**PSP - Java Coding Standard**

|  |  |
| --- | --- |
| Program Header | Java source files won’t have any header before package declaration (because no license, trademark, corporate or copyright information will be placed there) |
| Class Comments | * Class must be documented following JavaDoc standard and must be clearly defined what the class does or the responsibility that the class has in the program |
| Class Comments example | /\*\*  \* Class Specific description  \* @author Name of the author  \*  \*/ |
| Method Comments | * Referred as Java methods. * Must be documented following JavaDoc standard |
| Method Comments example | /\*\*  \* Brief Method Description  \* @param The parameters list and a description of each parameter (if any)  \* @return The return type and/or value (if not void).  \* @throws If an Exception is thrown by method (if needed)  \*/ |
| Identifiers | * Referred as the method or variable name * Must follow the Java naming standards and must use CamelCase * Use descriptive names and avoid abbreviations. * Method declarations must be placed in one line at least to the first opening parenthesis. * Opening and closing brackets must be placed in one single line at the end of the line. It means there must be only one bracket per line at the end. |
| Identifier Examples | int numberOfClauses (OK)  String customerName (OK)  int xy (No)  String var\_foo (No)  public void aMethod(String param){ (OK)  public void ->  -> anotherMethod(){ (No)  public method(){ code; { (No)  public methodd(){  code;  } (OK) |
| Comments | * Code must be documented briefly and in a concrete way to explain its behavior. Avoid over-comment inside methods or variables, unless needed because code complexity. * Comments must be placed in an independent line before the effective code (multiline and end-of-line comments) |
| Example | /\*  \* Multiline OK Comment  \*/  // End-Of-Line OK Comment  int customerBalance = 0 //He’s poor (Bad comment) |
| Blank spaces | * Programs must be accordingly spaced to avoid code to be seen as crowded. * At least one space must be set between Java elements (import and package statements, class, variable or method declarations) |
| Indenting | * Code must be indented after opening braces (next line) from the preceding level. * Open braces must be placed after function, method or class declaration. * Closing braces must be aligned with the line where piece of code (opening brace) was declared or open. |
| Indenting example | public void someMethod(){  if(someCondition == true){  doSomething();  }  } |
| Capitalization | * Constants must be all in uppercase, separated by underscore sign ( \_ ) * All other identifiers must be CamelCase notation. |
| Capitalization Example | public static final String PI\_CONSTANT\_STRING = “Pi” |

**PSP Counting Standard**

|  |  |  |
| --- | --- | --- |
| **Count Type** | **Type** | **Comments** |
| Physical / Logical | Logical | Only effective lines will be counted (no blanks or lines inside comments) |
| **Statement Type** | **Included (Y/N)** | **Comments** |
| **Executable** | Yes |  |
| **Non-Executable** |  |  |
| Declarations | Yes |  |
| Compiler directives | Yes | Package and Import statements |
| Comments | No |  |
| Blank lines | No |  |
| **Clarifications** |  | **Examples / Cases** |
| Methods |  | Methods will be counted as one according to coding standard |
| **Notes** |  |  |
| Note 1 |  | Method Arguments can be splitted into more than one line. |