POLITECNICO
MILANO 1863

**Politecnico di Milano**

COMPUTER SCIENCE AND ENGINEERING

Software Engineering 2

# Code Inspection

**PowerEnJoy**

Authors:
**Francesco Fabiani**
**Jagadesh Manivannan**
**Niccolò Pozzolini**

Professors:
**Elisabetta Di Nitto**
**Luca Mottola**

**Version 1.0** **February 5th 2017**

# Contents

# Chapter 1

# Description of the code

## 1.1 Revision history

| Version | Date | Authors | Summary |
|---|---|---|---|
| 1.0 | 05/02/2017 | Fabiani, Manivannan, Pozzolini | Initial release |

Table 1.1: Changelog of this document

# Chapter 2

# Results of inspection

This chapter contains the results of the code inspection that we did on the assigned class and methods. All the points of the checklist reported in the assignment were checked, and we also found other bad practices not listed in the checklist.

## 2.1 Notation

- The items of the checklist reported in the assignment will be referred as **C1**, **C2**, ..., while the general errors will be indicated with **Gen1**, **Gen2**, ...
- A specific line of code will be referred as follows: L.1234.
- An interval of lines of code will be referred as follows: L.1234-1289.

## 2.2 Report

### 2.2.1 Checklist

1. **C17** L.161
   The "else if" statement is not aligned with the beginning of the expression at the same level as the previous line.
2. **C17** L.132 L.88 L.179 L.186 L.196 L.205 L.276 L.308 L.430 L.462
   The "catch" statement is not aligned at the same level. This could be avoided either during the code development by proper checking.

3. **C17** L.189 L.350 L.360 L.506 L.516
   The "else" statement is not aligned at the same level. This could be avoided either during the code development by proper checking.

4. The package and import statements were found in the correct order as per the code inspection document.

5. The class or interface declarations order is followed correctly.

6. **C27** L.224 L.378
   Both the "switch" blocks are a copy paste in our service which paves way for duplicates. But this is quite preferable here rather than using different piece of code for the same logical implementation (i.e. for a better understanding).

7. **C27** L.218 L.271 L.301 L.344 L.372 L.425 L.455 L.500
   All the methods are long more than 10 lines and some methods are even more than 20 lines. This should be optimized by using fewer conditional checks covering all the requirements.

8. **C26**
   All the methods are grouped by functionality rather than by scope or accessibility.

9. The error messages are comprehensive and are in-line as per the code inspection document.

10. All switch statements are addressed by a break.

11. **C55** L.224 L.378
    There is no default branch for any of the switch statements. The solutions we suggest for this problem is to leave the code undisturbed, because all the days are present in the switch cases such that it executes any one of the switch case and so no default is needed. Without knowing the dayStart or dayEnd it would be wrong to define a default branch here.

### 2.2.2 General

Other errors found during the code inspection:

12. **Gen1** L.261 L.284 L.317 L.328 L.415 L.438 L.471 L.483
    No import statement used for Integer. Import that has to be used here was: java.lang.Integer or import java.lang.* can be used.

13. **Gene2** L.221 L.287 L.356 L.375 L.441 L.472 L.512
    No import for Double was done: java.lang.Double or import java.lang.* could have solved the issue.

14. **Gen3** L.418 L.448 L.264 L.394 L.336
    TechDataCalendar was spelled wrong here within the comments. The final letter is missing.