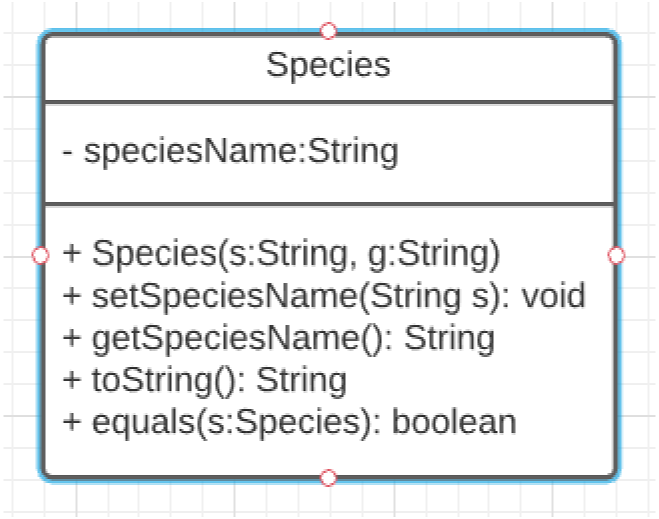
Inheritance Forum

Number 1

A. Genus is the parent class of Species and Species is the child class of Genus.

B. The relationship between them is ”part of” / aggregation relationship.

C. 
Species
- speciesName:String
+ Species(s:String, g:String)
+ setSpeciesName(String s): void
+ getSpeciesName(): String
+ toString(): String
+ equals(s:Species): boolean

D. – We can reuse the code because the parent-child relation so it is more efficient. Child class doesn’t need to rewrite same code as parent and child class can use methods in parent class.
- It is easier to read the code and structured.

E. i, because toString method are available in all class and it is overridden in each class to toString in each class is different so no error will be created.
ii, Overriding a method

Number 2

A. It is one of the fundamentals of OOP and encapsulation is a packaging/bundling of data with their methods. It is used to hide/expose values of a data object inside a class to determine whether it can be accessed by unauthorize party or not. One of the example of encapsulation is access modifier that is used to determine whether that value is private, public, or protected. If it is private, It can’t be changed from outside of the class but can be modified and fetched by using getter and setter.

B. – Protect what’s inside of the class from unwanted parties
- It is also easier for the programmer itself to understand the code because it is clearer and well structured.

```
C. setName(a);
   setCage(c);
   setTOA(s);
```

```
D. private String name;
   private int cageNumber;
   private Species toa;
```

E. Open Genus.java

F. – Advantage: In taxonomy, specimen is under species meaning that specimen is a subspecies meaning that specimen is a part of species. By implementing the same object relationship into code, we will also have a more structured code using objects.
   - Disadvantage: The methods in specimen might be dependent on the species class meaning that if something in species is changed, the specimen might also be affected and specimen class are not flexible in customizing its own methods.

Number 3

```
A.
Specimen(String a, int c, Species s){
    setName(a);
    setCage(c);
    setTOA(s);
    setMarking();
}
public void setMarking(String marking) {
          this.marking = marking;
}
public String getMarking() {
    return marking;
}
```

B. Open Specimen.java

```
C. 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
       }
```

Number 4

A. The abstract datatype is a datatype, where it contains a set of values and set of operations. Abstract meaning that, we can perform different operations using these datatypes. The ADT is made of with primitive datatypes, but operation logics are hidden.

B. Open Species.java
C. Open Species.java
D. Open Species.java