Maastricht University Department of Data Science and Knowledge Engineering

KEN1520 Software Engineering
Assignment 6 - Rendering SVG content

Instructor: Tom Pepels

Summary: This lab involves rendering SVG content loaded from file to the screen, using the provided Decorator classes.

SVG_Week_6_Code.zip contains code based on the week 2 and week 4 assignments that loads SVG files into memory and lists their content.

The Element interface has been extended with a render() method.

```
public interface Element
{
     public String label();
     public int compare(final Element other);
     public Element newInstance();
     public boolean load(final String expr);
     public abstract void render();
}
```

In this lab, you will complete the implementation of the main.SVGRenderer class to render the SVG contents to the screen. A working framework is provided for you, all you have to so is add relevant code at the point marked:

```
// **
// ** TODO: Draw SVG contents here.
// **
```

The executable SVGApp.jar shows how your final app should work.

A concrete Decorator class is provided for each Shape and Style type for rendering them in Java Graphics2D.

You can create the relevant Decorator for each Shape as follows:

```
for (Element element : svg.elements()) {
      Shape shape = null;
      Decorator decorator = null;
  switch (element.label()){
      case "circle":
           shape = (Circle)element;
```

```
                    decorator = new
DecoratorGraphics2DCircle((Circle)shape, g2dImage);
                    break;
                      …
            }
```

You can create the relevant Decorator for the shape's Style properties as follows:

```
     for (Style style : shape.styles())
          switch (style.label()) {
               case "stroke-width":
                    new DecoratorGraphics2DStrokeWidth
                    ((StrokeWidth)style, g2dImage).render();
                    break;
               }
```

For the purposes of this assignment, you only have to handle the "stroke-width" style. The fill colour and stroke colour for each shape have already been handled during SVG loading and stored in Shape.

If you have time, try implementing a DecoratorFactory class to create the appropriate Decorator for each element, then use it in your SVGRenderer code instead. This step is not mandatory but will be worth bonus marks.

Steps

1. Split into groups of eight. Anyone left over, just join a group.

2. Get familiar with the code and understand how it works.

3. Implement the missing code in main.SVGRenderer.java. Your code should draw all shapes shown in the test file test-1.svg in the correct colour and stroke widths.

Resources

Scalable Vector Graphics (SVG) is a plain text XML-based vector image format for 2D graphics, with an open specification (https://www.w3.org/TR/SVG11/).

The SVG Wikipedia page gives an overview of the SVG format:
https://en.wikipedia.org/wiki/Scalable_Vector_Graphics

Introduction to Scalable Vector Graphics gives more detail:
http://fivedots.coe.psu.ac.th/Software.coe/J2ME/SVG/x-svg-a4.pdf

Submission

Deadline for submission is **18:00 Wednesday 22/05/2020.**

Upload your work to the KEN1520 section on EleUM. Your submission should include:

1. Your source code as a single .zip file.
2. A PDF document containing your list of team members (names and student numbers) and any relevant notes, e.g. why you made certain design choices, etc.
3. Mail your work to tom.pepels@maastrichtuniversity.nl with the subject: **Assignment 6 Group <n>**