SMA: Software Modeling and Analysis

Practical Session

Week 05
Assignment 04

Discussion
Write a method.

Find the *longest inheritance chain* among all Smalltalk classes in the GT programming environment.

```smalltalk
(((SystemNavigation default allClasses
    collect: [:eachClass | eachClass -> eachClass classDepth])
  sorted: [ :a :b | a value > b value ])
  asOrderedDictionary)
keys first
```
Write a *method*.

Find all *abstract method overrides* in GT.

```small
(SystemNavigation default allMethods select: #isAbstract)
flatCollect: [:m | ((m methodClass allSubclasses flatCollect: #methods)
select: [:n | m selector = n selector ]) reject: #isAbstract ]
```
Write a *method*. Find all *query method implementing classes* in GT.

```systemnavigation
SystemNavigation default allClasses

select: [ :class | class methodDict keys

anySatisfy:
[ :sel | ('is*' match: sel) | ('was*' match: sel) | ('will*' match: sel)]].
```
Write a *method*.  
i) Find all *root methods* in GT.

```small
introducedMethods := [:class | class superclass
  ifNil: [:class methods]
  ifNotNil: [:class methods select:
    [:met | (class canUnderstand: met selector) &
      (class superclass canUnderstand: met selector) not ]]]].

SystemNavigation default allClasses flatCollect:
[:cl | introducedMethods value: cl].
```
A04 - Exercise 04 | Root methods (BONUS)

Write a *method*.

i) Find all *duck-typed methods* in GT.

```ruby
rootMethods sort: [:m1 : m2 | m1 selector < m2 selector].

rootSelectors := (rootMethods collect: #selector).

duckSelectors := (rootSelectors asBag removeAll: rootSelectors asSet; yourself) asSet.

rootMethods select:
    [:met | duckSelectors includes: met selector]
```
Dynamic manipulation of code.

**Step 1:**
Dynamically add instance variable.

**Step 2:**
Dynamically add method to the class Call.

**Step 3:**
Resend initial message to self.
Call>>#doesNotUnderstand: aMessage
| messageName |
messageName := aMessage selector asString.

(messageName = 'numberOfArguments')
ifTrue: [ (self class allInstVarNames includes: 'numberOfArguments')
ifFalse: [ self class addInstVarNamed: 'numberOfArguments'].
self class compile:
messageName, String cr,
messageName, ' := args size.', String cr,
'^', messageName, '.'.
^ aMessage sendTo: self.].
Assignment 05

Preview
A05 - Exercise 01 | General questions

General reasoning.

• Is code reading a problem? Justify your answer.

• Give an example where a custom tool improved productivity by addressing a problem.
Writing code.

• **How many classes exist** that use the pragma `<gtView>` at least in one of their methods? Implement code that counts those classes.

• **Write a GT inspector extension** that displays instances of a `DateTime` object in the following format:
  `YYYY-MM-DD HH:MM`
More coding.

• What are the supported annotation names in live documents? Provide code that enumerates the possible annotation names.

  Example: \texttt{${\text{class}:\text{Object}}}$

• \textbf{Create a live document} that always shows the current number of classes available in Pharo.

  Step 1: Create a method using the correct pragma that returns all classes.
  Step 2: Reference the method in your live document code.