UTSA CS 3443: Application Programming

Lab 01

Objectives:

- Familiarity with Eclipse IDE
- Java syntax, the String class
- Object oriented programming

Task: Just another Zoo!

We are looking for a basic software to help manage a zoo. Version 1.0 of the software will create Java objects to represent the zoo itself, the zones/areas within the zoo, and the animals in each zone. With this software, zoo operators will be able to plan and track animals in each zone of the zoo as it makes progress and opens.

Getting Started

Create a new project in Eclipse, following the lab guidelines. Create the following new Java classes, in the default package: (*Use these names exactly!*)

- Zoo.java
- Zone.java
- Animal.java

Lab1.java

Place the 'Lab1.java' class provided in the same default package within your project. Note that this code ***will not compile*** until you have completed the requirements of this lab. There will be syntax errors until all dependencies (classes and methods) are created and implemented.

Lab1.java has a main method and will be the class to run our application. Follow the remaining instructions for each class in this lab in order to get your code to compile **- do not change the Lab1.java class**.

Zoo

This class will represent a Zoo object, which we will define as having:

- A name, represented as a String (i.e.: Animal Kingdom Zoo)
- Zones, stored as an array of Zone objects (i.e. Zone[])
- A toString() method, which calls upon the toString() method in Zone.java to return as a String all needed information.
- An addZone(..) method, which takes as a parameter a Zone object and returns nothing.

This class must have a constructor and getters and setters to accommodate its variables.

Zone

This class will represent a Zone object, which we will define as having:

- A name, represented as a String (i.e.: Tiger Zone)
- An array of Animal objects.
- A toString() method which returns a String representation of all animals in the zone
- An addAnimal(..) method, which takes as a parameter an Animal object and returns nothing.

This class must have a constructor and getters and setters to accommodate its variables.

Animal

This class will represent an Animal object, which we will define as having:

- A name, represented as a String (i.e. Tigger)
- A type, represented as a String (i.e. Tiger)
- Whether or not the animal is Carnivorous. This will be represented as a boolean (i.e. true, for Tigger)
- A toString() method which returns a String representation of the animal object

This class must have a constructor and getters and setters to accommodate its variables.

As this lab is meant to review regular arrays in Java, no other data structure may be used to store the objects required. (No ArrayLists are permitted, for example).

Output

Once your code compiles, you will be able to examine the output of your program. The output of your program **must** match the format of the sample below. This sample is the result of running the Lab1.java class with the given main method.

```
Welcome to the "Animal Kingdom Zoo"!
-----------------------------------
Monkey Zone :
----------------
 >>  Monkey - Bubbles the Chimp (Vegetarian)
 >>  Monkey - Grape Ape (Vegetarian)

 Tiger Zone :
----------------
 >>  Tiger - Shere Khan (Carnivore)
 >>  Tiger - Tigger (Carnivore)

 Lion Zone :
----------------
 >>  Lion - Aslan (Carnivore)
 >>  Lion - Simba (Carnivore)
 >>  Lion - Christian (Carnivore)

 Elephant Zone :
----------------
 >>  Elephant - Elmar (Vegetarian)
```

Rubric:

- (10pts) Comments & Formatting - All code is properly formatted and commented. This includes your full name and UTSA ID (abc123) at the top of each class. Each class and method should have a description of what it does, and methods should additional have a description of any parameters and returned data, as applicable.

- (10pts) Submission - The Eclipse project is correctly submitted to Blackboard, as abc123-lab1.zip.
- (50pts) Correctness - All classes are declared exactly as described above and function appropriately.
- (30pts) Tests - The grader will test your code by modifying the main method in Lab1.java to add additional zones and animals. If coded according to the requirements of this lab, your submission will output the correct result. *Note however if you "hard code" any portion of the lab (except given Lab1.java, which you should not modify) your submission will fail this test case.*)

Submissions which do not compile will receive a maximum of 10 points in total.
