



Mestrado em Engenharia Informática e Computação

Teste, Verificação e Validação de Software

2021/2022 | 1º Semestre

STATIC TESTING

Practical Exercises

Ana Teresa Dias Silva | up201606703@edu.fe.up.pt

Maria Gonçalves Caldeira | up201704507@edu.fe.up.pt

Introduction

The following guide presents detailed information about the practical exercises for the Static Testing class. Before starting, it is necessary to set up the work environment, as explained in detail [here](#).

Snake program

These exercises are related to the well-known Snake Game.

There are two classes: *Board* (responsible for game board creation and management) and *Snake* (where the game is defined).

Issues Types and Severities in SonarLint

There are three types of issues detected by SonarLint:

1. **Bug** – A coding mistake that can lead to an error or unexpected behavior at runtime.
2. **Vulnerability** – A point in your code that's open to attack.
3. **Code Smell** – A maintainability issue that makes your code confusing and difficult to maintain.

Each issue has one of five severities:

1. **BLOCKER**
Bug with a high probability to impact the behavior of the application in production: memory leak, unclosed JDBC connection, etc. The code **MUST** be fixed immediately.
2. **CRITICAL**
Either a bug with a low probability to impact the behavior of the application in production or an issue that represents a security flaw: empty catch block, SQL injection, etc. The code **MUST** be immediately reviewed.
3. **MAJOR**
A quality flaw that can highly impact the developer productivity: an uncovered piece of code, duplicated blocks, unused parameters, etc.
4. **MINOR**
A quality flaw that can slightly impact the developer productivity: lines should not be too long, "switch" statements should have at least 3 cases, etc.
5. **INFO**
Neither a bug nor a quality flaw, just a finding.

Snake Issues

There are 54 issues in the base code: 49 on Board.java and 5 on Snake.java.

On *Board*, they are distributed in the following way:

- **40 code smells:**
 - Minor: 23
 - Major: 13
 - Critical: 4
- **9 bugs:**
 - Minor: 1
 - Major: 6
 - Blocker: 2

On *Snake*, there are only code smells:

- **Minor code smell:** 2
- **Major code smell:** 3

The Exercise

Considering the number of issues and the static testing context, it is difficult to define a concrete order to solve the errors. Therefore, we built a single exercise divided into several points.

As important as to solve all the issues, you should try to understand all of them and their possible implications on the program: read the title and the description and, if you have any doubts, look at the example code available on SonarLint.

If SonarLint does not automatically update when you solve an issue, you should do it manually.

First Step

Start by solving the issue on *Board.js*, at line 23, already solved in the demonstration.

Second Step

There is an issue common to both files: "The default unnamed package should not be used." Begin creating a package named "SnakeGame" and moving both files into it. How many errors do you get?

Hint: Try to rename the package to something like "snake_game"!

Third Step

Although the issues with a higher severity are a priority, we suggest that you start by solving all the minor bugs and code smells.

Then, try to fix the major issues. At last, solve the critical and blocker issues.

Try to play

Now that you are issue-free, test your ability to play the Snake Game and have fun!