Code Style Specification

Development tool
 IDE Eclipse

Custom language (functions names, comments…)
English

File names
The class name should begin with capital letter, following the CamelCase rule.

    Eg.: SomeClass.java

Class header
It must have a comment following the template below. Every new line must begin with an upper case.

    /**

    ClassName

    Description of the class function

    Link to the class at GitHub

    */

File Structure
Every class must follow the structure below.

1. Header;
2. Class or interface declaration;
3. Public, protected and private static variables respectively;
4. Public, protected and private instance variables respectively.;
5. Constructors;
6. Public, protected and private methods, respectively.

There must be a blank line between the package declaration and imports.

> Eg.:
> package example;
>
> import java.util.*;
> import java.another.import;

Indentation

The TAB must be used, always with four scapes width.

The line width should be 80 characters. Line breaks must be done always after comma and before operators. The indentation of the broken line shall be two TABs (eight spaces) after the current indentation (even if the line becomes greater than 80 chars).

Comments

Block comments, when necessary, must follow the template below.

> /*
> That's a multiline comment.
> In this case, it is necessary.
> */

One line comments must follow the template below.

> // That's just an one line comment.

Declarations

Variable declarations must be done in separated lines.

> Eg.:
> int clientAge;
> int clientCode;
> String clientName;
> String clientAddress;

Functions, classes, or interfaces declarations must be done following the template below. There must not be a space between opening parenthesis and the function, class or interface name. Brackets must be opened in the same line of declaration, and closed in an exclusive line, at the same indentation level. There must be a blank line after the declaration.

```
class SomeClass extends Object {

        public SomeMethod(int firstArgument, int secondArgument) {

                // Some fancy calculations.
        }
}
```

Control and/or repetitions structures

The declarations of this structures should be done following the template below. When this structures trigger just one command line, the block should be explained as always. If the structure does nothing, it must have an empty structure comment.

```
for (int i = 0; i <= 10; i++) {
        // This structure does nothing.
}
```

Nested *if* structures should follow the template above.

```
if (booleanExpression) {
        // Does nothing.
} else {
        if ( anotherExpression ) {
                // Does nothing too.
        } else {
                // Does nothing. Too.
        }
}
```

Methods

All global variables must be accessed using *this*.

```
Eg.: this.globalVariable = newValue.
```

Every function must begin with a description comment.

```
Eg.:
// This function modifies a client.
public void modifyClient() {

        // Code that modifies a client
}
```