

## 9. Exemplary Solutions: CodeScope

### Exercise 9.1

---

```
WComponent subclass: #CodeScope
  instanceVariableNames: 'package class method packageChanged classChanged'
  classVariableNames: ''
  poolDictionaries: ''
  category: 'CodeScope'
```

```
CodeScope >> initialize
  super initialize.
  packageChanged := false.
  classChanged := false.
```

```
CodeScope >> style
'^'.column {
  float: left;
  font-family: Arial, Helvetica;
}
select {
  width: 220px;
  margin-right: 15px;
}
.information {
  margin-top: 20px;
  margin-left: 5px;
}'
```

```
CodeScope >> packages
^ PackageInfo allPackages asSortedCollection:
  [:a :b | a packageName <= b packageName]
```

```
CodeScope >> classes
^ self package classes
```

```
CodeScope >> methods
^ self getClass methods
  asSortedCollection: [:a :b | a selector <= b selector].
```

```
"Do not call the accessor of 'class' #class as this method is defined
in superclasses"
```

```
CodeScope >> getClass
^ class
```

```
CodeScope >> renderContentOn: html
  html form: [
    self renderPackageColumnOn: html.
    self renderClassColumnOn: html.
    self renderMethodColumnOn: html ]

CodeScope >> renderPackageColumnOn: html
  html div class: #column; with: [
    html strong: 'Packages: '; break.
    html select
      list: self packages;
      selected: self package;
      labels: [:pkg | pkg packageName];
      callback: [:pkg | self package: pkg];
      beSubmitOnChange;
      size: 30].

CodeScope >> renderClassColumnOn: html
  html div class: #column; with: [
    self package ifNotNil: [
      html strong: 'Classes: '; break.
      html select
        list: self classes;
        selected: self getClass;
        labels: [:cl | cl name];
        callback: [:cl | self class: cl];
        beSubmitOnChange;
        size: 30 ] ].

CodeScope >> renderMethodColumnOn: html
  html div class: #column; with: [
    self getClass ifNotNil: [
      html strong: 'Methods: '; break.
      html select
        list: self methods;
        selected: self method;
        labels: [:cm | cm selector];
        callback: [:cm | self method: cm];
        beSubmitOnChange;
        size: 30 ] ].
```

---

## Exercise 9.2

---

```
CodeScope >> numberOfClasses
  ^ self classes size
```

---

```
CodeScope >> renderPackageColumnOn: html
  html div class: #column; with: [
    ...
    html break.
    self package ifNotNil: [
      html div class: #information; with: [
        html text: 'Number of classes: '.
        html strong: self numberOfClasses asString ] ] ].
```

---

### Exercise 9.3

---

```
CodeScope >> numberOfMethods
  ^ self getClass methods size
```

```
CodeScope >> numberOfClassMethods
  ^ self getClass class methods size
```

```
CodeScope >> renderClassColumnOn: html
  html div class: #column; with: [
    self package ifNotNil: [
      ...
      self getClass ifNotNil: [
        html div class: #information; with: [
          html text: 'Number of methods: '.
          html strong: self numberOfMethods asString.
          html break.
          html text: 'Number of class methods: '.
          html strong:
            self numberOfClassMethods asString]]]]
```

---

### Exercise 9.4

---

```
CodeScope >> renderMethodColumnOn: html
  html div class: #column; with: [
    self getClass ifNotNil: [
      ...
      self method ifNotNil: [
        html div class: #information; with: [
          html text: 'Lines of code: '.
          html strong: self method linesOfCode asString ] ] ] ].
```

"The following three modifications to the mutators of 'package', 'class' and 'method' are necessary to not show old information for the other types of entities when changing package or class."

---

```
CodeScope >> package: aPackageInfo
    (self package ~= aPackageInfo and: [self package notNil]) ifTrue: [
        packageChanged := true. class := nil. method := nil].
    package := aPackageInfo
```

```
CodeScope >> class: aClass
    packageChanged
        ifTrue: [classChanged := true]
        ifFalse: [
            (self getClass ~= aClass and: [self getClass notNil]) ifTrue: [
                classChanged := true].
            class := aClass].
    packageChanged := false.
```

```
CodeScope >> method: aCompiledMethod
    classChanged
        ifTrue: [method := nil]
        ifFalse: [method := aCompiledMethod].
    classChanged := false.
```

---