

Software Development (CS2514) Assignment 3

Interfaces (Due: 9 April. Marks: 5)

General Comments

Carefully read the submission guidelines before you submit the assignment.

Like all other exercises, this is an exercise about implementing *maintainable* classes. You should always assume the specifications may change (slightly) and make sure your implementation can accommodate these changes with the minimum amount of effort.

Before you start implementing your classes, please make sure you understand the API. If you don't you'll make your life much more difficult.

Learning Objectives

For this assignment you will learn about interfaces. You will learn how to design an interface-based class hierarchy for the Zoo example from Lecture 8. You should already know the structure of the class hierarchy so all you have to do is translate this into (1) a hierarchy that uses interfaces that describe the API and (2) classes for the intermediate and bottom levels of the hierarchy.

You may find that using interfaces, though clean, requires a lot of programming (typing). A lot of the programming is straightforward. To save you some typing, there's only need to implement one concrete instance of the feline animals, one concrete instance of the canine animals, and a concrete class for the hippo. However, adding other feline and canine classes should be easy so you should still implement the proper hierarchy.

Your aim should be to maximise code re-use in the hierarchy, so *don't use copy-and-paste* to "implement" shared behaviour. **Important:** remember that you can use instance of concrete classes and *delegation* to implement shared behaviour.

Main Details

Implement the animal class hierarchy from Lecture 7 using interfaces.¹

Submission Details

- Please remember that all classes, attributes, and public methods should be commented using a proper Javadoc comment.

¹So you are not allowed to implement subclasses.

- The Javadoc class comment should explain the purpose of the class; *not the purpose of the assignment*.
- Please provide your name and student ID as part of your class Javadoc. You can use the @author tag for this:

```
@author Java Joe (ID 12345678)
```

Java

- Use the cs2514 moodle site to upload your program as a single .tgz archive called *Lab-3.tgz* before 23.55pm, 9 April, 2018. To create the .tgz archive, do the following:
 - ★ Create a directory Lab-3 in your working directory.
 - ★ Copy Animal.java, Main.java and your other user-defined Java files into the directory. Do not copy any other files into the directory.
 - ★ Run the command 'tar cvfz Lab-3.tgz Lab-3' from your working directory. The option 'v' makes tar very chatty: it should tell you exactly what is going into the .tgz archive. Make sure you check the tar output before submitting your archive.
 - ★ Note that file names in Unix are case sensitive and should not contain spaces.
- Note that the format of your submission should be .tgz: do *not* submit zip files, do *not* submit tar files, do *not* submit bzip files, and do *not* submit rar files. If you do, it may not be possible to unzip your assignment.
- No marks shall be awarded for programs that do not compile.