**Question Set 1**

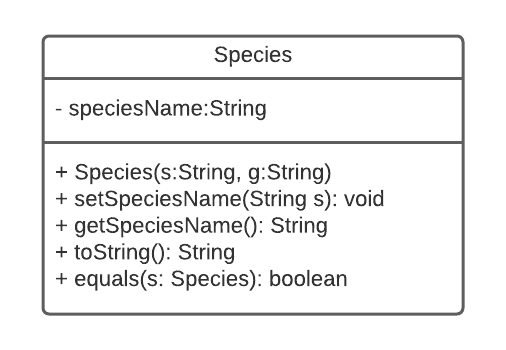
1. State the relationship between the Genus and Species objects

It is a parent-child relationship, where Genus is the parent class and Species is the child class (subclass).

1. State the relationship between the Species and Specimen objects

There is no relationship between the Species and Specimen objects.

1. Construct the unified modelling language (UML) diagram for the Species object



Attribute

Method

1. Two ways inheritance can benefit:

* Allows to re-use the code (Attributes and Methods) in the parent class. When the child class wants to implement or have the same attributes or methods, it can access the one that have initialized in the parent class without having to write the same code again.
* The code becomes more readable as inheritance create a relationship between classes where they can derive existing attributes and methods. Hence, it is more structured.

1. toString()
2. Inheritance allows us to override the methods of the parent class where those methods can be designed uniquely in the child class. Hence, each of the toString() method in each classes has its own unique functionality and it does not cause an error.
3. The term for this property is overriding.

**Question Set 2**

1. The term of encapsulation

Encapsulation is an OOP concept where it wraps attributes (variables) and methods together as a single unit, which we known as classes. These attributes of a class are private and it can only be accessed through methods in its own classes. However, we can access it outside its own class with public setter and getter methods. This concept is also known as data hiding where it gives protection to the information (attributes and methods) from being accessed outside of its own class.

1. Outline two benefits provided by encapsulation

* It prevents object accessed from unwanted clients. This concept will hide all data and methods in class and inner implementation of a class will not be visible to the user. The only way to access it is through the getter and setter methods created.
* It improves the code reusability and can be used all through the application or across multiple applications, such as a specific class can be reused whenever such type of object is required.

1. Identify an accessor method in the Specimen class

Accessor method is also known as getter method. Here, in the Specimen class, we have 3 getter methods, which are getName(), getCage(), and getTOA().

1. Identify an instance variable in the Specimen class

There are 3 instances variables declared in the Specimen class, which are name, cageNumber, and toa.

1. Construct code
2. One advantage and one disadvantage of having the Specimen object as a sub-class of the Species object

* Advantage: it makes the program more structured. Since Specimen places lower than Species in the hierarchy of animal, the relationship defines well with Specimen act as the sub-class. Here, the specimen can also access all the data and methods inside its parent class which in this case is Species.
* Disadvantage: it is more inflexible as further subclasses customization will be harder because all the methods and data are defined in a more general superclass.

**Question Set 3**

1. Changes that would be needed in order to add a description of each animal’s individual markings to the program

Add private instance variable with name “marking” with its data type is equal to String. In the Specimen constructor, we add another parameter which is ‘String marking’ and add ‘this.marking = marking’. Also, we can also add a setter and getter method for the marking instance.

public Specimen(String a, int c, Species s, String marking) {  
 setName(a);  
 setCage(c);  
 setTOA(s);  
 this.marking = marking;  
}

public String getMarking() {  
 return marking;  
}  
  
public String setMarking(String marking) {  
 this.marking = marking;  
}

b) Construct method countSpecimens(Specimen[] animals, Species s)

c) Pseudocode for listSpecies(Specimen [] animals)

listSpecies (Specimen[] animals) {  
 LinkedList<String> speciesList = new LinkedList <String>  
 for (each animal in animals) {  
 if (animal's species has not existed in speciesList) {  
 insert animal's species into speciesList  
 }  
 }  
 return allSpecies  
 }

**Question Set 4**

1. Identify the features of an abstract data type (ADT)

Abstract data type behaviour is defined by a unique set of attributes and set of operations. The abstract data type is made of with primitive data types, and we can perform different operations. However, the logic behind those operations are hidden (we don’t know how those operations work).