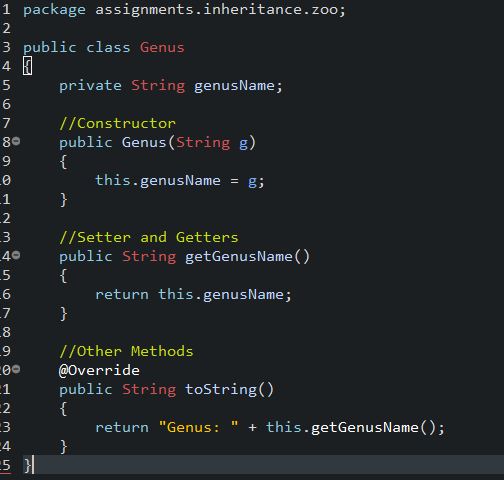
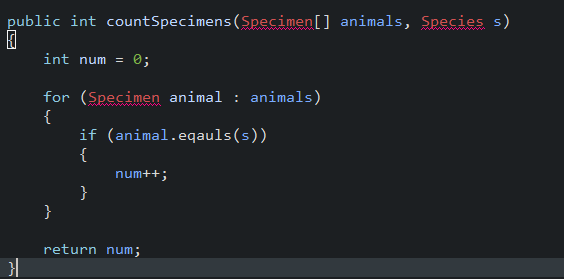
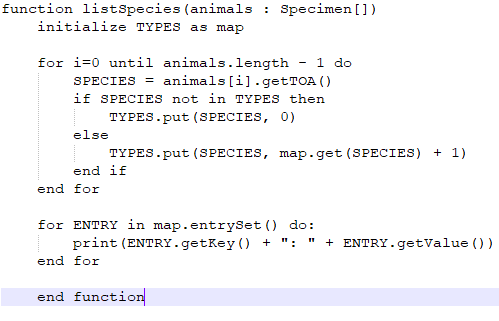
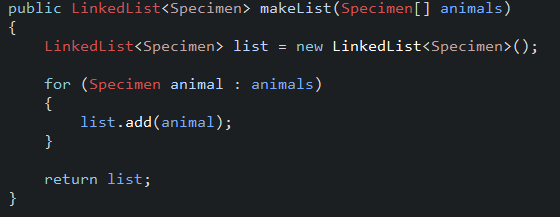
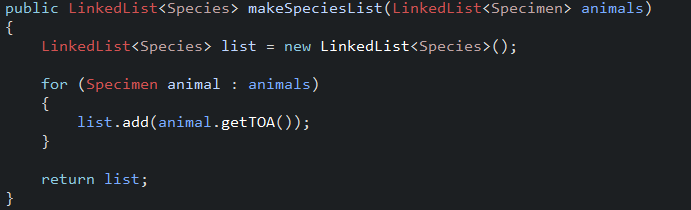
* 1. Genus is the parent class of Species class.
  2. Specimen instantiates an instance of the class Species as an instance variable.
  3. 
     1. They can utilize any fields or methods inherited from the Genus class without having to rewrite code specific for the Species class.
     2. They can represent a species as an object in the Specimen class and gain access to the public methods provided by the Species class.
     3. Because the toString() method in the Species class overrides the same method in Genus class.
     4. Overriding
  4. Encapsulation means data hiding, so certain fields and methods cannot be accessed or used by other classes.
     1. You can prevent other classes from directly accessing and changing certain variables in the class so as to not disrupt the flow of data in that class.
     2. You can provide input validation or do other operations before the variable is updated with the new value.
  5. getTOA()
  6. Toa
  7. 
  8. Advantage: The Specimen class will have direct access and inherit any of the public fields and methods from the Species class.  
       
     Disadvantage: It may prove to become quite overwhelming to inherit all the fields and methods from the Species class since the Specimen class doesn’t seem to have the purpose of being a type of Species, but rather a container for it.
  9. A private field called “markings” would need to be added to the Species class. Accessor and mutator methods would need to be added there as well in order to get data from the variable and to set the value of it. The toString() method can also be updated to display the value of “markings”.
  10. 
  11. 
  12. Abstract data types have their behavior defined by a series of values and operations, however the implementation is not explicitly defined.
  13. 
  14. 
  15. 