Name : Sunny jovita [2301939046]

Inheritance 2

Question 1

1. State the relationship between the Genus and Species objects.

Ans = species object inherit (extends) the genus object.

1. State the relationship between the Species and Specimen objects.

Ans = species object is a field inside the specimen(not an inheritance) both have a relationship.

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

Ans =

-speciesName 🡪 private

+Species(String s, String g) 🡪 public constructor with arguments

+getSpeciesName() 🡪 public

+setSpeciesName(String s) 🡪 public

+toString() 🡪 public

+equals(Species sp) 🡪 public

1. Outline **two** ways in which the programming team can benefit from the way the

relationships between the three objects, Specimen, Species and Genus, have been represented in the code

Ans = from this relationship class, if the superclass is updated(Genus) then, the specimen also all the species will be updated to, because the classes are related each other, all the species objects can be treated ad genus objects too(because species classes are inside the specimen, and specimen objects are extended from Genus.

1. The Genus class implements a toString() method that produces an output string that

is different from the one produced by the toString() method in the Species class.

Consider the following code fragment:

Species human = **new** Species ("homo", "sapiens");

System.out.println(human.toString() );

1. Outline why calling the toString() method in this code does not cause an error.

Ans = calling toSring method doesn’t cause an error because, species inherits the genus method(so the species can call the genus’s method) even though we can call the Genus’s method, but still Species’s toString overrides Genus’s toString method

1. Identify the term for this property.

Ans = override the Species’s method (?)

Question 2

1. Define the term *encapsulation*.

Ans = is a concept in OOP that refers to the idea of bundling(combine) data and methods that work within one unit(class). It prevents other classes form doing anything inside the class field by settting them to be private and provide getter and setter methods for these fields.

1. Outline **two** benefits provided by encapsulation.

Ans = protects data from unwanted access by users, reduces human errors

1. Identify an accessor method in the Specimen class.

Ans = public String getName(), public int getCage(), public Species getToa()

1. Identify an instance variable in the Specimen class.

Ans = private String name, private int cageNumber, private Species toa

1. Construct code for the Genus object including a constructor, accessor methods and a toString() method.

Ans = public class Genus

{

   private string genusName;

   public Genus (string name){

     this.genusName = name;

   }

   public string getGenusName(){

     return genusName;

}

public void setGenusName(string name){

  this.genusName = name;

   }

   @Override

   public string toString(){

     return "Genus: " + genusName;

  }

}

1. Outline **one** advantage and **one** disadvantage of having the Specimen object as a sub-class of the Species object

Ans = advantage : Specimen objects would inherit all the attributes of the species object - would allow code in specimen object to access species-related methods directly

Dis : data in the species class may not be consistent across the associated Specimens

Question 3

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

Ans :  
there should be accessor (get/set) methods for the markings the toString() method(include the description of the markings)

1. Construct a method countSpecimens( Specimen[] animals, Species) that will output the number of specimens of the given species in the zoo.

Ans :

void countSpecimens(Specimen[] animals, Species s){  
// int sCount = 0;  
// int i;  
// for(i = 0; i < animals.length; i++){  
// if (s.equals(animals[i].getTOA()){  
// sCount++;}  
// }  
// System.out.println(sCount);  
// }

1. Construct an algorithm in pseudocode for listSpecies( Specimen[] animals ), which will generate a list of the different species in the zoo.

Ans =

animals[] 🡪 is an array containaing specimen objects

species 🡪 is a species object

start looping from 0 to the number of elements in animals, if the species equals to the animals(includes equal to the type of the animals), it’ll plus 1 to the num of the species. Then in the end of the loop, it’ll be print all the list of the species.

Question 4

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

Ans = List ADT (data is generally stored in key sequence in a list which has a head structure consisting of count, pointers, and also the address to compare data in the list.), Stack ADT(the pointer’s data is stored instead of data being stored in each node), Queue ADT(Each node contains a void pointer to the data and the link pointer to the next element in the queue. The program’s responsibility is to allocate memory for storing the data.)

1. Construct a method, makeList(), that builds a linked list containing the Specimen objects from an input array.

Ans = ….. I’m sorry sir, I don’t really understand how to to this one….

1. Construct a method, makeSpeciesList( LinkedList animals ), that will return a linked list of Species objects, one for each animal specimen present in the zoo. Note that the list returned will contain duplicate Species objects if the zoo has more than

one specimen of a given species.

Ans = ….. I’m sorry sir, I don’t really understand how to to this one….

1. Construct a method, makeSpeciesListUnique( LinkedList allSpecies ), that will take a linked list of Species objects as described in (3) and produce a new linked list

that contains only one Species object for each species in the zoo.

Ans = ….. I’m sorry sir, I don’t really understand how to to this one….