

Object Oriented Programming

Exercise Series 7

Exercise 1 Take a look at the classes `Location` and `Country`. What could be improved with these two classes? Make the changes you deem required.

Exercise 2 In this second exercise, you are asked to implement a new class `City` that stores the following informations: its name, its area, its population, and the country (for now only represented as a `String`) it is in.

Where would this new class fit in the inheritance hierarchy you established in the previous exercise? Draw a diagram representing the subclassing relation that you would define between your classes, and explaining which application of inheritance is used in each branch of this relation.

Bonus: Let us now consider that we want to store the country variable (of an instance of `City`) as an object of the class `Country`. Modify the class `City` such that, during instantiation, if a `Country` with an identical name has already been instantiated, then the instance variable points towards it, and otherwise a new instance of `Country` is created.

Exercise 3 The file `ListOfPlaces.csv` contains a list of countries and cities formatted as follows:

- Each line corresponds to a unique location.
- Each field is separated by a semi-colon ";".
- The first field is either 1 for a country, or 2 for a city.
- Then follows, the name, area, and population of the location.

Additionally, a country will also contain a *calling code* described by the symbol "+" followed by a positive number:

```
1;[CityName];[Area];[Population];[CallingCode]
```

And a city will contain the name of the country it belongs to:

```
2;[CityName];[Area];[Population];[Country]
```

First, have a look at the files `HighDensity.java`. Use the javadoc to get familiar with the classes and method used. Compile it and then run it.

Modify the class `HighDensity`, such that its `getHighDensityLocation()` method returns an instance of a `Location` that corresponds to the place with the highest population density, among those present in the input file. Test it within the `main()` method.

Exercise 4 Finally, make sure that the input file satisfies the three following properties:

- The first field of each line can only be either 1 or 2.
- The fields corresponding to the area and the population only contain digits.
- The calling code field has "+" as its first symbol, followed by at least one digit (and only digits).

If not, throw a dedicated `Exception` describing the issue, and the line in which it appeared.