

Sailor - Pioneers of the digital entrepreneurship

Coding Standards

Revision History

Version	Modifier	Date	Description of Change
0.1	E. Ricca	14/04/2020	Initial rough draft.

Table of Content

1. Introduzionepag.1
2. Source File Basicspag.1
3. Code Formattingpag.3
4. Declarationspag.4

1. Introduction

This document serves as the definition of Sailor's general coding standards for source control in the different kind of programming languages.

2. Source file basics

1. File name

Every file must be named following these rules:

- The name has to be in "*lowerCamelCase*":
`fileName`, `test01`, `testFileOfFunction`
- The name has to be significant:
`featureX`, `homePage`
- File names can't start with numbers

2. Source file Control

A source file consists of (*in order*):

- License or copyright informations, if present
- Package statement
- Import statements
- Exactly one top-level class
- Exactly one blank line separates each of the present section

3. Code Formatting

1. Braces

- Braces use: Braces are used with every statement, when their programming language is structured to do so. Braces must be used even if the body is empty or contains only a single line
- Braces style: Braces must follow the *"Egyptian brackets"* style or *Kernighan and Ritchie* style

```
if(condition){  
    //do this  
}
```

2. Horizontal Alignment

Terminology Note: *Horizontal Alignment* is the practice of aligning certain parts of the code with the following parts. Horizontal Alignment is permitted but not required.

```
int a = 0; //not aligned  
int b = 0; //not aligned
```

3. Spacing Between Lines

When the programming allows it, the programmer must put empty lines like the following:

- 2 empty lines between each method
- 1 empty line between logically different blocks

```
int a = 0  
int b = 0  
methodA(){  
    //do this
```

```
}
```

```
methodB(){  
    methodA()  
    //do that  
}
```

4. Declarations

1. **Variables**

Variables must be declared in "*lowerCamelCase*" and the name must be significant.

2. **Methods and Classes**

Methods and classes must be declared in "*lowerCamelCase*" and the name must be significant.