

## Laboratory

### Coding Standar

**Date:** 21/06/2020

**Student:** Neythan León Vásquez

**Teacher:** Ing. Mónica Villavicencio.

**Repository:** <https://github.com/NeythLevanest/CodingStandarTest>

#### Objectives:

- Use Checkstyle as a code quality metric tool to detect and enforce Java coding standards.
- Obtain a html report with the different detected issues in the code using Maven.

#### Requirements

- Eclipse IDE for Java Developers >= 2019-09 R (4.13.0) • Git

#### Introduction

Checkstyle is an open source tool that enforces coding conventions and best practice rules for Java code. It automates the process of checking Java code to spare humans of this boring (but important) task. This makes it ideal for projects that want to enforce a coding standard. [1]

#### Prerequisites

1. Install Maven for Eclipse.
2. Install the Eclipse Checkstyle plug-in

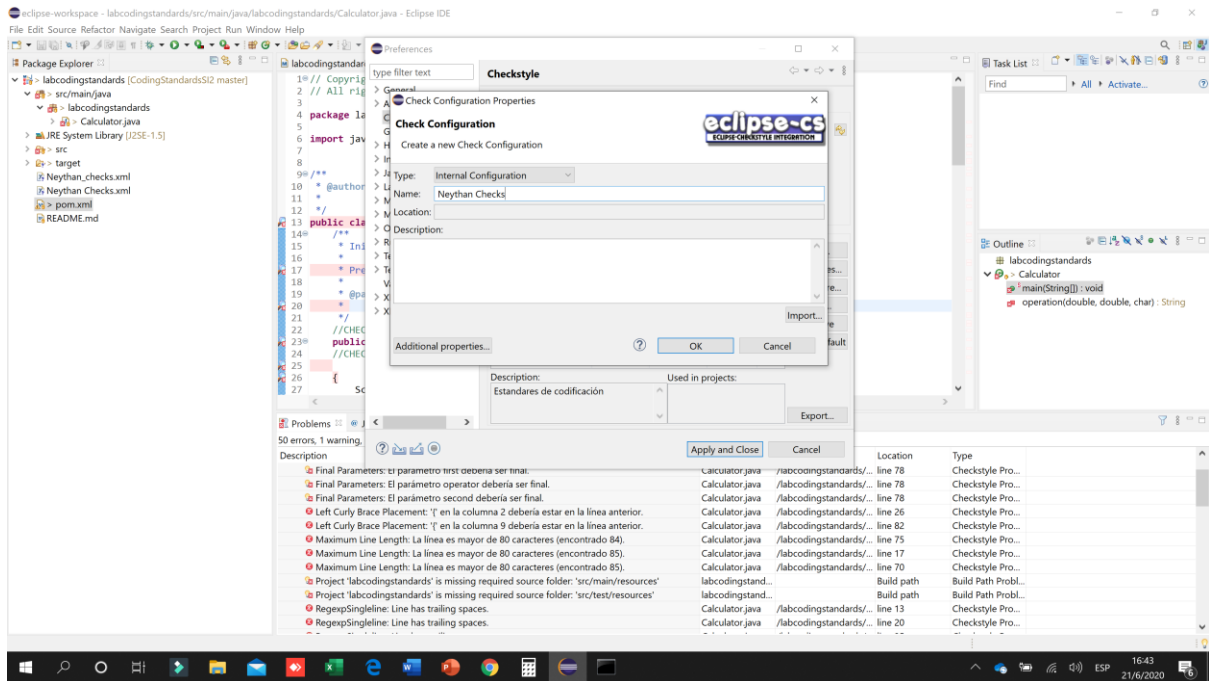
The latest version of eclipse brings Maven already installed: <https://www.eclipse.org/downloads/>

