

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

Should have a comment following the template above. Every new line must begin with capital letter.

```
/**  
  ClassName  
  Description of the class function  
  Link to the class at GitHub  
*/
```

File Structure

Every class should follow the structure above.

1. Header;
2. Class or interface declaration;
3. Static variables, publics, protecteds and privates, respectively;
4. Instance variables, publics, protecteds and privates, respectively.;
5. Constructors;
6. Methods.

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 should be used, always with four spaces width.

The line width must be 80 characters. Line breaks must be done always after comma and before operators. In case of parenthesis, should be avoided to break lines between them.

The indentation of the broken line should be two TABs (eight spaces).

Comments

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

```
/*
```

```
That's a multiline comment.
```

```
In this case, it is necessary.
```

```
*/
```

One line comments must follow the template above.

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

Declarations

Variable declarations should be done in separated lines.

Eg.:

```
int clientAge;
```

```
int clientCode;
```

```
String clientName;
```

```
String clientAddress;
```

Functions, classes, or interfaces declarations should be done following the template above. Should not be a space between opening parenthesis and the name. The block should be opened in the same line of the declaration, and closed in an exclusive line, in the same indentation level. Should 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 above template. When this structures trigger just one command line, the block should be explained as always. If the structure does nothing, must a have a comment saying this.

```

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 ( outraExpressao ) {
        // Does nothing too.
    } else {
        // Does nothing. Too.
    }
}

```

Methods

All global variables should be accessed using *this*.

Eg.: `this.globalField = newValue.`

Every function should begin with a description comment.

```

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

    // Code that modifies a client

}

```