

## 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	<ul style="list-style-type: none"><li>- 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</li></ul>
Class Comments example	<pre>/**  * Class Specific description  * @author Name of the author  *  */</pre>
Method Comments	<ul style="list-style-type: none"><li>- Referred as Java methods.</li><li>- Must be documented following JavaDoc standard</li></ul>
Method Comments example	<pre>/**  * 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)  */</pre>
Identifiers	<ul style="list-style-type: none"><li>- Referred as the method or variable name</li><li>- Must follow the Java naming standards and must use CamelCase</li><li>- Use descriptive names and avoid abbreviations.</li><li>- Method declarations must be placed in one line at least to the first opening parenthesis.</li></ul>
Identifier Examples	<pre>int numberOfClauses (OK) String customerName (OK) int xy (No) String var_foo (No) public void aMethod(String param){ (OK) public void -&gt;     -&gt; anotherMethod(){ (No)</pre>
Comments	<ul style="list-style-type: none"><li>- 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.</li><li>- Comments must be placed in a independent line before the effective code (multiline and end-of-line comments)</li></ul>
Example	<pre>/*</pre>

	<pre> *   Multiline OK Comment */ // End-Of-Line OK Comment int customerBalance = 0 //He's poor (Bad comment) </pre>
Blank spaces	<ul style="list-style-type: none"> <li>- Programs must be accordingly spaced to avoid code to be seen as crowded.</li> <li>- At least one space must be set between Java elements (import and package statements, class, variable or method declarations)</li> </ul>
Indenting	<ul style="list-style-type: none"> <li>- Code must be indented after opening braces (next line) from the preceding level.</li> <li>- Open braces must be placed after function, method or class declaration.</li> <li>- Closing braces must be aligned with the line where piece of code (opening brace) was declared or open.</li> </ul>
Indenting example	<pre> public void someMethod(){     if(someCondition == true){         doSomething();     } } </pre>
Capitalization	<ul style="list-style-type: none"> <li>- Constants must be all in uppercase, separated by underscore sign ( _ )</li> <li>- All other identifiers must be CamelCase notation.</li> </ul>
Capitalization Example	<pre> public static final String PI_CONSTANT_STRING = "Pi" </pre>

## 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
<b>Executable</b>	Yes	
<b>Non-Executable</b>		
Declarations	Yes	
Compiler directives	Yes	Package and Import statements
Comments	No	
Blank lines	No	
<b>Clarifications</b>		<b>Examples / Cases</b>
Methods		Methods will be counted as one according to coding standard
<b>Notes</b>		
Note 1		Method Arguments can be splitted into more than one line.