CS 349 Task 2: Analysis of Design by Contract

Description

This task demonstrates design by contract on an extension of Task 1. Due to its complexity, the objective is to analyze an existing solution instead of creating your own. The analysis is both static and dynamic.

The entities are clams, dogs, fish, ghosts, three kinds of birds, and two kinds of trees. Each has its own way of attacking and defending itself, which are always the same for any interaction. Although this solution is standalone, it naturally coordinates well with the State and Observer patterns to be investigated later. To this end, it includes additional sequencing functionality that was not present in Task 1. Specifically, the attacker has separate states of beginning an attack (1), ending an attack (3), and updating the attack any number of times between these limits (2). Likewise, the entity being attacked accepts the attack (A), refreshes itself in some defensive action any number of times (B), and releases the attack when it is over (C). (The verb choice is awkward for clarity because it uniquely corresponds to the code.)

For example, a flying bird attacking a dog produces this output, where the first column corresponds to the encoding above.

```
1: AirBird has begun attacking Dog by chirping really annoyingly
A: Dog is being attacked by AirBird by chirping really annoyingly
2: AirBird is again attacking Dog by chirping really annoyingly
B: Dog is defending itself against AirBird by barking
3: AirBird has finished attacking Dog by chirping really annoyingly
C: Dog is no longer being attacked by AirBird by chirping really annoyingly
```

Requirements

The objective is to understand how this system functions as a correct and safe design that is scalable and extensible. The appropriate way to represent the answers is covered in lecture.

1. Draw the inheritance hierarchy of the classes and indicate which interfaces they implement.
2. Depict the entity constraints in graph form.
3. Depict the entity constraints in the table form. Number each cell uniquely for use in (4).
4. The entities are somewhat contrived, but their prescribed and proscribed behaviors have a pattern. For each entity interaction in (3), explain in your own view why the constraints are this way. In other words, based on the constraints, describe what you think my model of the cartoon world is.
5. Trace the execution of `f14cs349task2.test.Test.runTest2()`, which reflects the behavior of an `EntityBirdAir` attacking an `EntityDog`. Print out the source code in the PDF version and tape the pages together as a single document with pages 1 and 2 in the first column, 3 and 4 in the middle, and 5 and 6 in the right. Draw a red unbroken line from the first statement in MAIN through the last as execution proceeds.
6. How does `EntityBirdAir` differ from `EntityBirdGround`?
7. Explain how the code prevents invalid combinations at compile time.

Deliverables

Submit this assignment on paper in class. Fold it nicely and put your name prominently on the front or top.