Extra exercises – Week 3

COMP250, Winter 2023

January 17, 2023

Note: These exercises are for extra practice only. Do not submit them as they will not be graded.

Intro to objects and classes

1. Suppose I have a class `Dog` inside the package `animals`. Suppose Dog is declared to be package private (no modifier). Is `Dog` visible from within the package `animals.domestic` ?

2. Assume there is a class called `Wizard` which is part of a package called *characters*. Let the following snippet of code be part of the `main` method of the class `BaneOfTorog` from the package *campaigns*:

```
double[] rolls = new double[4];
for (int i = 0; i < rolls.length; i++) {
    rolls[i] = Math.random();
}
Wizard w = Wizard.generateChar("Gandalf", rolls);
```

   Assuming that the code above compiles and runs correctly, write the header of the method `generateChar` from the `Wizard` class.

3. Define a new type `Book`:
   - Every Book should have a title (`String`), an author (`String`), and a price (`double`).
   - Write two constructors:
     - One takes the title and the price as inputs. It uses these inputs to initialize the corresponding attributes and it assigns "Anonymous" to the attribute representing the author.
     - The other takes the title, the author, and the price as input and initializes the attributes accordingly.
   - Other methods (it is up to you to decide whether the methods should be static or not):
     - add a method `onSale()` which modifies the price of the book by cutting it into half.
     - add the `toString()` method.
     - add a method `isMoreExpensive()` which takes one book as input and returns `true` if *this* book is more expensive than the one received as input, `false` otherwise.

4. Define a new type `BookStore` (it is up to you to decide whether the methods should be static or not).
   - Each BookStore has a name (`String`) and a set of Books (and array of `Book`s).
   - Add a couple of constructors: one receiving just a `String` as input, the other receiving both a `String` and an array of `Book`s.
   - Add a method `sale()` which starts a sale in the bookstore by selling all the books as half price.
   - Write the `toString()` method to see useful information about a `Bookstore` object.
   - Write a method `getRecommandation()` which returns the title of the most expensive book in the bookstore.
   - Write a method `betterEquipped()` which takes two bookstores as input and returns the one which has a greater number of books.