# Section 3.1

In this report, we will be using EclEmma to report the coverage metrics. EclEmma only supports statement and branch coverage, but does not support condition coverage.

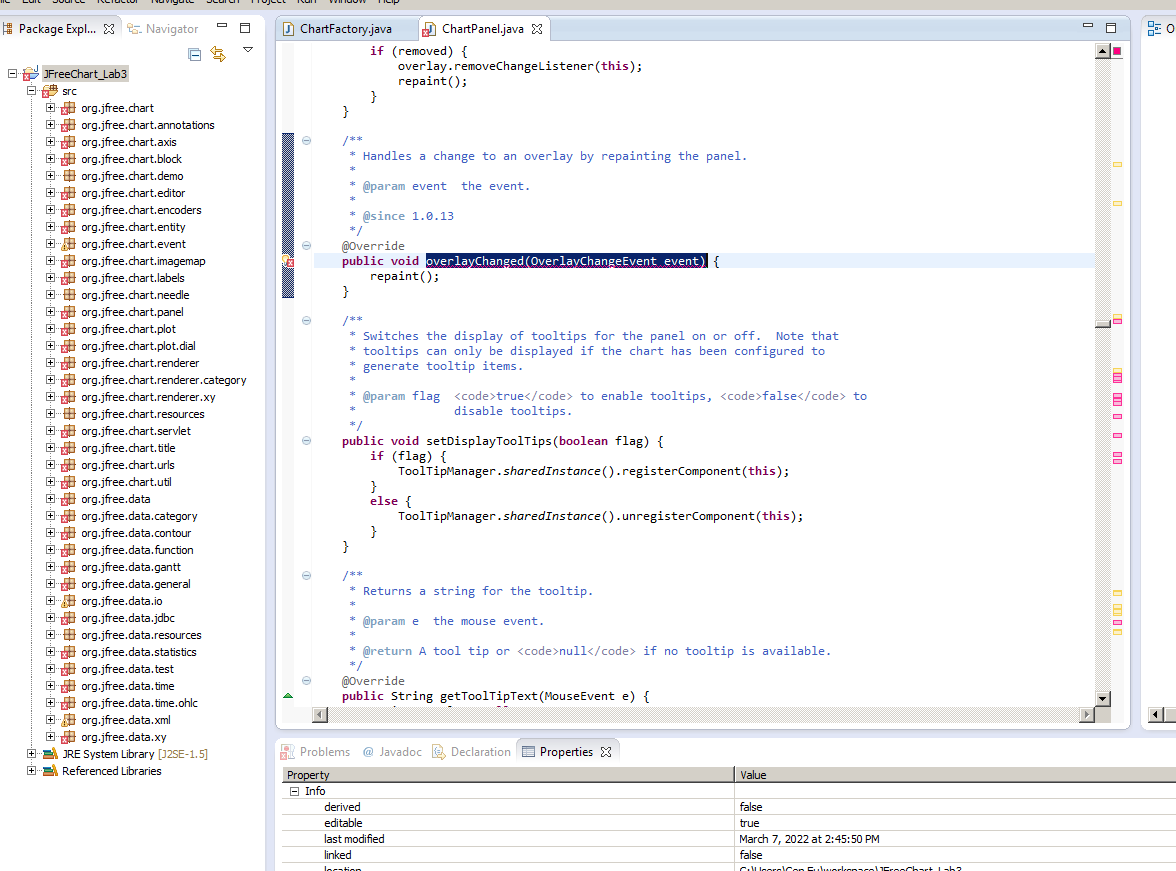
Other tools have been experimented but we cannot get the condition coverage to work. For example, CodeCover do have condition coverage, but it seems that it is discontinued from support.

When we run CodeCover on the current version of eclipse, it will produce the following error:

Plug-in "org.codecover.eclipse" was unable to instantiate class "org.codecover.eclipse.junit.JUnitLaunchConfigurationDelegate".org/eclipse/osgi/framework/internal/core/BundleHost

Upon researching this issue on stackoverflow, it seems that only Eclipse Kepler can be used with CodeCover.

We have also attempted to install Eclipse Kepler on another system to test CodeCover, but unfortunately we have multiple issues running the jfreechart code in that version of Eclipse. (see screenshot below):



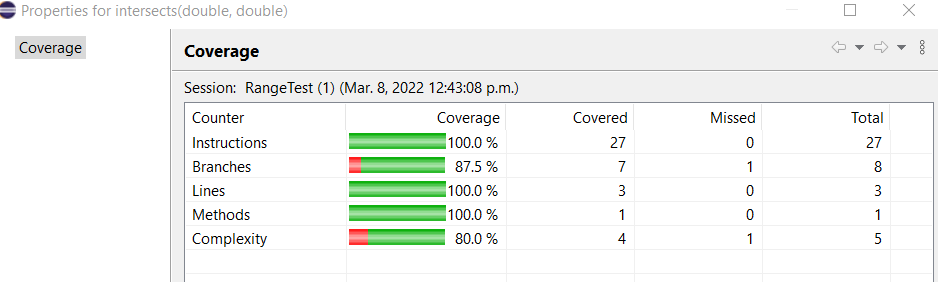
JaCoco has basically the same featureset as EclEmma, there are no differences and thus EclEmma is used.