#### Activity

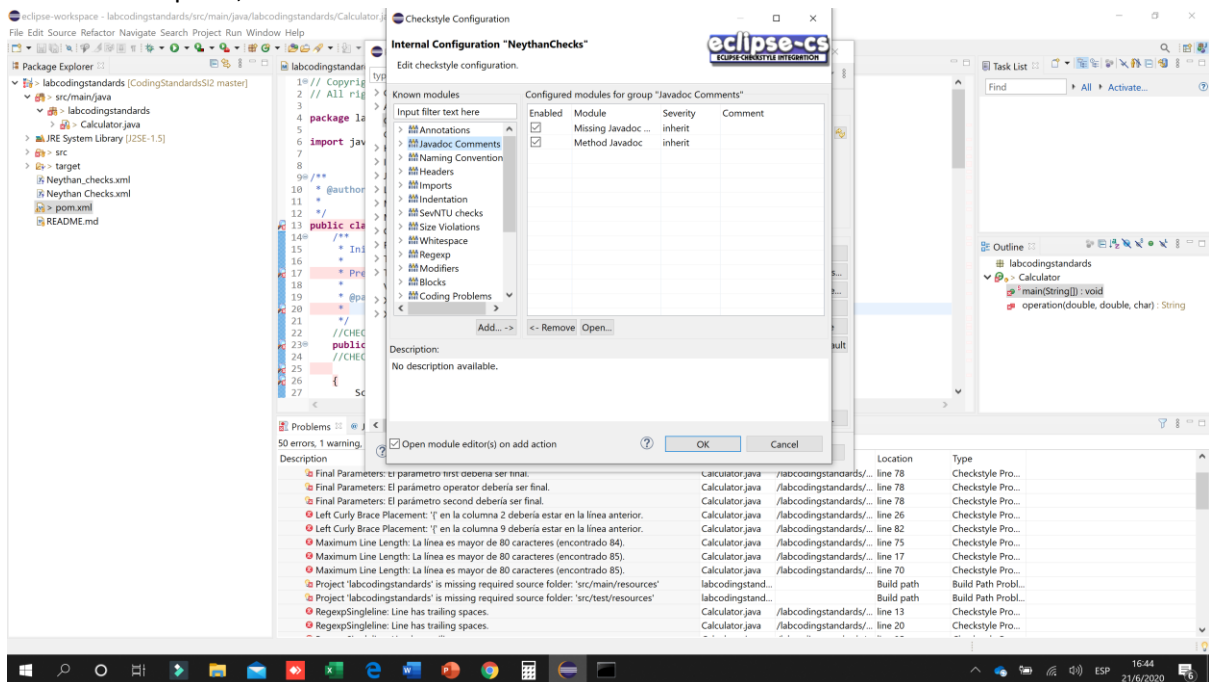
##### Configure Checkstyle

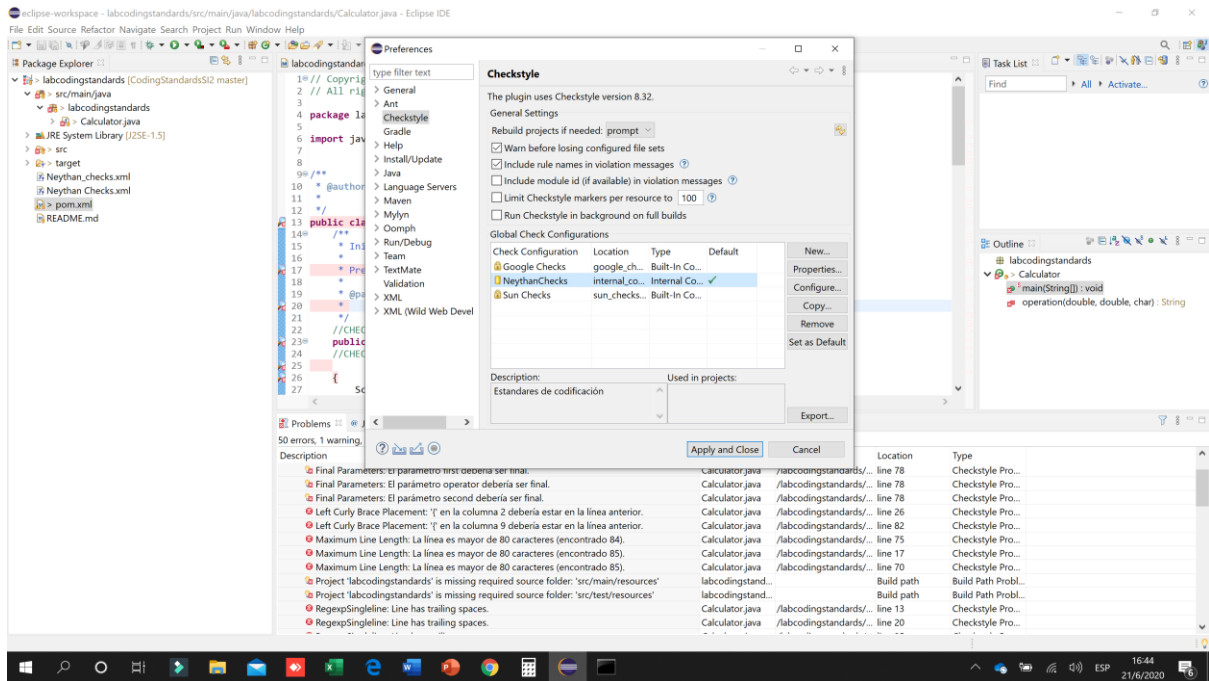
Checkstyle gives the option to create your own custom set of coding standards that is specifically design for your project. To do so, follow these steps:

