© Manfred Kerber
Alexandros Evangelidis

School of Computer Science
University of Birmingham

Worksheet 6

MSc/ICY Software Workshop

Assessed Exercise: 9% of the continuous assessment mark.

Submission: Monday 9 January 2017 2pm

5% late submission penalty within the first 24 hours. No submission after 24 hours.

JavaDoc comments are mandatory. Follow the submission guidelines on

`https://canvas.bham.ac.uk/files/3242907`.

Assessment via viva. In your submission write all classes in a single directory, called WS1-6, from which you create your zip file.

Exercise 1: (Basic, 20%) In a computer game Christmas trees should be put in the background of a scene. Your task in this and the following exercise is to build corresponding Christmas trees that come with some variation.
The simplest type of tree has the shape of a (green) triangle plus a (brown) rectangle, the trunk. The brown trunk has a size of 2 units times 4 units, the width of the triangle at the bottom is 12 units, its height 12 units, as displayed in Figure 1. Trees can be re-sized by a scaling factor of type `int` which is the same for the `x` and for the `y` coordinate.

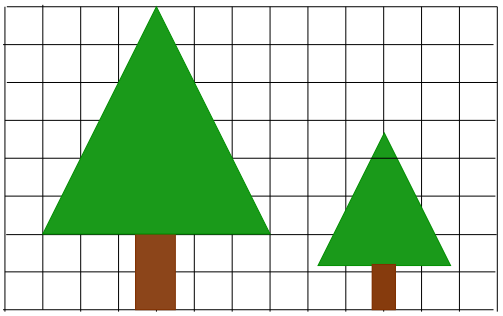

Figure 1: Two simple Trees

Write a class `Trees` to draw simple Christmas trees as in Figure 1, each made up of one green triangle and one brown rectangle. The constructor `public Trees(int[] xTrees, int[] yTrees, int[] scaleTrees)` takes three `int` arrays which store the `x`- and `y`-values of the bounding boxes of the trees as well as their scales, respectively. (E.g., a tree with `scale` 2 is twice as tall and wide as a tree of `scale` 1). Use the `fillPolygon` method to draw your trees. Put at least 5 trees on a panel to display them in a main method.

Exercise 2: (Basic, 20%) Write a class `StarTrees` that extends `Trees` so that each star tree has a star at its top (as described, e.g., at `https://en.wikipedia.org/wiki/Star_polygon`). A star is characterized by the number of vertices and the number of steps to come to the next vertex. Stars should be drawn correctly for the cases where the number of vertices and the steps do not have common divisors. Write a constructor `public StarTrees(int[] xTrees, int[] yTrees, int[] scaleTrees, int numberOfVertices, int steps)`. Put at least 5 trees (with stars that have 11 vertices and step size 4) on a panel to display them in a main method (see, e.g., Figure 2).

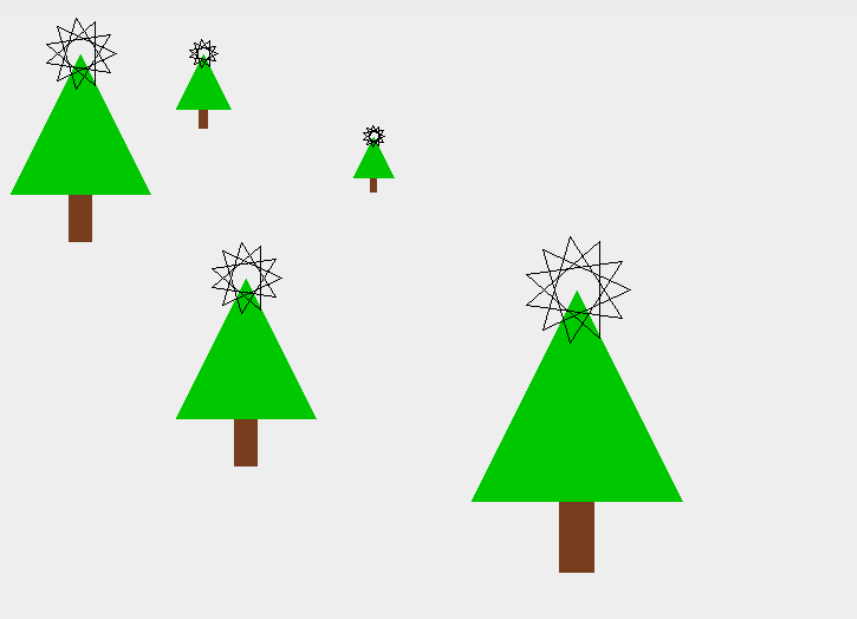

Figure 2: 5 `StarTrees`. Each star has 11 vertices and step size 4

Exercise 3: (Medium, 20%) Assume you are to build a GUI `HtmlReadGuiInner.java` that allows a user to download an HTML file of their choice to a local file of their choice. To this end they have to fill in the URL of the HTML file they want to download, the file name to which it should be written, and then press a button called `"DOWNLOAD"`.

Implement a corresponding program that makes use of inner classes. Give appropriate feedback of whether the download succeeded or not.

Exercise 4: (Medium, 20%) Repeat the previous exercise, however, instead of using inner classes make use of functions (lambda expressions).
(Call the class `HtmlReadGuiLambda.java`.)

Exercise 5: (Medium, 20%) Write again the same GUI, however, this time write all `ActionListeners`/`ChangeListeners` in separate files (in a similar way as we have seen in the example of the conversion of temperatures).