**findClassInternal(String name)**

1. findClassInternal(string className)

* parameter *name* does not give enough information about what it represents. (name -> className) (line 2727)
* variable entry -> resourceEntry (line 2736)
* clazz->classEntry (line 2742)
* pkg -> packageObject (line 2756)
* validate function is not specific enough. validate(String) ->validateClassName(string)

1. No issues
2. No issues
3. No issues
4. No issues
5. No issues
6. No issues
7. Indention:
   1. Line 2739: There are more than 7 spaces for indention. It should be reduced to 4.
   2. Line 2761 and 2775: There are just 2 spaces used for indention. It should be 3 or 4.
8. No issues
9. Allman style is used
10. Braces:
    1. Line2730: if-statement has just one statement and is not surrounded by curly brackets. Correct way should be:

if (!validate(name)) {

throw new ClassNotFoundException(name);

}

* 1. Line2738: if-statement has just one statement and is not surrounded by curly brackets. Correct way should be:

if (entry == null){

throw new ClassNotFoundException(name);

}

* 1. Line2743: if-statement has just one statement and is not surrounded by curly brackets. Correct way should be:

if (clazz != null){

return entry;

}

* 1. Line2747: if-statement has just one statement and is not surrounded by curly brackets. Correct way should be:

if (entry.binaryContent == null){

throw new ClassNotFoundException(name);

}

* 1. Line2754: if-statement has just one statement and is not surrounded by curly brackets. Correct way should be:

if (pos != -1){

packageName = name.substring(0, pos);

}

* 1. Line2791: if-statement has just one statement and is not surrounded by curly brackets. Correct way should be:

if (!sealCheck){

throw new SecurityException

("Sealing violation loading " + name + " : Package "

+ packageName + " is sealed.");

}

1. There are blank lines that don’t separate sections, those are probably present to separate things conceptually different. However this requirement is respected which means that sections are correctly separated by blank lines in the entire class.
2. No issues
3. No issues
   1. Line 2788 is separated before an operator “II”
   2. Line 2793 is separated before an operator “+”
4. No issues
5. No issues
6. Class and constructors are very good commented as well as the function *findClassInternal.* Code inside this function is very less briefly commented.
7. Comments does not contain dates
8. No issues
9. No issues
10. ??
11. ??
12. No issues
13. No issues
    1. No issues
    2. No issues
    3. No issues
    4. In class WebappClassLoader there are first declared private static variables instead of public ones. Lines 157, 160, 167.
    5. Instance variables are declared in wrong order in class WebappClassLoader. Starting from line 352
    6. No issues
    7. No issues
14. Methods are properly grouped by functionality
15. No issues
16. The variables and class members are properly defined and have correct visibility in a way that all of private types has getters and setters implemented for them.
17. No issue
18. No issue