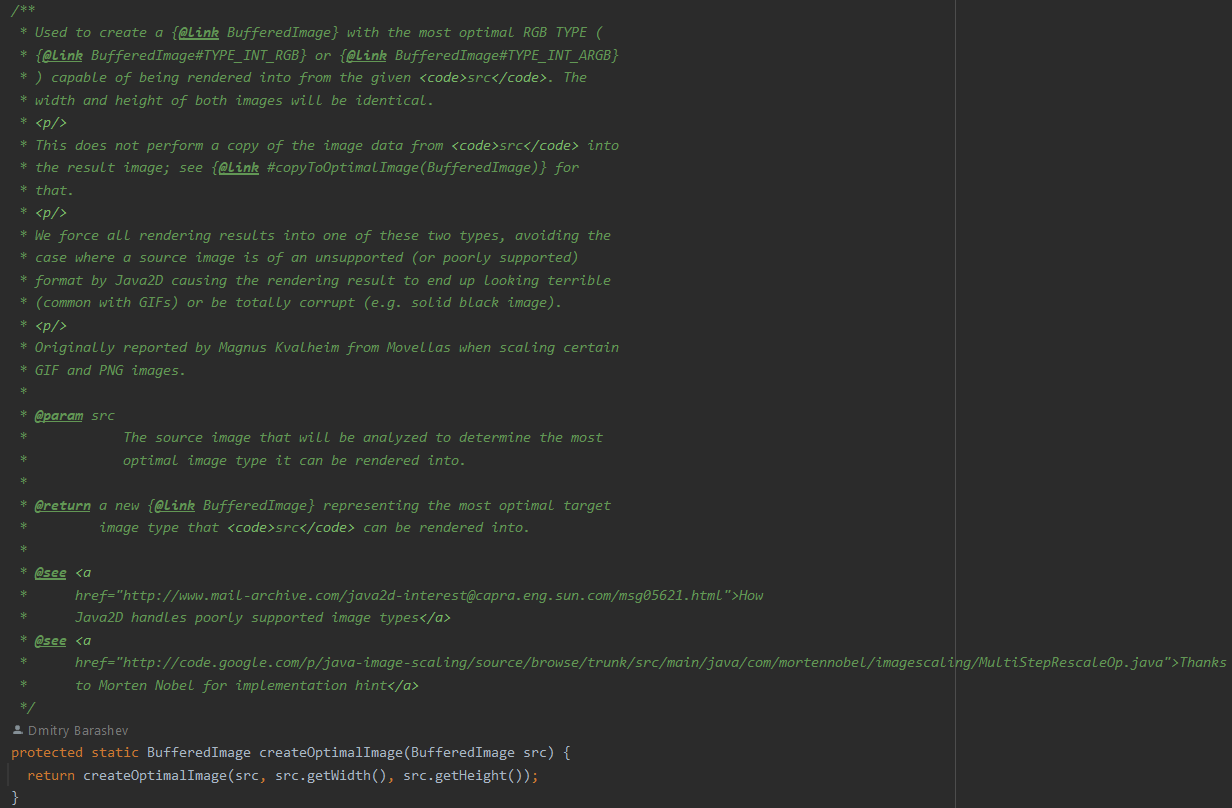
**Code smells**

**Comments**

**Location:** ganttproject/src/main/java/org/imgscalr

**Class:** Scalr.java



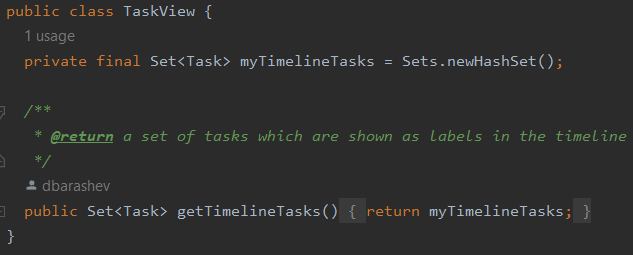
**Explanation:** In this class we can see too many comments above (and even inside) the methods and variables. This type of commenting can become very confusing to understand, and can be an indicative of bad or complicated design.

**Refactoring:** A solution could be reducing a bit the size of the comments of each method, making it more organized and easy to understand. The decision can be done through importance of comment, leaving the more important ones, such as succint descriptions of the parameters and methods (basic documentation), and explanations of algorithms.

**Data class**

**Location:** ganttproject/src/main/java/net/sourceforge/ganttproject/task

**Class:** TaskView.java



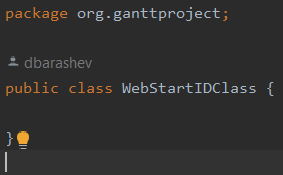
**Explanation:** This class is composed of an instance variable and its getter method. Hence, the class is too small and has no real funcionality, it only has data. This indicates that it’s not really necessary.

**Refactoring:** One possible solution could be removing the class and replacing it by a variable of the type Set<Task> where the object of this class is being called/used.

**Dead code**

**Location:** ganttproject/src/main/java/org/ganttproject

**Class:** WebStartIDClass.java



**Explanation:** This class is never being used in any part of the code, so it is unreachable. Besides that, it doesn’t contain any information at all, being even more useless and increasing the code complexity unnecessarily.

**Refactoring:** The class could be removed, as it is not being used, and the code would be cleaner and simpler, only containing useful implementations.