

## 10. Exemplary Solutions: Extending CodeScope

The following code has been added or changed since Exercise 9 *CodeScope 1* to implement the new requirements.

---

```
CodeScope >> numberOfArgs
  ^ self method numArgs.
```

```
CodeScope >> numberOfTemps
  ^ self method numTemps.
```

```
CodeScope >> renderMethodColumnOn: html
  html div class: #column; with: [
    ...
    self method ifNotNil: [
      html div class: #information; with: [
        html text: 'Lines of code: ';
        strong: self method linesOfCode asString.
        html break; text: 'Number of args: ';
        strong: self numberOfArgs asString.
        html break; text: 'Number of temps: ';
        strong: self numberOfTemps asString ] ] ].
```

---

```
CodeScope >> numberOfInstances
  ^ self getClass allInstances size
```

```
CodeScope >> hierarchyNestingLevel
  ^ self getClass allSuperclasses size
```

```
CodeScope >> linesOfCodeInClass
  ^ self getClass linesOfCode
```

```
"This is an extension method placed in method protocol '*CodeScope' "
Behavior >> linesOfCode
  ^ self methods inject: 0 into: [:sum :method | sum + method linesOfCode].
```

```
CodeScope >> numberOfInstVars
  ^ self getClass instVarNames size
```

```
CodeScope >> percentageOfAccessors
  self numberOfMethods = 0 ifTrue: [^0].
  ^ (self getClass methods select: [:method | method hasInstVarRef]) size /
    self numberOfMethods * 100 asFloat printShowingDecimalPlaces: 2
```

```
CodeScope >> percentageOfUncommentedMethods
```

---

```
self numberOfMethods = 0 ifTrue: [^0].
^ self getClass numberOfUncommentedMethods /
  self numberOfMethods * 100 asFloat printShowingDecimalPlaces: 2

"This is an extension method placed in method protocol '*CodeScope' "
Behavior >> numberOfUncommentedMethods
^ (self methods select: [:method |
  (self firstCommentAt: method selector) isEmpty]) size

CodeScope >> percentageOfExtendingMethods
self numberOfMethods = 0 ifTrue: [^0].
^ (self getClass methods select: [:method | method sendsToSuper]) size /
  self numberOfMethods * 100 asFloat printShowingDecimalPlaces: 2

CodeScope >> percentageOfOverridingMethods
| subclasses |
self numberOfMethods = 0 ifTrue: [^0].
subclasses := self getClass allSubclasses.
^ (self getClass methods select: [:method |
  subclasses anySatisfy: [:class |
    class includesSelector: method selector]]) size /
  self numberOfMethods * 100 asFloat printShowingDecimalPlaces: 2

CodeScope >> averageLinesOfCodeInClassPerMethod
self numberOfMethods = 0 ifTrue: [^0].
^ self linesOfCodeInClass / self numberOfMethods
  asFloat printShowingDecimalPlaces: 2

CodeScope >> averageNumArgs
self numberOfMethods = 0 ifTrue: [^0].
^ (self getClass methods inject: 0 into: [:sum :method |
  sum + method numArgs]) /
  self numberOfMethods asFloat printShowingDecimalPlaces: 2

CodeScope >> averageNumTemps
self numberOfMethods = 0 ifTrue: [^0].
^ (self getClass methods inject: 0 into: [:sum :method |
  sum + method numTemps]) /
  self numberOfMethods asFloat printShowingDecimalPlaces: 2

CodeScope >> renderClassColumnOn: html
...
html div class: #information; with: [
  html text: 'Number of methods: ';
  strong: self numberOfMethods asString.
```

```
html break; text: 'Number of class methods: ';
  strong: self numberOfClassMethods asString.
html break; text: 'Number of instances: ';
  strong: self numberOfInstances asString.
html break; text: 'Hierarchy nesting level: ';
  strong: self hierarchyNestingLevel asString.
html break; text: 'Lines of code: ';
  strong: self linesOfCodeInClass asString.
html break; text: 'Number of inst vars: ';
  strong: self numberOfInstVars asString.
html break; text: 'Percentage of accessor methods: ';
  strong: self percentageOfAccessors asString, '%'.
html break; text: 'Percentage of uncommented methods: ';
  strong: self percentageOfUncommentedMethods asString, '%'.
html break; text: 'Percentage of extending methods: ';
  strong: self percentageOfExtendingMethods asString, '%'.
html break; text: 'Percentage of overriding methods: ';
  strong: self percentageOfOverridingMethods asString, '%'.
html break; text: 'Average LOC per method: ';
  strong: self averageLinesOfCodeInClassPerMethod asString.
html break; text: 'Average number of args per method: ';
  strong: self averageNumArgs asString.
html break; text: 'Average number of temps per method: ';
  strong: self averageNumTemps asString] ] ] ].
```

---

```
CodeScope >> numberOfMethodsInPackage
  ^ self classes inject: 0 into: [:sum :class | sum + class methods size]
```

```
CodeScope >> linesOfCodeInPackage
  ^ self classes inject: 0 into: [:sum :class | sum + class linesOfCode]
```

```
CodeScope >> averageNumMethodsPerClass
  self numberOfClasses = 0 ifTrue: [^0].
  ^ self numberOfMethodsInPackage /
    self numberOfClasses asFloat printShowingDecimalPlaces: 2
```

```
CodeScope >> averageLinesOfCodePerClass
  self numberOfClasses = 0 ifTrue: [^0].
  ^ self linesOfCodeInPackage /
    self numberOfClasses asFloat printShowingDecimalPlaces: 2
```

```
CodeScope >> averageLinesOfCodePerMethod
  self numberOfMethodsInPackage = 0 ifTrue: [^0].
  ^ self linesOfCodeInPackage /
    self numberOfMethodsInPackage asFloat printShowingDecimalPlaces: 2
```

```
CodeScope >> percentageOfUncommentedClasses
self numberOfClasses = 0 ifTrue: [^0].
^ (self classes reject: [:class | class hasComment]) size /
  self numberOfClasses * 100 asFloat printShowingDecimalPlaces: 2

CodeScope >> percentageOfUncommentedMethodsInPackage
self numberOfMethodsInPackage = 0 ifTrue: [^0].
^ (self classes inject: 0 into: [:sum :class |
  sum + class numberOfUncommentedMethods]) /
  self numberOfMethodsInPackage * 100 asFloat printShowingDecimalPlaces: 2

CodeScope >> renderPackageColumnOn: html
...
html div class: #information; with: [
  html text: 'Number of classes: ';
  strong: self numberOfClasses asString.
  html break; text: 'Number of methods: ';
  strong: self numberOfMethodsInPackage asString.
  html break; text: 'Lines of code: ';
  strong: self linesOfCodeInPackage asString.
  html break; text: 'Avg. number of methods per class: ';
  strong: self averageNumMethodsPerClass asString.
  html break; text: 'Avg. lines of code per class: ';
  strong: self averageLinesOfCodePerClass asString.
  html break; text: 'Avg. lines of code per method: ';
  strong: self averageLinesOfCodePerMethod asString.
  html break; text: 'Percentage of uncommented classes: ';
  strong: self percentageOfUncommentedClasses asString, '%'.
  html break; text: 'Percentage of uncommented methods: ';
  strong: self percentageOfUncommentedMethodsInPackage asString, '%'] ] ].
```

---