Requirements Document Reassessment

Nicholas Lorentzen

Roger Wistar

CO552

Wild Kingdom Requirements Document

**Purpose of Project:**

To create a program that shows effective use of different means of class design through a simple simulation of life.

**Type of System:**

A Java program.

**Deliverables:**

The project will require that you turn in a .zip file named “WildKindom\_username” that contains various files needed for the project. The main java file to run the program known as WildApp must contain the instructions necessary to run the entire simulation from one method call. This is already implemented when you receive the file, but must be expanded on to include new classes. You will then complete the partially completed Bug class so that it is used in the WildApp class. After this you can move on to include the following classes: QuickBug, TiredBug, Rock. These will act in manners similar to the Bug class and will use either inheritance or an interface to show good class design. For example, your Rock class must implement an Actor interface that includes four methods. Your final product should have each “animal” displayed with a different mark on the board: Bug (A, B, C), Rock(\*), QuickBug(Q), TiredBug(t) and should prompt the user to hit “n” to initiate a turn.

**Rubric:**

The project will be graded as found on Canvas on two different standards scored out of 5 (Points per requirement in parentheses) each defined below:

S01(Coding Style): You must write code that is fully documented (2); has effective identifier names (1); makes sensible use of whitespace (1); and has correct indentation (1).

For this, you must follow the style guidelines posted on Canvas to receive full credit.

S02(Creating Programs): Can write programs with no syntax errors (1); no runtime errors (1); no logical errors (3)

For this, the program must accomplish what is outlined in the deliverables without any errors.