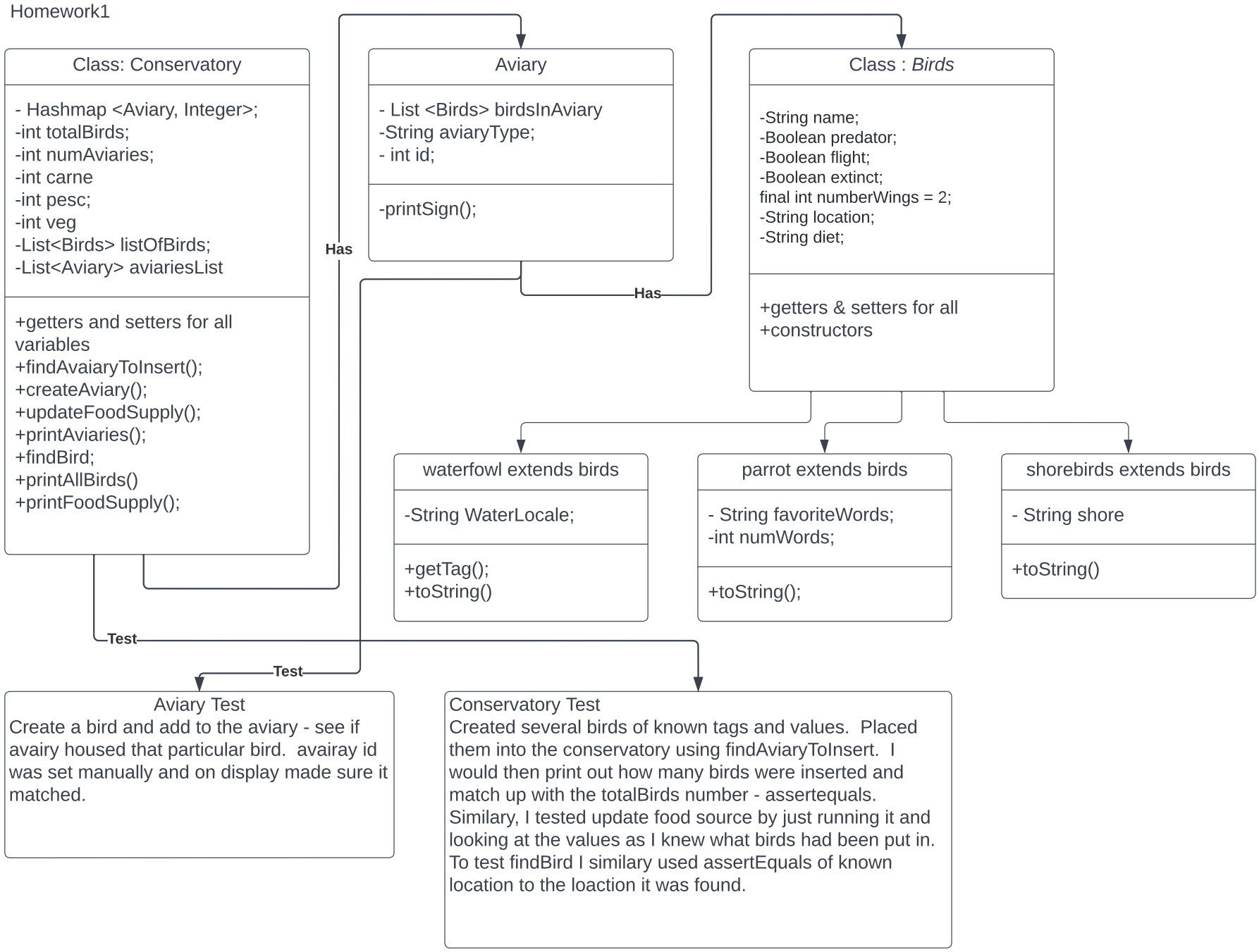
Homework1



How will your design encapsulate the different types of birds?
Initially, I had a boolean variable for each classification of bird that would make later distribution easier but I felt like that negated the purpose of this assignment. As such, I decided to make a class for each bird that would have an additonal method e.g. Parrots needed to demonstrate their vocabulary. All the data is encapsulated in each class respectively and accessed through getters and setters. I left predatory birds as a boolean simply because there was not an additional function or attribute that differentiated it from other birds except being a predator - thus I decided to leave it a boolean attribute.

What fields you expect to have in each class? What is their access and why?
The bird class is the main one that other bird types will extend from. I created extension classes for those that would have an additional method e.g parrot vocab. I made access all private. I did make numberOfWings final as a quick google stated that all birds have two wings. I did not further utilize this data variable. I choose private as we would not like visitors to manipulate our information e.g number of birds or change feed supply quatities.

What is your strategy for testing your design? How will this convince someone else that your code works correctly?
I tested each class as it was being constructed. After creating bird class I created several birds as outlined by the assignment and made sure they were being toString() and i could access the attribute. Once I built the randomBirdGenerator - i made sure I was generating birds. I made sure that the birds generated were able to be placed into the aviary. I did this with user created birds and I would print out the aviary list - this was more effective in this instance than assertEquals simply because it visually allowed me to see that each was in fact inserted into the aviary. To test conservatory was a little more intricate as I needed to ensure birds would be placed in to correct corresponding aviary. I did this by creating several test birds and inserting them into the conservatory and then proceeding to test the print all birds function as well as teh list aviaries function. Similarly, this allowed me to see that each bird was placed within the correct aviary and that a new aviary was created if necessary. To further test this I looped the bird adding an additional three times to see if the birds and avairies were filled in correctly. I used assertEquals in findBird to check if each bird was in the aviary it should be in the verify.