SMA: Software Modeling and Analysis

Practical Session

Week 04
You have to **attend the lecture to reveal such slides.**

*Disclaimer:*

The content that has been shown on this slide is irrelevant for the exam.
Assignment 04

Discussion
Write a method.

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

```
(((Smalltalk 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 the Pharo system.

```
(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.

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*.

Find all *root methods* in the off-the-shelf Pharo image.

```plaintext
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].
```
Dynamic extension of code.

Step 1:
Redefine code, add instance variable, add method receiver.

Step 2:
Execute code and observe results.
doesNotUnderstand: aMessage
|messageName|
messageName := aMessage selector asString.
messageName = 'numberOfCalls'
ifTrue: [ (self class allInstVarNames includes: 'numberOfCalls')
ifFalse: [ self class addInstVarNamed: 'numberOfCalls'].
self class compile:
messageName, Character cr asString, 'numberOfCalls := calls size.', Character cr asString, '^numberOfCalls'.
^ aMessage sendTo: self.]
Assignment 05

Preview
A05 - Exercise 01 | General questions

General reasoning.

• Is code reading a problem? Argue the answer.

• Give an example where a custom tool improved the productivity in addressing a problem or issue.
A05 - Exercise 02 | Smalltalk coding

Writing code.

• **Write an inspector extension** that prints the instance of a DateAndTime object in the following format: YYYY-MM-DD  HH:MM

• **How many inspector extensions exist** that use a table presentation? Implement code that counts those extensions.
More *textual questions*.

• *Enumerate at least two different embedded artifacts* that are essential for a live document.

• What is the *main difference between extensions* of the old and the new inspector?