Smalltalk: Software Visualization

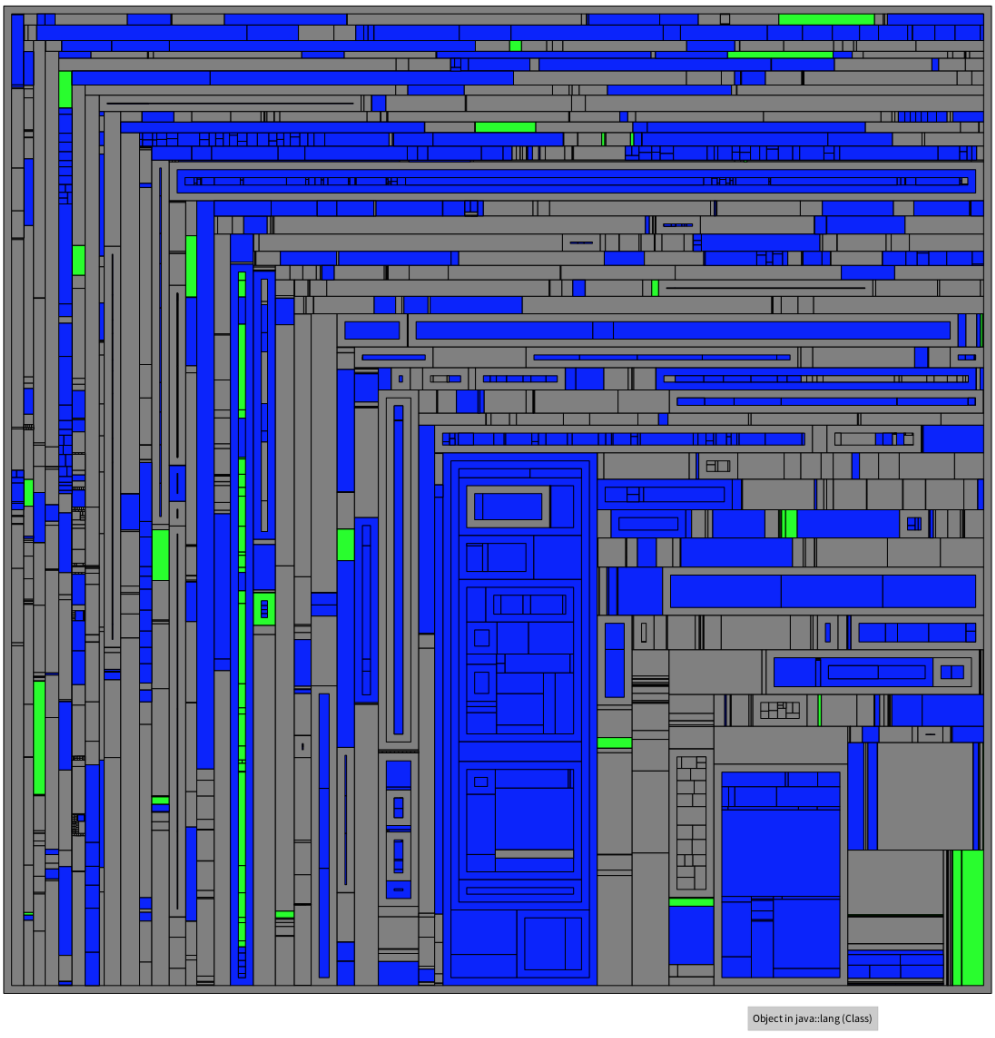
Please submit this exercise to the git repository before 10h15, November 1, 2016.

Exercise 1: Using Roassal API build a visualization to analyze Tomcat's class hierarchy.

Build a visualization that uses a Treemap to depict class hierarchy and that maps to the area of the tiles the number of lines of code of the class they represent. Use two different colors to identify the classes that contain deprecated methods and the ones that might be tests.

Hints:
(1) you can assume that the name of test classes contain the Test word.
(2) Include in the visualization stubs of the Java core and libs classes (*e.g.*, Object).
(3) Notice that the model of the methods of a class contains its annotations.

Note: Download Tomcat's model and sources.



Exercise 2:

Polymetric Views provides insights of the various aspects of a software by mapping multiple metrics to the attributes of rectangles that represent software entities (*e.g.*, classes). In this exercise you have to create a visualization that maps Tomcat classes to rectangles.
Each rectangle has to encode:
(1) the #hierarchyNestingLevel in its height
(2) the #numberOfAttributes to its width
(3) #fanOut in its color; and the x-position of rectangles must encode #fanOut and y-position =#fanIn.
The resulting visualization should look like the one below. Analysis of the visualization and highlight insights that you found.
Hint: How fanOut relates to other metrics?

Note: For the exercises, *file out* the appropriate source code and commit the .st file to the git repository on https://bitbucket.org.