1. Go to "Window → Preferences" and click "Checkstyle" on the side menu.
2. Check "Include rule names in violation messages" in General Settings
3. Click "New" to start creating a configuration file. 4. Select "Internal Configuration" under Type.
4. For the Name, type "Neythan Checks" and add a Description.



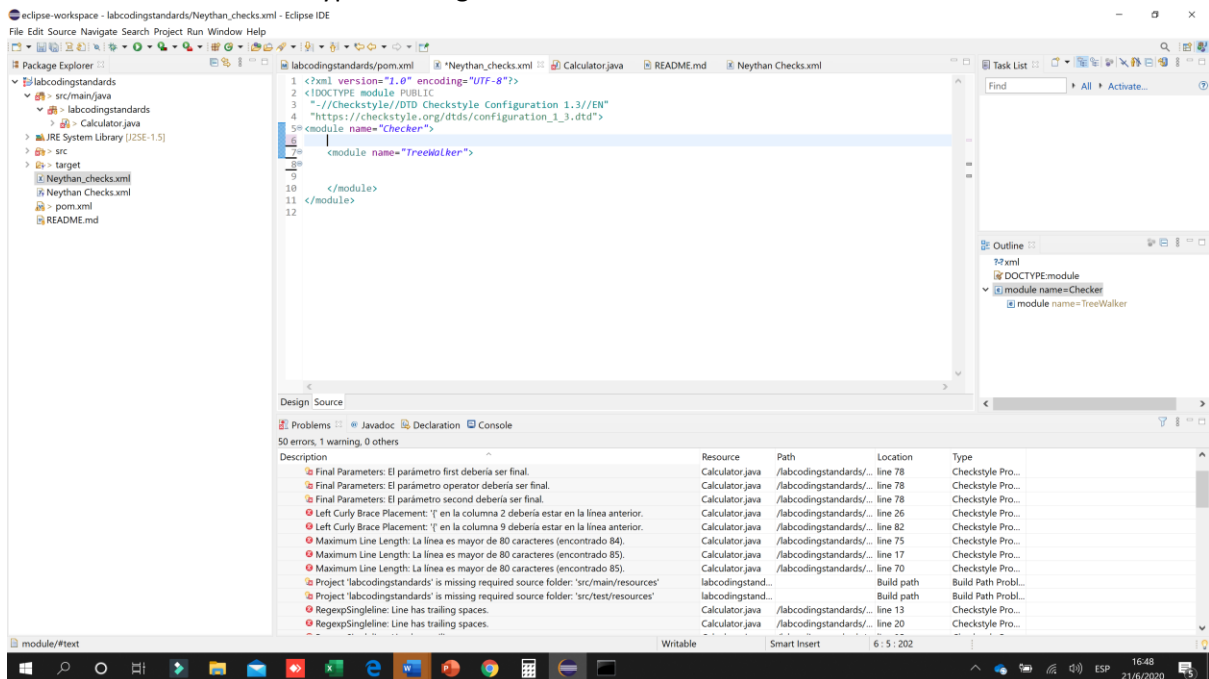
5. Click on "OK" to create the file.
6. The configuration file will be ready to add new rules. Double click on it or click "Configure"
7. All available Checkstyle modules are displayed in groups and are ready to be added to the project.
8. Let's add some rules. Under "Javadoc Comments", click "Missing Javadoc Method", read the description, then click "Add... ->"



