Exercise 1 - October 18th, 2018
Getting Started

Preface: Plagiarism

„Plagiarism is the "wrongful appropriation" and "stealing and publication" of another author's "language, thoughts, ideas, or expressions" and the representation of them as one's own original work.“ [https://en.wikipedia.org/wiki/Plagiarism]

All exercise submissions will be checked for signs of plagiarism, both automatically and manually. Any case of plagiarism will result in expulsion from the class for all team members.

Summary: *you have to write your exercise submissions on your own* (in collaboration with your teammate). *If you use any external sources or help, **including other students**, you have to cite them. **IMPORTANT: this also includes Stack Overflow!***

Prerequisites – Github

- Create a new account at https://github.com/ (if you don't already have one).
- (Optional) Follow the tutorial at https://guides.github.com/activities/hello-world/.
- Fork the repository at https://github.com/mmbuw/se-ws18-exercise-1
- Clone the newly forked repository to your local machine(s).
- Use GitHub throughout the class to collaborate, send pull requests, fix issues, …
- Make sure that e-mail from GitHub is delivered to a regularly-read account!

Prerequisites – Java & Make

- If necessary, install the Java Development Kit (JDK 11, https://jdk.java.net/11/) and Make (http://gnuwin32.sourceforge.net/packages/make.htm) on your computer.
- If you are using a university pool computer, these tools are already installed.

Part 1 – Basics

In your newly forked repository, create a new class `Image` with the following attributes:

- A *public* byte array to store raw 24-bit RGB image data (3 bytes per pixel).
- A constructor `Image( int width, int height )` that creates an `Image` object with the specified width and height.
- A method `set( int x, int y, int val )` which sets a single pixel at position `(x,y)` to the RGB value represented by `val` (Note: use zero-based indexing, i.e. the upper left corner pixel has position (0,0). Note: only consider the lower 24 bits of `val`).
- A method `write( String filename )` which writes the image data to a file represented by `filename`. As image format, use the trivial PPM format with binary encoding (see http://netpbm.sourceforge.net/doc/ppm.html for the file structure).

Part 2 – Testing

The repository contains a minimal test framework (usage example in `TestSuite.java`). All tests are executed automatically when you run `make`.

- There are 4 sample test cases that your `Image` class must pass.
- Complete the test method `ImageTest3` in `TestSuite.java`.
  (Hint: use 4 assert statements similar to `ImageTest2`).

Deliverable

On Github, create a pull request with the requested features by Thursday, November 1st, 12:00am (noon). Documentation is available at https://help.github.com/articles/using-pull-requests/. No commits created after this date will be considered. In the comment, specify the full name of each team member (max. 2).

Scoring

- A pull request has been submitted on Github: **1 point.**
- `Image.java` compiles when built via `make`: **1 point.**
- Submission passes all 3 predefined test cases: **1 point.**
- `ImageTest3` has been finished correctly: **1 point.**
- `write(...)` method creates a valid PPM file: **1 point.**
- Submission passes additional visual tests of PPM file: **1 point.**

  Total: **6 points.**