

CodeScope

Main Author(s): S. Ducasse, Université de Savoie, stephane.ducasse@univ-savoie.fr

Currently developed on: Squeak, Can be used for lectures using Squeak may rely on some method specific for VisualWorks, latest version 15 available on <http://kilana.unibe.ch:8888/codescope>, contact person: bergel@iam.unibe.ch

The fact that Smalltalk classes and methods are objects too make easy to analyze applications written in Smalltalk. We ask you to build an application analyzing applications, classes and methods and producing reports using Seaside.

Here are some screenshots of a possible implementation of CodeScope.

CodeScope

Package:

Number of classes: 13
 Number of inst vars: 45
 Number of lines of code: 4254
 Method average lines of code: 8.42s2
 Class average lines of code: 327.23s2
 Class average of inst vars: 3.46s2
 Number of uncommented classes: 1

Class:

Hierarchy:

- Collection
- SequenceableCollection
- ArrayedCollection
- ByteArray
- CompiledMethod

Hierarchy nesting level: 7
 Number of methods: 108
 Number of inst vars: 0
 Number of extending methods: 0
 Number of overriding methods: 10
 Number of accesses to inst vars: 0
 Number of instances: 54788
 Number of uncommented methods: 32

Method:

Number of lines: 10
 Number of arguments: 2
 Has comment? yes

Class Method:

Number of lines: 2
 Number of arguments: 1
 Has comment? no

CodeScope

Package:

Number of classes: 21
 Number of inst vars: 21
 Number of lines of code: 3638
 Method average lines of code: 7.65s2
 Class average lines of code: 173.23s2
 Class average of inst vars: 1s0
 Number of uncommented classes: 7

Class:

Hierarchy:

- Collection
- SequenceableCollection
- ArrayedCollection
- String
- Symbol

Hierarchy nesting level: 6
 Number of methods: 213
 Number of inst vars: 0
 Number of extending methods: 3
 Number of overriding methods: 39
 Number of accesses to inst vars: 0
 Number of instances: 217932
 Number of uncommented methods: 58

Method:

Number of lines: 14
 Number of arguments: 0
 Has comment? yes

Class Method:

Here is the list of information that we would like to gather from one application and some hints how you can get this information. We consider a category as a package.

Methods

- Number of lines (without entry comments)
- Number of arguments

Classes

- Hierarchy nesting level
- Number of methods / class methods
- Number of instance variables
- Number of uncommented classes
- Number of extending methods (sending super)
- Number of overriding methods
- Number of accesses to instance variables
- Number of instances existing in the system (allInstances)
- Number of uncommented methods (aClass firstCommentAt: sel)

Packages

- Number of classes in a package (inspect SystemOrganization, Smalltalk at: #Array).
- Method average line of codes
- Class average line of codes
- Number of class reference inside/outside the application
- Total and average number of (lines of code, instance variables....)