© Manfred Kerber
Alexandros Evangelidis

School of Computer Science
University of Birmingham

Worksheet 3

MSc/ICY Software Workshop

Assessed Exercise: 2% of the module mark.

Submission: Thursday 2 November 2017 2pm

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

JavaDoc comments are mandatory. Follow the guidelines in `submission.pdf`

Public tests will be provided a week before the submission deadline. The exercises must pass all these public tests.

Exercise 1: (Basic, 25%) Selection sort is a simple sorting algorithm which sorts arrays from the start by going once through the array and always selecting the smallest element from the rest of the array and swapping it with the element currently under consideration. E.g. (with the bar indicating the position of the element under consideration)

```
    [|4,  3,  6,  1,  9,  2]  Smallest element being 1
->  [ 1, |3,  6,  4,  9,  2]  Smallest element being 2
->  [ 1,  2, |6,  4,  9,  3]  Smallest element being 3
->  [ 1,  2,  3, |4,  9,  6]  Smallest element being 4
->  [ 1,  2,  3,  4, |9,  6]  Smallest element being 6
->  [ 1,  2,  3,  4,  6, |9]  Smallest element being 9
->  [ 1,  2,  3,  4,  6,  9|] end of array reached
```

Give a Java implementation of selection sort using loops,
`public static int[] selectionSort(int[] numbers)` in a class `SelectionSort`.
NOTE: Since any sorting algorithm would pass the tests which we will provide, there will be an inspection by your tutor of whether you actually implemented `selectionSort`. You will get marks only for an implementation of `selectionSort`.

Exercise 2: (Basic, 25%) Translate the cartoon below into a program, implemented in a class `StarRating` with a method `public static String interpret(double rating)`.

Concretely, write a method `public static String interpret(double rating)` that interprets any rating $1 \leq x \leq 5$, (which could be interpreted as the rating averaged over several individual ratings). A `"CRAP"` rating starts at `1.0` and may extend up to, but excluding, `4.0`, `"OK"` should start from `4.0`, `"EXCELLENT"` from `4.5`, and `"[HAS ONLY ONE REVIEW]"` should be returned for `5.0`. For values less than `1.0` or bigger than `5.0` your method should throw an `IllegalArgumentException`.

© Randall Munroe `xkcd.com/1098/`

Exercise 3: (Medium, 20%) Write a method that returns an ArrayList of all those integers in a range between the integer `from` inclusively to the integer `to` exclusively which do contain a particular digit, that is, a method
`public static ArrayList<Integer>`
`allIntegersWith(int from, int to, int containedDigit)` in a class `Contains`. E.g., `allIntegersWith(23, 53, 3)` contains the following elements: 23, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, and 43.

Exercise 4: (Advanced, 15%) Add to the class `PGMImage.java` from the lecture so that it contains three further methods:

(a) `public int[][] quarter(String filename)` which averages four adjacent cells and replaces them with a single cell in a PGMImage by averaging and rounding. Your method should return the resulting two-dimensional array and write it to a PGMImage file. For instance:

$$\begin{bmatrix} 10 & 20 & 30 & 40 \\ 10 & 10 & 20 & 60 \\ 25 & 25 & 30 & 50 \\ 25 & 45 & 10 & 10 \end{bmatrix} \text{ would result in } \begin{bmatrix} 13 & 38 \\ 30 & 25 \end{bmatrix}$$

If the number of rows or the number of columns in the array are odd ignore the last row and/or column.

(b) `public int[][] rotate(String filename)` which rotates the image by 90 degrees clockwise. Your method should return the resulting two-dimensional array and write it to a PGMImage file. For instance:

$$\begin{bmatrix} 13 & 38 \\ 30 & 25 \end{bmatrix} \text{ would result in } \begin{bmatrix} 30 & 13 \\ 25 & 38 \end{bmatrix}$$

(c) `public int[][] bw(String filename)` which transforms the image into a black and white image and writes it to a file. To this end, find the average grey value, all pixels above the average should go to white, all equal or below to black. Your method should return the resulting two-dimensional array and write it to a PGMImage file.

Exercise 5: (Advanced, 15%) Write a class `ArrayMethods`. Assume that a two-dimensional array `a` of type double is given (e.g., with dimension `m` × `n`, temperatures measured at `m` different weather stations at `n` different points in time). For the array write the following methods:

- `public static double min(double[][] a)`,
- `public static double max(double[][] a)`,
- `public static double mean(double[][] a)`, and
- `public static double median(double[][] a)`,

which compute the minimum, the maximum, the mean value (the average value), and the median (that is the middle most value of all values).
For instance, in an array `double[][] a = {{2.0, 3.1, 1.0}, {5.0, -3.8}};` the minimum is `-3.8`, the maximum `5.0`, the mean value 1.46 (computed as the sum with 7.3 divided by 5), and the median as 2.0 (since two values are smaller and two values are bigger than it; -3.8 and 1.0 being smaller and 3.1 and 5.0 being bigger). For the median of an array with an even number of numbers in total return the average of the two median values. For instance, for `double[][] a = {{2.0, 3.1, 1.0}, {5.0, -3.8, 6.0}};` the two elements in the middle are 2.0 and 3.1 (with -3.8 and 1.0 smaller and 5.0 and 6.0 bigger), that is, you should return the average of the two, which is 2.55.