Object-Oriented Programming

Exercise session 8

Exercise 1

Open the three Java source files provided along with this document and observe how they implement cloning and equivalence checking. Then, answer the following questions.

— Why doesn't the `clone()` method from the `Book` class clone `author` and `title` ?

— Then, what is the benefit of the `clone()` method implemented in the `Book` class ?

— How does the `Book` class implement `equals()` and `hashCode()` ?

— Why is `log.get(i).clone()` in the `clone()` method from `BookLog` casted ?

— What would the `main()` method of `BooksTest` print if the `clone()` from `BookLog` performed superficial cloning ? Why ?

— In the `clone()` method from `BookLog`, why isn't the cloning of the instance variable `log` enough to achieve deep cloning ? Confirm your answer by browsing the Java documentation.

Tip : to make the cloning in the `BookLog` class superficial, simply remove (or comment) the 3 last instructions in the `try` block in the `clone()` method.

Exercise 2

Create the following classes :

— A `Person` class. Instances of this class should store a last name and a first name, both given at instantiation. They should also feature an optional short bio (as a `String` object). This bio can be set or left empty at instantiation, and should be modifiable after instantiation.

— A `Group` class. An instance of this class should store an ordered group of people as instances of the `Person` class. People should be added to the `Group` via an `add()` method which receives a single `Person` object. The order of the objects is the order of insertion.

Then, expand these classes to make them comply with equivalence checking mechanisms and implement deep cloning. They should also provide a `toString()` method returning a text description of the object in `String` format.

Regarding equivalence checking, you can use the same strategies as shown in the program involved in Exercise 1 as well as the following equivalence guidelines :

— Two `Person` objects will be equivalent if they have the same last name, first name and bio.

— Two `Group` objects will be equivalent if they store the same amount of `Person` objects and if these objects are pairwise equivalent and stored in the same order.

Finally, create a test program to ensure your implementation works.