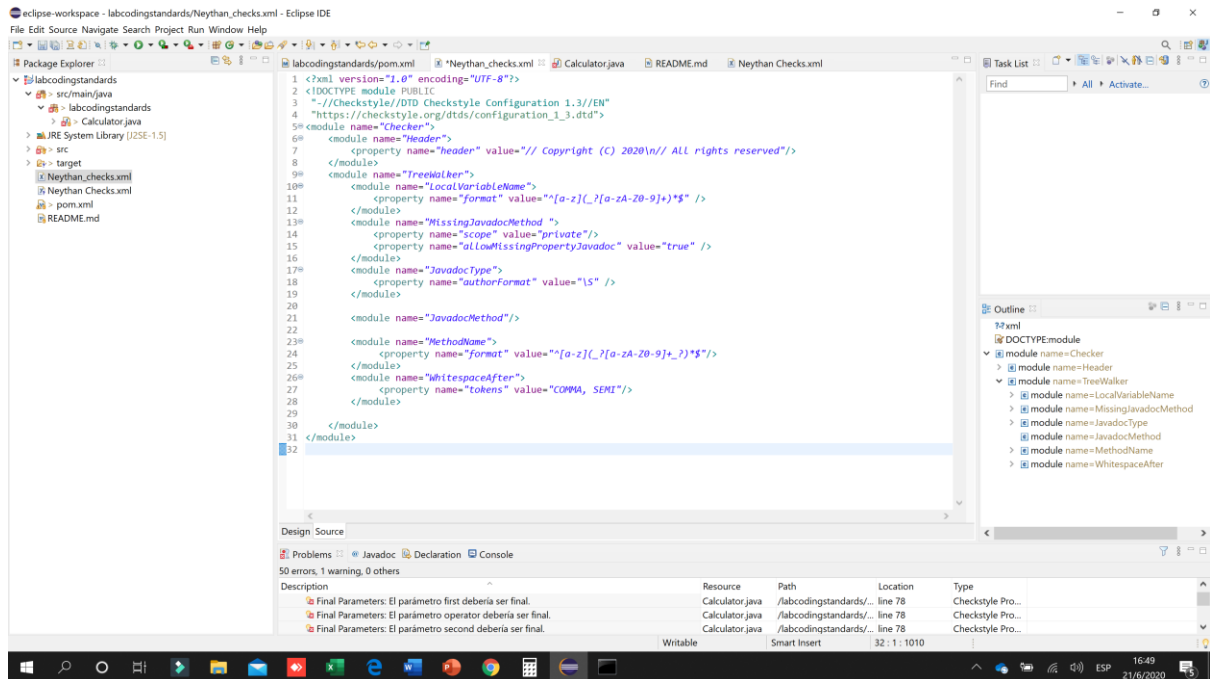


## Configure Checkstyle using the XML

1. Create a new file with name as "Neythan\_checks.xml" in my case.
2. Add the next typical configurations document.

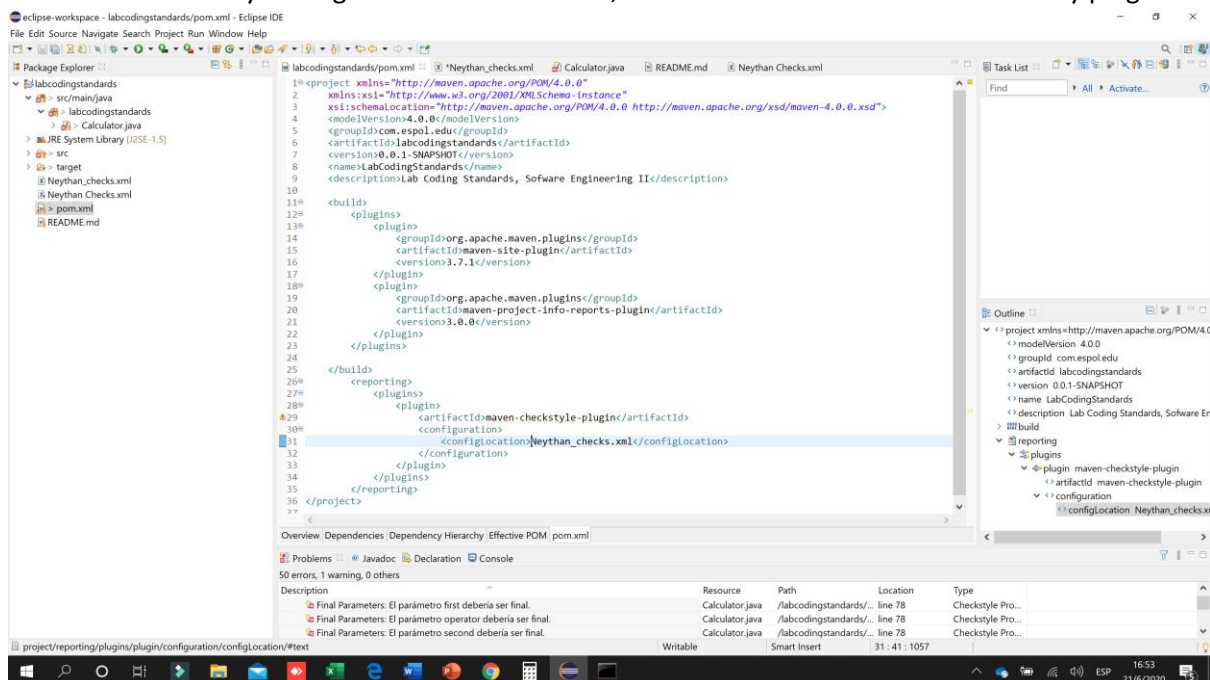


3. Add all module configuration needed to this practice. For this first part, you should omit `javacMethod`, `MethodName`, and `WhitespaceAfter` modules to the first test running Maven.



