

The (Non-Exhaustive) Java Style Guide

December 15th, 2017

This document gives an overview of the most common conventions for good Java style, which we follow in this lecture. This style guide should help you to increase the readability and understandability of your program. Based on this style guide, we grade the code quality of your lab submissions.

Please note: This list of conventions is **not** complete. There may be other conventions to follow, but we listed those being most important to us.

Naming

- constants in all uppercase letters, compound words with underscore
- camel-casing for variables and methods
- camel-casing for classes (first letter upper case)
- speaking names
- short names (variables, methods)

Styling

- use of appropriate loops (for, while, do-while)
- code reuse (abstractions, methods, etc.)
- concise (short) methods, where possible
- scope-reduction of variables (e.g., don't use global variables unless required)

Organization

- order of program parts (package name, imports, instance variables, constructor(s), methods)
- correct and consistent indentation (do **not** use tabs, instead use white spaces)
- comments where required or helpful to understand code (do **not** use obsolete or irritating comments)
- proper access modifiers (encapsulation), e.g. keep instance variables private
- exceptions for error handling

Additional Recommendations

- braces for all blocks (i.e., also for blocks with a single statement)
- Javadoc compatible comments

If you are thrilled of Java conventions now, you might check the [Google Java Style Guide](#).