© Manfred Kerber
Alexandros Evangelidis

School of Computer Science
University of Birmingham
19 November 2019

Worksheet 5

MSc/ICY Software Workshop

Assessed Worksheet: 3% of the module mark (5% for the 20cr version).
Submission Deadline is Tuesday, 3 Dec 2019, at 12:00 noon via Canvas.
Follow the submissions guidelines on Canvas. JavaDoc comments are mandatory.

Note that this worksheet will be assessed in a viva (oral presentation of your code to a tutor). Hence prepare for all exercises main methods to demonstrate your running code. The tutor may want to make modifications to the main method in order to see how the graphics will change. Furthermore note that WE will NOT provide JUnit tests for this worksheet. Still YOU should submit such tests for those parts of your programs that can be tested with JUnit tests.

Exercise 1: (Basic, 30%) Using the `javafx.scene.shape.Polygon` class, write a class `Upload.java` that displays an upload icon as shown below.

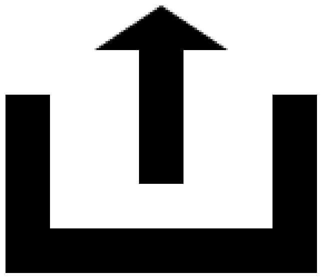
Fig. 1: Upload icon

Exercise 2: (Medium, 30%)
Write a minimal class `Expenditure.java` that represents an expenditure with the two field variables `private String description` and `private int value`. Write a constructor and the getters.
Note that you can sort an array of type `Expenditure` from biggest to smallest by the `sort` method from the Java API class `java.util.Arrays` as follows:

```
Arrays.sort(expenditures, (Expenditure exp1, Expenditure exp2) ->
                           exp2.getValue() - exp1.getValue());
```

Assume that a company uses an array of type `Expenditure` to represent all its expenditures. You have to write a class `Waffle.java` in order to represent expenditures as a Waffle chart. See, for instance, `https://en.wikipedia.org/wiki/Pie_chart#Square_chart_/_Waffle_chart`.
For an array of up to `maximum` elements, display all elements (biggest to smallest). For an array with more than `maximum` elements, display only the `maximum-1` biggest ones (biggest to smallest) and accumulate the others summed up as an entry with the description `"Other"`. For instance, with `maximum = 8;` and `expenditures` as left below the waffle chart should look like right below (Fig. 2).

```
Expenditures expenditures =
  new Expenditure[]
    {new Expenditure("Salaries", 11000),
     new Expenditure("Paper", 2000),
     new Expenditure("Rent", 5000),
     new Expenditure(
      "Most popular books on Java etc.",
      10000),
     new Expenditure("Heating", 3000),
     new Expenditure("Coffee/Tea", 7000),
     new Expenditure("Biscuits", 8000),
     new Expenditure("Travel", 18000),
     new Expenditure("Electricity", 1000),
     new Expenditure("Pencils", 3000)
    });
```

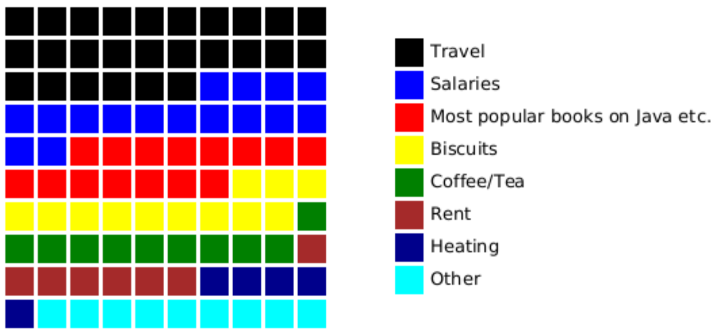

Fig. 2: Waffle chart

<u>Exercise 3:</u> (Advanced, 30%) Like in exercise 2, we want to represent expenditures, however, this time as a pie chart. Write a class `Pie.java` to represent all expenditures in a corresponding array of type `Expenditure`. Write a method `public void start(Stage stage) throws Exception` that displays the expenditures as a pie chart with the `description` placed next to the corresponding part of the pie chart. Take care that the description does not overlap with the pie chart itself.
Do not make use of the `javafx.scence.shape.PieChart` class, but only the two shapes `javafx.scene.shape.Circle` and `javafx.scene.shape.Line`. You need also `javafx.scence.text.Text`. For instance, with the values from Exercise 2 your pie chart should look like the figure on the left (Fig. 3). The right figure (Fig. 4) gives an example how the descriptive text should NOT look like.

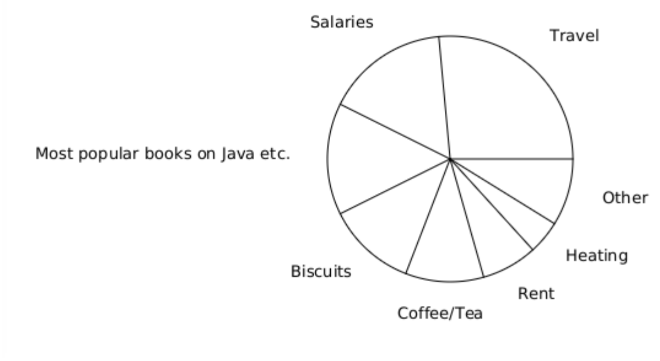

Fig. 3: How the pie chart should look.

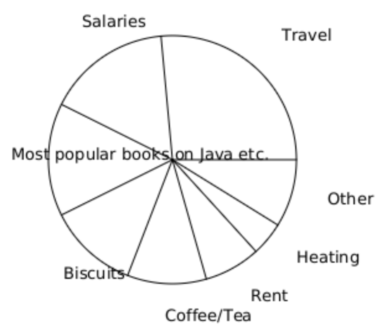

Fig. 4: How the pie chart should NOT look.

<u>Exercise 4:</u> (Debugging, 10%)
Code provided on the assignments Canvas page as `DisplayFunctionArea.java` should display the area between a function and the `x`-axis. For instance, for a method call `displayFunctionArea(x -> x*x*x - 8 * x * x, -2, 9, 500);` the area should be displayed as below to the left (Fig. 5). However, the actual display is incorrect and is shown as below to the right (Fig. 6). The whole area is compressed in the left upper corner and the `x`-axis does not show. Summarize the problems with the code and provide a corrected version that displays the area and the `x`-axis appropriately.

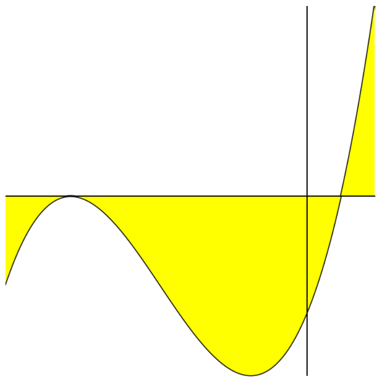
Fig. 5: How the display should look in the example.

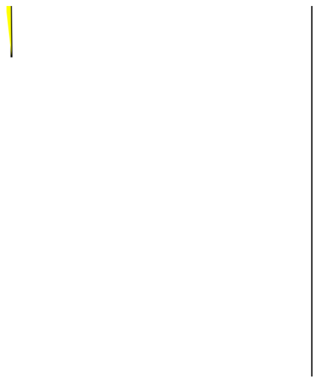
Fig. 6: How the display actually looks.