SMA:
Software Modeling and Analysis

Practical Session
Week 07
Assignment 06

Discussion
Smalltalk coding.
Build a sunburst visualization to analyze test coverage of the Collection class hierarchy.

```smalltalk
| sb |
sb := RSsunburstBuilder new.
sb sliceShape withBorder.
sb sliceColor: [:shape |
    (Smalltalk includesKey: (shape model name, 'Test') asSymbol)
    ifTrue: [ Color green ]
    ifFalse: [ Color gray ]].
sb explore: Collection using: #subclasses.
RSMnormalizer size
    shapes: (sb shapes select: #isSLeaf);
    normalize: #numberOfLinesOfCode.
sb build.
sb canvas @ RSCanvasController.
^sb canvas
```
A06 - Exercise 02 | Tree Layout Visualization

**Smalltalk coding.**

Build a *tree layout visualization* to gather an overview of classes with subclasses that contain the string `Array` in their names.

```plaintext
| canvas shapes |
canvas := RSCanvas new.
shapes := (Collection withAllSubclasses) collect: [:class |
  (class subclasses notEmpty and: ["Array" match: class name])
  ifTrue: [RSEllipse new model: class; color: Color green; popup ]
  ifFalse: [RSEllipse new model: class; color: Color gray; popup ]].
canvas addAll: shapes.
RSEdgeBuilder line
  canvas: canvas;
  connectFrom: [:class | class superclass ].
RSNormalizer size
  shapes: shapes;
  normalize: [:cls | cls numberOfMethods].
RSTreeLayout on: shapes.
canvas edges pushBack.
canvas zoomToFit.
^canvas
```
Create a visualization to analyze the class dependencies between the Collection class hierarchy and the RSLayout class hierarchy.
A06 - Exercise 04 | Discussion

Visualization reasoning.

Comment on the strengths and limitations of each visualization you just created.

Sunburst visualization
Strengths: nice for hierarchies, tiles reveal relationship to parent tile
Limitations: hard to evaluate manually (circular sections are hard to estimate)

Tree layout visualization
Strengths: advantages like sunburst + area easier to estimate
Limitations: requires much space

Node-link visualization
Strengths: advantages from sunburst and tree layout, supports multiple overlaid relation visualizations
Limitations: very high resolution needed for further manual inspection
Assignment 07

Preview
General knowledge.

a) What is the cyclomatic complexity? Explain!
b) Which other metrics do you know?
c) Do metrics always express problems?
d) How and when are nowadays metrics integrated into development processes?
Writing code.

Find all classes that have > 100 methods in modelArgo.
Writing code and interpretation of the results.

a) Find all methods in `modelArgo` that have:
   1) > 150 lines of code, and
   2) an acyclomatic complexity of < 4

b) Apply your implementation to `modelSolr`.
   Which differences can you see in the result?

c) Is it appropriate to use the same thresholds for any models? Justify!
Writing code.

Add a method to `FAMIXType` to obtain the ATFD metric for its instances.

**Important:**

1) **ATFD counts the attributes from other classes used by a class.**

2) **We only care for methods that begin with “set” or “get”.**