**findClassInternal(String name)**

***Naming Conventions***

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

***Indention***

1. 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.
2. No issues

***Braces***

1. Kernighan and Ritchie style is used
2. 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.");

}

***File*** ***Organization***

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

***Wrapping*** ***Lines***

* 1. Line 2788 is separated before an operator “II”
  2. Line 2793 is separated before an operator “+”

1. No issues
2. No issues

***Comments***

1. Class and constructors are very good commented as well as the function *findClassInternal.* Code inside this function is very less briefly commented.
2. Comments does not contain dates

***Java*** ***Source*** ***Files***

1. No issues
2. No issues
3. ??
4. ??

***Package*** ***and*** ***Import*** ***Statements***

1. No issues

***Class*** ***and*** ***Interface*** ***Declerations***

1. 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
2. Methods are properly grouped by functionality
3. No issues

***Initialization*** ***and*** ***Declaration***

1. The variables and class members are properly defined and have correct visibility.
2. No issues
3. No issues
4. No issues
5. No issues
6. Declarations in lines 2733, 2734, 2736,2752, 2753 and 2757 are not on the beginning of the block

***Methods*** ***Calls***

1. No issues
2. No issues
3. No issue

***Arrays***

1. No arrays in this function
2. No arrays in this function
3. No arrays in this function

***Object Comparison***

1. No issues because there are just objects reference comparison (==null).

***Output*** ***Format***

1. No functions for printing output messages are present in this function.
2. No functions for printing output messages are present in this function.
3. No functions for printing output messages are present in this function.

***Computation,*** ***Comparisons*** ***and*** ***Assignments***

1. Brutish Programming is present in follow lines of code

if (securityManager != null) {

// Checking sealing

if (pkg != null) {

boolean sealCheck = true;

if (pkg.isSealed()) {

sealCheck = pkg.isSealed(entry.codeBase);

} else {

sealCheck = (entry.manifest == null)

|| !isPackageSealed(packageName, entry.manifest);

}

if (!sealCheck)

throw new SecurityException

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

+ packageName + " is sealed.");

}

}

1. No issues
2. Lines 2788-2789: Parenthesis are used to avoid operator precedence problems.
3. No issues
4. No issues
5. No issues
6. No issues
7. No issues

***Exceptions***

1. Line 2792: SecurityException is thrown and is not properly handled in the function, and it is neither stated in the function declaration.
2. No issues

***Flow*** ***of*** ***control***

1. No switch statements are present in this function.
2. No switch statements are present in this function.
3. No loops are present in this function.

***Files***

1. No files are used by the function.
2. No files are used by the function.
3. No files are used by the function.
4. No files are used by the function.

**findResourceInternalFromRepositories(String name, String path)**