Chickens & Eggs

Goals of the Assignment

The goal of this assignment is to practice creating and using Java classes, enumerations, and random. Please read this document ***in its entirety*** before seeking help from the course staff. As always, you will be expected to demonstrate good software engineering practice, including **unit tests**, good code documentation, and the use of version control.

Problem Statement

Chicken eggs come in two basic colors (brown and white). They also come in 6 different sizes (each of which has a minimum weight in ounces): Peewee (1.25oz), Small (1.5oz), Medium (1.75oz), Large (2.0oz), Extra-Large (2.25oz), and Jumbo (2.5oz). Every egg has a size and color, and may or may not be cracked. Two eggs are considered equal if they are the same color and size, and are both cracked (or not).

Every chicken has a name and may lay eggs. A chicken will always lay eggs of the same color and size. For example, a chicken named "Henny Penny" always lays large, brown eggs.

Activities

Here are some general guidelines to follow when implementing your solution to this assignment.

  a. Put all of your code into a package named "`chickens`".
  b. **Use proper encapsulation**. Only write accessors and mutators as needed.
  c. **You must write unit tests** for all classes and non-trivial methods, e.g. you do not need to write tests for accessors and mutators (though you may use them in your other tests).

1. Create a new Java type to represent the color of a chicken egg. Given that chicken eggs come in one of only two colors, consider the best way to create a type that may only have one of a specific list of predefined values.

2. Create a new Java type to represent the size of a chicken egg. Chicken eggs come in one of six sizes, each of which has a minimum weight in ounces. Given a value of this new type, it should be possible to get the minimum ounces. For example, an Extra-Large chicken egg should be 2.25 ounces.

3. Create a new Java type to represent a chicken egg. An egg has the following attributes/methods:
   a. Eggs are created with a color and size. New eggs are uncracked.
   b. An egg may be cracked. Once cracked, an egg cannot be uncracked.
   c. The string representation of a chicken egg is formatted like this example: "An uncracked LARGE (2.0oz) BROWN egg".
   d. Two eggs are considered equal if they are the same size and color, and are both cracked or uncracked. For example, two large, brown, uncracked eggs would be considered equal. Two peewee, white eggs would not be equal if only one of the eggs was cracked.

4. Create a new Java type to represent a chicken. A chicken has the following attributes/methods:
   a. A chicken has a name and always lays eggs of the same size and color. For example, a chicken named "Henny Penny" always lays large, brown eggs.
   b. A chicken can lay an egg with its respective size and color. There is a 50% chance that the egg will be cracked when laid.
   c. The string representation of a chicken is formatted like this example: "Henny Penny, a chicken that lays JUMBO WHITE eggs".
   d. Two chickens are considered equal if they lay eggs of the same size (regardless of the color).
   e. The Chicken class should include a static method to generate a chicken with a random name, egg size, and egg color. You must select from at least 10 unique names. It is up to you what names to use. Do not worry about testing this method.

5. Create a `ChickenFarm` class that defines a `main` method with the appropriate signature.
    a. Prompt the user to enter how many chickens are on the farm.
    b. Use the static method on the Chicken class to generate the corresponding number of chickens.
    c. Print the string representation of each chicken and the first egg that it lays to standard output (see the output examples below). Note that, because the name selection for each chicken is random, you may have two or more chickens with the same name.

6. Use the drawing tool of your choice to draw a UML class diagram of the entire program. It should contain at least 5 classes or other types. Make sure to include the appropriate relationships between the types. Save your diagram as an image file in your repository and push it to GitHub along with your solution.

Sample Run:

```
Enter the number of chickens on the farm: 7
Chicken Little, a chicken that lays EXTRA_LARGE WHITE eggs, laid An uncracked
EXTRA_LARGE (2.25oz.) WHITE egg.
Hillary Fluff, a chicken that lays MEDIUM WHITE eggs, laid A cracked MEDIUM
(1.75oz.) WHITE egg.
M-egg-atron, a chicken that lays EXTRA_LARGE BROWN eggs, laid An uncracked
EXTRA_LARGE (2.25oz.) BROWN egg.
Eggatha Christie, a chicken that lays SMALL WHITE eggs, laid An uncracked
SMALL (1.5oz.) WHITE egg.
Sal Monella, a chicken that lays LARGE BROWN eggs, laid An uncracked LARGE
(2.0oz.) BROWN egg.
M-egg-atron, a chicken that lays MEDIUM BROWN eggs, laid A cracked MEDIUM
(1.75oz.) BROWN egg.
Optimeggs Prime, a chicken that lays JUMBO WHITE eggs, laid An uncracked
JUMBO (2.5oz.) WHITE egg.
```

Submission Instructions

You must ensure that your solution to this assignment is pushed to GitHub *before* the start of the next lecture period. See the course syllabus for the rubric that will be used to evaluate your submission.