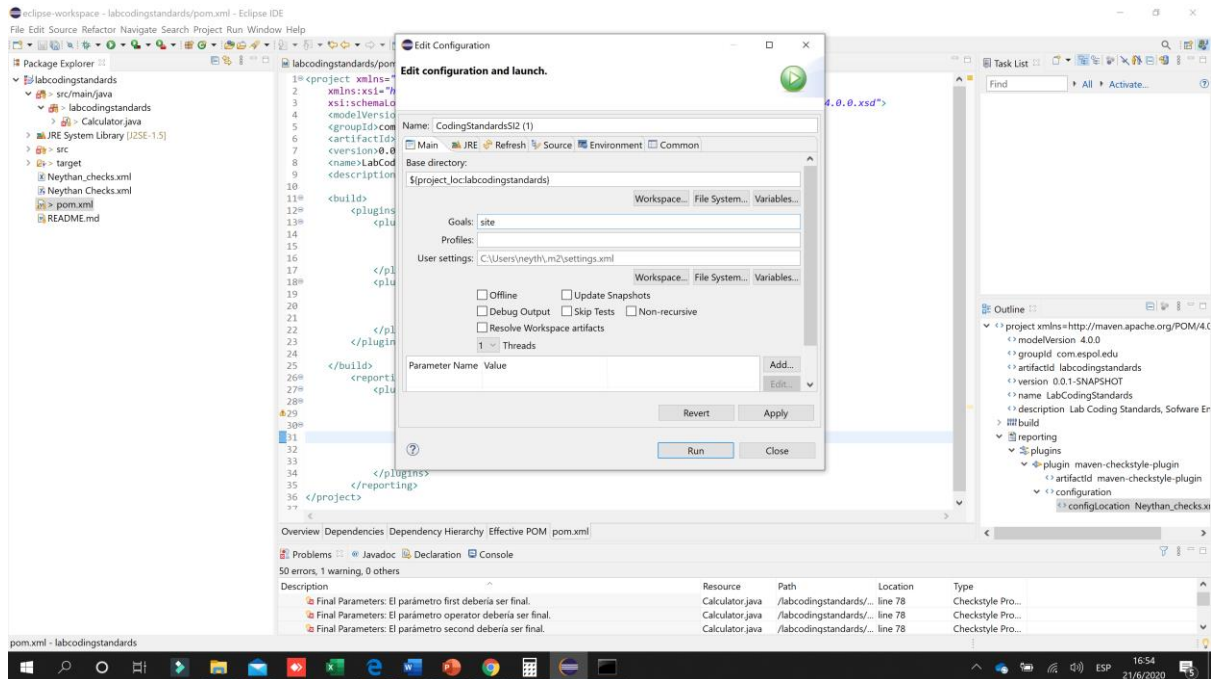
## Checkstyle with Maven

Checkstyle integrates well with Maven, but first we need to add the necessary plugins.



The configLocation tag is used to specify what file to use in the checking process. If omitted, the selected file in the Checkstyle configuration window will be used.

