CS102 Course Project, Spring 2018
(Part Ⅱ)

The goal of the second part is to generate the word cloud based on the results produced by the first part. Specifically, you are required to finish the following task:

Task 3 (Visualization): You should visualize the results of Task 2 by drawing the word cloud on the screen. The more often words appear in the text, the bigger they should be drawn (note that the relationship isn't necessarily linear). You should design a placement that avoids the overlap of words (the `getBounds` method of our provided `TextBoard` class may be helpful, see appendix). For image drawing, you can either set up a default color scheme (for example, using the same color for all words) or generate colors randomly. If you choose to do the latter, you should make sure that the color contrast is good enough, and that the hue of the words that have a similar size is also significantly different.

For Task 3, you will get a basic score if (1) the word cloud is correctly drawn, (2) there is no overlap of words, (3) the words have appropriate font sizes. It is fine if you only draw horizontal lines. You will get higher scores if your programs fulfill the following advanced requirements:

- The visualization algorithm is creative and produces an image with an artistic look. For example, you may use the overloaded `write` methods of `TextBoard` to set different background colors or rotation angles when drawing different words.
- The word cloud can be saved to disk as a JPG or PNG file. The `java.awt.image.BufferedImage` and `javax.imageio.ImageIO` classes may be helpful.
- The background color of the whole word cloud image can be configured by passing RGB color values (https://en.wikipedia.org/wiki/RGB_color_model) as command-line arguments.

Appendix: Graphical User Interface (GUI)

As GUIs are not part of the curriculum of this course, you are provided with a `TextBoard` class that writes text in different ways in a window on the screen. The program is well-documented and you can understand the functionalities after reading the code and comments. A test program `TestTextBoard`, which demonstrates the common uses of `TextBoard`, is also provided to help you get started.

You may find these links helpful (PHP code isn't hard to read when you know Java):

https://24ways.org/2010/calculating-color-contrast/

https://www.splitbrain.org/blog/2008-09/18-calculating_color_contrast_with_php

In addition, you can find many websites generating word clouds on the web, try to get inspiration from them.