src\main\java\org\jabref\migrations

“Measuring programming progress by lines of code is like measuring aircraft building progress by weight.” -- Bill Gates

My task is to analyze the line code metrics for the folder “migrations”.

The purpose of using this type of metrics is:

1. **Check the size of code units.** The analysis of different units of the program allows us to see some of the shortcomings made in the process of writing the code. For example, a method longer than 20-30 lines can be quite difficult to understand and work with. It is also highly undesirable for classes that are too long to search for lines of code and read the code.
2. **Estimate the size of project.** We can estimate the number of logical lines of code and physical lines of code, compare them. However, applying this metric, we must understand that the number of lines of code is not equal to well-made code.

We must clearly understand that the length of the code is not an indicator of the productivity and efficiency of the programmer. It is not uncommon for 20 lines of code to be significantly worse than 100 lines of code in terms of time or space complexity.

**Method metrics**

Analyzing the methods of the folder in question, we can conclude that most of the lines of code consist of actual code, rather than comments. There is a appropriate amount of comments in places where it is necessary, namely before long and more complex methods.

**Class metrics**

Looking at the classes, we can also say that the number of lines of code exceeds the number of comments. No unnecessary comments, everything looks structured and easy to read. In general, we can conclude that the classes and interfaces of this folder are quite balanced.