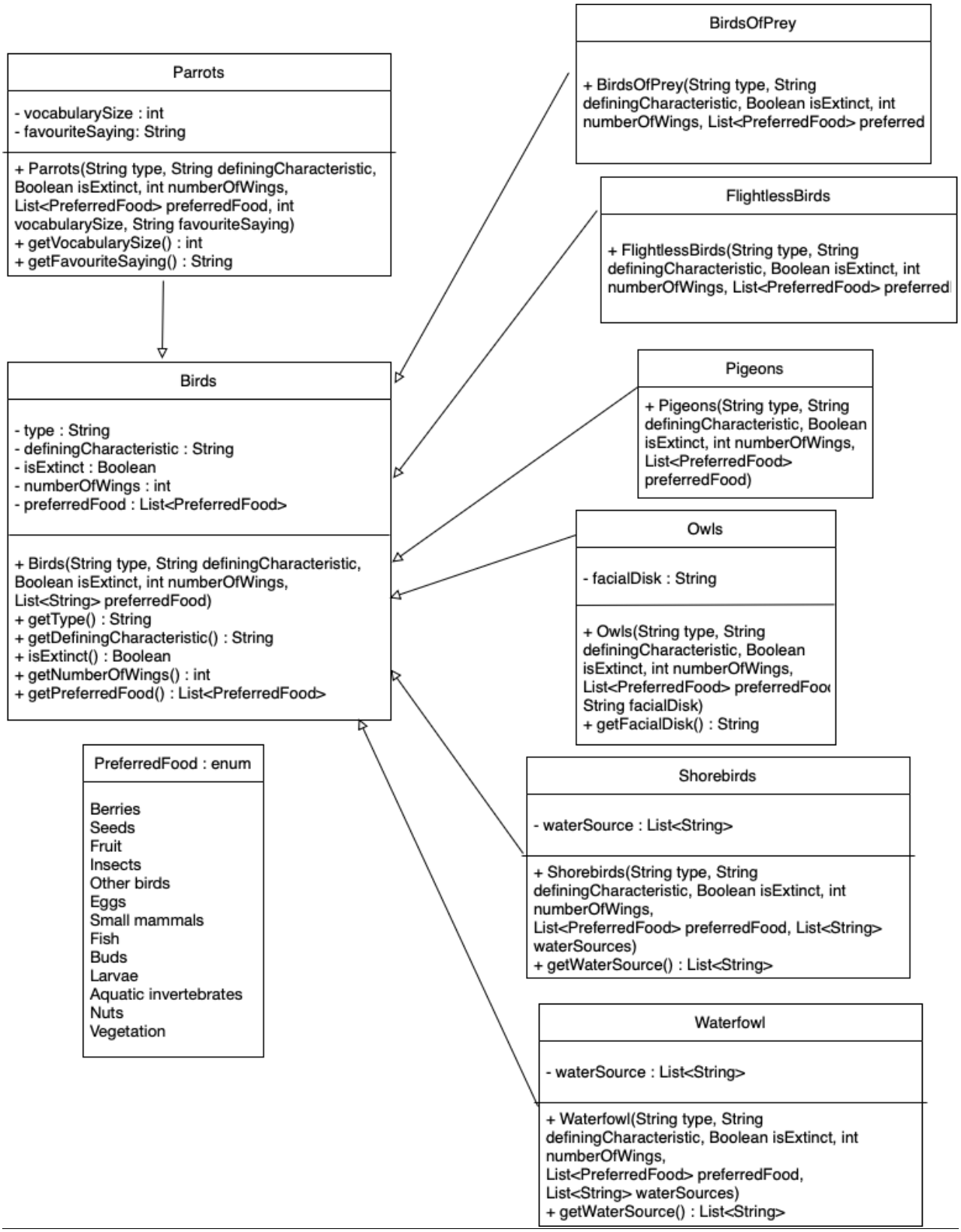
CS5010 – Programming Design Paradigm

Abstract

UML Class diagram and testing plan for Part - 1

Kiruthiga Muralidharan

UML Class Diagram:



Test Plan:

Test Method	Description	Input Assignment	Expected Output
testPreferredFood()	Checks whether the list of preferred food is returned	Pigeons("Columbiformes", "Feeds bird milk", false, 2, ["Berries", "Seeds"]);	["Berries", "Seeds"]
testPreferredFoodLimit()	Tests whether the size of the preferred food list is between 2 & 4	Pigeons("Columbiformes", "Feeds bird milk", false, 2, ["Berries"]);	Preffered food items list must be between 2 & 4.
testIsExtinct()	Checks the isExtinct value	FlightlessBirds("Apterygiformes", "Long, flexible beak with a downward curve on the end", false, 0, ["Berries", "Seeds"]);	false
testDefiningCharacteristic()	Checks the definingCharacteristic value	FlightlessBirds("Apterygiformes", "Long, flexible beak with a downward curve on the end", false, 0, ["Berries", "Seeds"]);	Long, flexible beak with a downward curve on the end
testWaterSource()	Checks the waterSources for the birds that live on water	Waterfowl("Anseriformes", "Live near Water Source", false, 2, ["Berries", "Seeds"], ["Freshwater"]);	["Freshwater"]
testFacialDisk()	Check the facial disk shape of Owls	Owls("Strigiformes", "Soft plumage", false, 2, ["Berries", "Seeds"], "Circular");	Circular
testVocabularySize()	Checks the vocabularySize of the Parrots	Parrots("Psittaciformes", "Short, curved beak", false, 2, ["Berries", "Seeds"], 20, "Welcome to PDP class!");	20

testFavouriteSaying()	Checks the favouriteSaying of the Parrots	Parrots("Psittaciformes", "Short, curved beak", false, 2, ["Berries", "Seeds"], 20, "Welcome to PDP class!");	Welcome to PDP class!
testNumberOfWings()	Checks the numberOfWings value	Pigeons("Columbiformes", "Feeds bird milk", false, 2, ["Berries", "Seeds"]);	2
testType()	Checks the type/category of the Birds	Pigeons("Columbiformes", "Feeds bird milk", false, 2, ["Berries", "Seeds"]);	Columbiformes

Test Plan Explanation - Samples:

Initiating objects for all base & Inherited classes:

1. Test the creation of different bird types, instatiating the objects for all base & inherited classes
2. Test Data Example: Create a parrot with type "Budgerigar," vocabularySize 29, favorite saying "Welcom to PDP!"
3. Expected Output: Object must contain the required values, all the values must be printed when toString() is called.

testPreferredFood:

1. Test assigning food items to a bird.
2. Test Data: Assign "berries" and "seeds" to a pigeon's constructor.
3. Expected Output: The foodItems list should contain these items.

testIsExtinct:

1. Set a bird's isExtinct to true.
2. Test Data: Create a flightless bird (Kakapo) and set isExtinct = true.
3. Expected Output: Correct flag for extinction must be present in the object.

testDefiningCharacteristic:

1. Test the defining characteristics of birds – Distinct feature for the bird
2. Test Data: Create a BirdsOfPrey with definingCharacteristic as "sharp, hooked beaks with visible nostrils"
3. Expected Output: The definingCharacteristing must contain the above.

testWaterSource:

1. Test if shorebirds and waterfowl are correctly assigned to live near bodies of water.
2. Test Data: Assign the “Pacific Ocean” as the waterSource for a horned puffin.
3. Expected Output: Correct body of water present in the object.

testFacialDisk:

1. Test facial disk shape for Owls
2. Test Data: Assign “Circular” as the facialDisk for an Owl.
3. ExpectedOutput: The object must contain Circular value as facialDisk