SMA:
Software Modeling and Analysis

Practical Session

Week 05

Assignment 04

Discussion

A04 - Exercise 01 | Hierarchy traversal

Write a *method*.

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

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

A04 - Exercise 02 | Method overrides

Write a *method*.

Find all ***abstract method overrides*** in GT.

```
(SystemNavigation default allMethods select: #isAbstract)

flatCollect: [ :m | ((m methodClass allSubclasses flatCollect: #methods)

select: [ :n | m selector = n selector ]) reject: #isAbstract ]
```

A04 - Exercise 03 | Query methods

Write a *method*.

Find all ***query method implementing classes*** in GT.

```
SystemNavigation default allClasses

select: [ :class | class methodDict keys

anySatisfy:
[ :sel | ('is*' match: sel) | ('was*' match: sel) | ('will*' match: sel)]].
```

A04 - Exercise 04 | Root methods

Write a *method*.

i) Find all ***root methods*** in GT.

```
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.

```
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]
```

A04 - Exercise 05 | Dynamic coding

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.

A04 - Exercise 05 | Dynamic coding

```
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.

A05 - Exercise 02 | Inspector extensions

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 DateAndTime object in the following format:
  `YYYY-MM-DD HH:MM`

A05 - Exercise 03 | Live documents

More *coding*.

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

  Example: ${class:Object}$

- ***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.