

## Assignment 11 — 25.11.2020 – v1.2

### Querying the Class Hierarchy

Please submit this exercise by mail to [pascal.gadiant@inf.unibe.ch](mailto:pascal.gadiant@inf.unibe.ch) before 02. December 2020, 10:15am.

**You must submit your code as editable text, i.e., use plain text file(s).**

In order to perform this assignment, you must import “Project Analyzer” and “Hierarchy Analyzer” from Github. You can do that in GT with just few steps. First, click on the “Git” button on the main screen. Next, click on the “+” sign at the top right, just next to the button “Fetch All”. In the overlay view, click on “clone” and then enter the URL below into the text field.

```
https://github.com/poojaruhal/sma2020.git
```

Next, click on “Clone”, then double-click on the value “sma2020” that should now appear in the list. In the upcoming view on the right hand side, please click on all three “Load” buttons. You successfully completed the clone of the repository when you see “Up to date” for every package. Once again, we recommend to save your image when you finished the clone process.

#### Exercise 1: Exploring projects (3 pts)

In GT, a project does not have a clear structure, but instead it consists of multiple packages. In this exercise, we will work on an easy to use project explorer that can present all the packages or classes of a project.

*NB:*

- *GT follows the convention that all package names with the same base name (i.e., prefix) belong to the same project. For example, packages related to the project `Collection` start with the prefix “`Collection`”, e.g., `CollectionsAbstract`, `CollectionsStack`, etc.*
  - *We define a “package group” as a list of one or more packages that use the same base name.*
- a) Describe the steps required to add an extension method. You can find an example in the Github repository you just cloned.
  - b) How many unique projects exist in the GT image? The class `RPackageOrganizer` can assist you with this task.
  - c) Collect all packages of the `Collections` project. The extension method `packageGroup` in `RPackage` can assist you with this task.

#### Exercise 2: Exploring hierarchies (7 pts)

A very common task in static code analysis is to fetch information about the inheritance relation between different classes. This requires the traversal of subclasses and superclasses, and the computation of hierarchy depth for each class. In this exercise, you have to map the GT class hierarchy to the project structure as discussed in the lecture. You are supposed to consider the `Object` class as root of all hierarchies, because all classes inherit from `Object`.

- a) Calculate the class hierarchy depth of the `Collection` class.
- b) Does the superclass of `HashedCollection` reside in the same package? You can use the extension method `isSuperclassInAnotherPackage` in `Class` from the Github repository.
- c) Do the subclasses of `HashedCollection` reside in the same package? You can use the extension method `isSubclassesInAnotherPackage` in `Class` from the Github repository.
- d) Calculate the depth of `Collection` in the package hierarchy. You can use the message implementation `HA_ClassExplorer calculatePackageLevel` from the Github repository.
- e) Add reasonable class comments to the classes `HA_ClassExplorer` and `PA_ProjectExplorer`. With the help of your comments, ensure the classes will be understood by other developers.

*NB: You can refer to the Pharo class comment template shown in the lecture, however you are not required to. You should write just about the information you believe is necessary to understand the class.*

- f) In this task, you are supposed to implement the method `calculateProjectLevel` in the class `HA_ClassExplorer`.