Rainamira Azzahra

2301900391

**Question Set 1**

1. Genus is the superclass/parent class and Species is the subclass
2. Specimen represent an instance of the Species class as an instance variable.

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

1. 1. It inherited any fields and methods from the Genus class without rewriting the code in the Species class.

2. it can instantiates Species as an object in the Specimen class and can access to the public methods in the Species class.

1. 1. Because there is also a toString() method in the Species class that override the same thing like in the Genus class.

2. Overriding

**Question Set 2**

1. Wrapping up of data under a single unit, so it will protect any variable from being accessed outside of the class.
2. 1. Prevent other classes from directly accessing and changing certain variables since it will distract the flow.

2. It increased flexibility. If we want to make the variables visible to other classes we can use setter and getter.

1. getTOA()
2. Toa
3. A screenshot of a cell phone

   Description automatically generated
4. Advantage: The Specimen class will automatically inherits any fields and methods from the Species class.

Disadvantage: I feel like it’s not the most efficient way because the Specimen class doesn’t have anything to do with the value inside of the Species class.

**Question Set 3**

1. Added a private field called “markings” to the Species Class. Also added accessor and mutator so it can be access to get and set the data. Updated the toString() method to show the value of “markings” .

A screenshot of a cell phone

Description automatically generated

1. function listSpecies(animals : Specimen[])

Initialize VARIETY as mp

for i=0 until animals.length -1 do

SPECIES = animals[i].getTOA()

If SPECIES not in VARIETY then

VARIETY.put(SPECIES, 0)

else then

VARIETY.put(SPECIES, mp.get(SPECIES))

end if

end for

end function

**Question Set 4**

1. Abstract Data Type is a type or class for objects which behaviour is defined by a set of value and a set of operations. It does not specify the implementation.
2. A screenshot of a cell phone

   Description automatically generated
3. A screenshot of a cell phone

   Description automatically generated
4. A screenshot of a cell phone

   Description automatically generated