0 Objective and Instructions

In Part B of Project 6, you will explore the inheritance feature of Java, which facilitates incremental development of software.

You will extend the simulation system developed in part A. Make the ocean more interesting by adding our three old friends to the sea: SeaWeed, Tuna, and Shark! You will also need to complete the P6B class to create the ocean and instances of the three creatures. However, you are not allowed to change the Controller class given in Part A. You will make one (only one) modification to the Creature class given in Part A. The Ocean class that you have written in Part A should work here without any modification.

For Part B, you will submit your completed files SeaWeed.java, Tuna.java, Shark.java, and P6B.java. Submit BadOcean.java only if you do the optional Bonus question in Part B.

1 Citizens of the Sea

Below, we describe how the three kinds of creatures are extended from the original Creature class. First, you need to modify the given Creature class by making the fields protected. This is the only change that you will make in class Creature. Recall that the visibility modifier protected allows a field (or method) to be inherited by subclasses.

If you had difficulties with P5 but haven’t looked at the solutions yet, read the solutions now! Learn from the solutions, but don’t copy the solutions. The new SeaWeed, Tuna, and Shark have the same behavior as in P5, but we expect that you will improve the class implementation now that you know how to use arrays and inheritance.

Classes SeaWeed, Tuna, and Shark extend class Creature. This is a requirement of Project 6. Do not write SeaWeed, Tuna, and Shark classes that are independent of class Creature.

Define a constructor for each class since constructors are not inherited:

- public SeaWeed(Ocean ocean)
- public Tuna(Ocean ocean)
- public Shark(Ocean ocean)

where ocean is where the creatures will reside.

Below are the methods (behaviors) that are specific to SeaWeed, Tuna, and Shark—you need to define these methods in SeaWeed, Tuna, and Shark to override those in class Creature.

- public String toString()
  This method returns the name of the creature and its location. For instance, the toString method of a Shark may return a String “Shark (3, 2)” if the Shark resides at position (3, 2) in the ocean.
- public char abbreviation()
  This method should return ‘W’ for SeaWeed, ‘T’ for Tuna, and ‘S’ for Shark.
- public void simulate()
  The rules of simulating the three kinds of creatures are exactly the same as that in Project 5, so please follow the description in P5 for the simulate method.

One important difference between this project and P5 is that you do not have the method getContenttype in the Ocean class and you only have one getCreature method (instead of the trio getSeaWeed, getTuna, and getShark). In order to differentiate between SeaWeed, Tuna, and Shark, you need to use the operator instanceof. To see how it works, let us take a look at an example.

Suppose you have object referenced by a variable c of type Creature and you want to know whether the object is in fact a SeaWeed, you can use the expression
c instanceof SeaWeed

which will return a boolean value, where true indicates that c is an instance of SeaWeed. Note that instanceof is an operator, not a method (notice that there’re no parentheses for parameters). The expression

null instanceof SeaWeed

where SeaWeed may be replaced by any valid class name, evaluates to the value false.

Note: Use the instanceof operator; do not add methods getSeaWeed, getTuna, and getShark in class Ocean. In Part B you should not change the class Ocean created in Part A.

Besides the specified methods described above, you may write additional private helper methods and declare private variables as necessary.

2 The P6B Class

Now put SeaWeed, Tuna, and Sharks in the sea and let them play!

In order to put the new types of creatures into the ocean, you should not use the old P6A class, which only knows to create some instances of the Creature class. Complete a new class P6B that has almost the same structure as P6A, but P6B creates and passes an array of type Creature where the cells (of the array) reference instances of SeaWeed, Tuna, and Shark.

The initial state of the ocean that you will create should looks like the following:

```
~ ~ ~
~ S T
W ~ ~
~ ~ ~
~ ~ ~
```

3 Bonus: A New Ocean

You can explore an extension to the Ocean. Suppose you want to simulate an ocean with some tough living condition, where at the beginning of every other round one of the creatures in the sea dies due to bad weather, pollution, or whatever. Each creature has the same probability dying.

Implement a class BadOcean that extends the original Ocean class. Class BadOcean represents an ocean with the difficult living condition described above. In order to get your bonus point, you should make your code efficient and compatible with both Part A and Part B, i.e., you can use the BadOcean class by simply replacing the following statement in P6A or P6B:

```
Ocean myOcean = new Ocean(h, w);
```

becomes

```
Ocean myOcean = new BadOcean(h, w);
```

You will need to use the visibility modifier protected for the fields in class Ocean. This should be the only change that you make to class Ocean created in Part A of this project.