**Code Quality Module Practical Task**

**Necessary Tools**

Java Development Kit 11+

Apache Maven 3.6.0+

Git 2.24+

Docker

##Task

(1 star)

**1. Run Sonarqube docker container**

- Instruction: https://docs.sonarqube.org/latest/

**2. Login to Sonarqube web interface**

- default password: admin

**3. Configure a new project with token**

Graphical user interface, text, application

Description automatically generated

**4. Analyse your pet project with SonarQube. Use generated token for it (you can get project from previous modules)**

- Instruction: <https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-maven/>

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

**5. If there are any issues, correct it**

**6. Prepare .doc report with screenshot for every step**

(2-3 stars)

**7. Configure a new project with token**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**8. Create a new maven project with a few (10+) issues, e.x. :**

- Don't use method reference

- Use switch operator. Define 'default' in the top of operator

- etc.

Text

Description automatically generated

**9. Analyse your project**

Graphical user interface, text, application, email

Description automatically generated

**10. Try to skip all issues with @SuppressWarnings**

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**11. Prepare .doc report with screenshot for every step**

(4 stars)

**12. Remove @SuppressWarnings for project from previous section**

**13. Create a new Sonar java profile (copy it from default java profile)**

Graphical user interface, application

Description automatically generated

**14. Deactivate rules for issues from previous section**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**15. Link new profile to your project**

Graphical user interface, application

Description automatically generated

**16. Analyse your project**

Graphical user interface, application, website

Description automatically generated

**17. Prepare .doc report with screenshot for every step**

(5 stars)

**18. Create a new Quality Gate with:**

- Duplicated code (> 10)

- Issues (> 20)

- Code smells (> 40)

- Bugs (> 35)

- Unit test succsess (< 40)

Graphical user interface, text, application

Description automatically generated

**19. Link it to your project**

Graphical user interface, application, website

Description automatically generated

**16. Analyse your project**

A screenshot of a computer

Description automatically generated

**17. Prepare .doc report with screenshot for every step**

**References**

* [Clean Code](https://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882)
* [Agile Software Development: Code Quality](https://www.linkedin.com/learning/agile-software-development-code-quality/why-code-quality-is-important?u=2113185)
* [11 Best Practices and Tools to Improve the Java Code Quality](https://www.romexsoft.com/blog/improve-java-code-quality/)
* [S.O.L.I.D](https://www.baeldung.com/solid-principles)