1. Save the file, right Click on it, then “Run as → Maven build...”.
2. Type “site” for the Goals and Click “Run”.



3. Wait for the process to finish.
4. Go to your project directory, open "target→ site".
5. Several html files are show. Open "index.html".
6. Click on "Project Reports → Checkstyle" to see a detailed Checkstyle report of your project.
7. Add the rules that before you omitted "javadocMethod, MethodName, and WhitespaceAfter" and test new again

LabCodingStandards - Checkstyle

Archivo C:/Users/neyth/Escritorio/TallerEspol/CodingStandardsS12/INFORME-PRE-CORRECCIÓN/site/checkstyle.html

## LabCodingStandards

Last Published: 2020-06-21 | Version: 0.0.1-SNAPSHOT

**Project Documentation**

- Project Information
- Project Reports
- Checkstyle

Built by **maven**

### Checkstyle Results

The following document contains the results of Checkstyle 8.29 with Neythan\_checks.xml ruleset. [View](#)

#### Summary

Files	Info	Warnings	Errors
1	0	0	10

#### Files

File	I	W	E
labcodingstandards/Calculator.java	0	0	10

#### Rules

Category	Rule	Violations	Severity
header	Header	1	Error
javadoc	JavadocType	1	Error
	MissingJavadocMethod	2	Error

## Details

labcodingstandards/Calculator.java

Severity	Category	Rule	Message	Line
Error	header	Header	Line does not match expected header line of '// Copyright (C) 2020'.	1
Error	javadoc	JavadocType	Type Javadoc comment is missing @author tag.	13
Error	javadoc	MissingJavadocMethod	Missing a Javadoc comment.	15
Error	naming	LocalVariableName	Name 'First' must match pattern '^[a-z][_?a-zA-Z0-9]+*\$'.	23
Error	whitespace	WhitespaceAfter	',' is not followed by whitespace.	52
Error	whitespace	WhitespaceAfter	',' is not followed by whitespace.	52
Error	javadoc	MissingJavadocMethod	Missing a Javadoc comment.	58
Error	naming	MethodName	Name 'Operation' must match pattern '^[a-z][_?a-zA-Z0-9]+_*\$'.	58
Error	whitespace	WhitespaceAfter	',' is not followed by whitespace.	58
Error	whitespace	WhitespaceAfter	',' is not followed by whitespace.	58

Copyright © 2020. All rights reserved.

## 8. Correct any errors/warnings in the code that have been generated in the report.

```

import java.util.Scanner;

/**
 * @author Neythan León
 */
public class Calculator {
    /**
     * Initialize the program Calculator.
     * Precondition: 'args' should contain at least one element, the user's name.
     * @param args the command-line arguments.
     */
    //CHECKSTYLE:OFF
    public static void main(String[] args)
    //CHECKSTYLE:ON
    {
        Scanner reader = new Scanner(System.in);
        System.out.print("1. +\n2. -\n3. *\n4. /\n\nEnter an operator: ");
        char operator = reader.nextLine().charAt(0);
        double first;
        double second;
        String input;

        while (true) {
            System.out.print("Enter first number: ");
            input = reader.nextLine();
            try {
                first = Integer.parseInt(input);
                break;
            } catch (NumberFormatException e) {}
        }
    }
}

```

Problems: 50 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
Final Parameters: El parametro first debería ser final.	Calculator.java	/labcodingstandards/...	line 78	Checkstyle Pro...
Final Parameters: El parametro operator debería ser final.	Calculator.java	/labcodingstandards/...	line 78	Checkstyle Pro...
Final Parameters: El parametro second debería ser final.	Calculator.java	/labcodingstandards/...	line 78	Checkstyle Pro...

## 9. Generate a new report. It should not have any errors/warnings

LabCodingStandards - Checkstyle

LabCodingStandards - Checkstyle

+

← → ↻ ⓘ Archivo | C:/Users/neythv/Escritorio/TallerEspol/CodingStandards/S12/INFORME%20CON%20CORRECCIONES/site/checkstyle.html

🔍 ☆ 🛑 🌐 📄 📁 🌱 ⋮

# LabCodingStandards

Last Published: 2020-06-21 | Version: 0.0.1-SNAPSHOT


LabCodingStandards

Project Documentation

▶ Project Information

▶ Project Reports

Checkstyle

Build by  maven

## Checkstyle Results

The following document contains the results of [Checkstyle](#) 8.29 with Neythan\_checks.xml ruleset. 🔗

### Summary

Files	Info	Warnings	Errors
1	0	0	0

### Files

File	I	W	E
------	---	---	---

### Rules

Category	Rule	Violations	Severity
----------	------	------------	----------

### Details

Copyright © 2020.All rights reserved.

Windows icons

Taskbar icons

17:00  
21/6/2020