

Smalltalk: Reflection

Please submit this exercise to the git repository before 10h15, October 14, 2014. Don't forget to register to the [SDE student list](#). Also [register for the course](#) by Oct 10!

Exercise 1

Write a method which finds the top 10 classes in Pharo with the largest number of methods.

Exercise 2

Write a method which finds all the abstract methods in the Pharo programming environment.

Exercise 3

Write a method which finds all overriding methods in the Pharo system.

Exercise 4

Based on the code from the lecture and exercise ([CallGraph-exercises.st](#), [Calls.txt](#)), do the following exercise:

By redefining the method `JMethod>>doesNotUnderstand: aMessage` dynamically add the instance variable `numberOfCalls` to the class `JMethod`, and the method

```
JMethod>>numberOfCalls
  numberOfCalls := calls size.
  ↑ numberOfCalls
```

and by executing the code

```
(CallGraph fromFile: 'Calls.txt') methods collect: [ :each | each numberOfCalls]
```

print the number of calls for every method in the call graph.

Exercise 5

Find all the "duck-typed" methods in the `Collection` hierarchy. A method is "duck-typed" if implemented in two class hierarchies, without a common superclass implementing the same method.

Note: For these exercise please refer to Chapters 3 to 9 and Chapters 13 and 14 of the [Pharo by Example](#) book. Note: For the exercises, *file out* the appropriate source code and commit the `.st` file to the git repository on <https://bitbucket.org>.