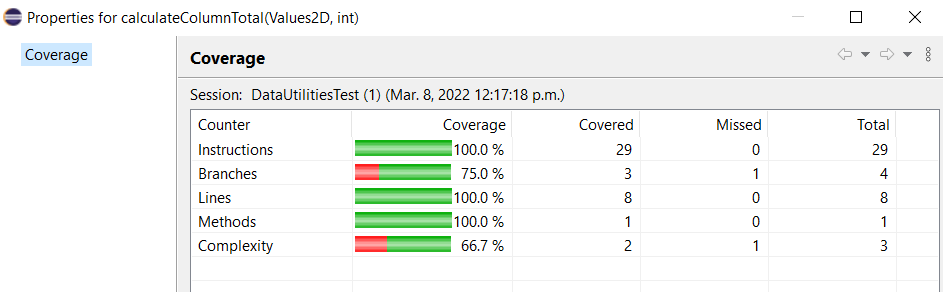
As for Clover, we have not been able to obtain the 30-day evaluation key successfully from my.atlassian.com. However, given that it is a paid tool, we believe that EclEmma is still a better solution as it is more user-friendly to install and does not require a key while providing similar functionality.

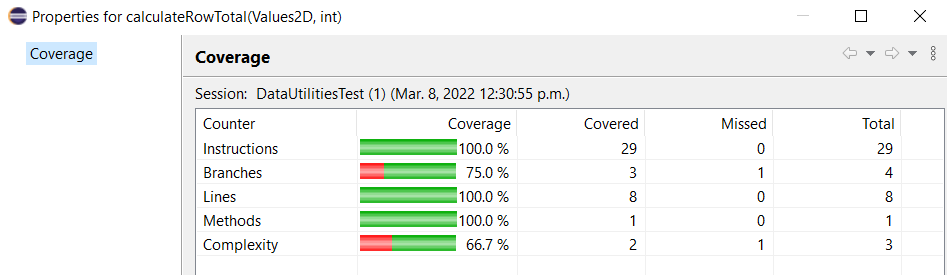
Coverlipse (<http://coverlipse.sourceforge.net/>) and Cobertura (<http://cobertura.github.io/cobertura/>) had not been tested.

Thus, going forward, the tests made in this report will be using EclEmma.

For coverage metrics:



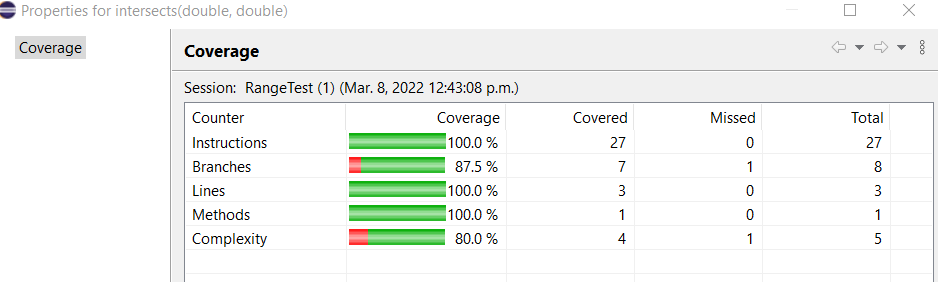




# Section 3.3

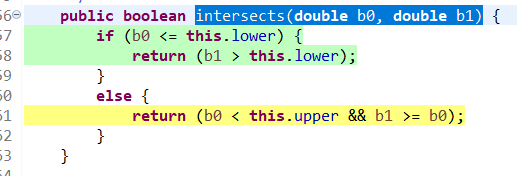
## Improving Coverage for intersects

Using EclEmma, the coverage for instructions, branches, lines, and methods is:



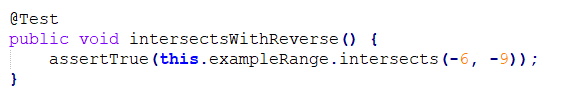
Although we have reached the minimum coverage of 70% coverage for branch, we can still find ways to improve this coverage.

When we analyze our code using branch coverage:

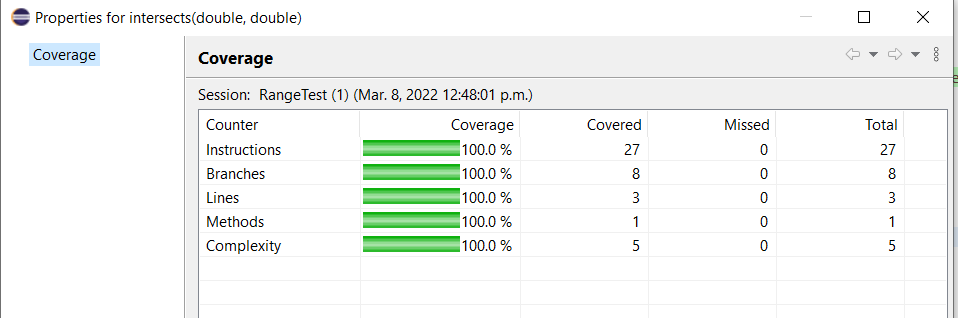


We have found that the branch that we did not cover is when b0<b1.

To improve on the coverage, we have added the following test case to add a null value.

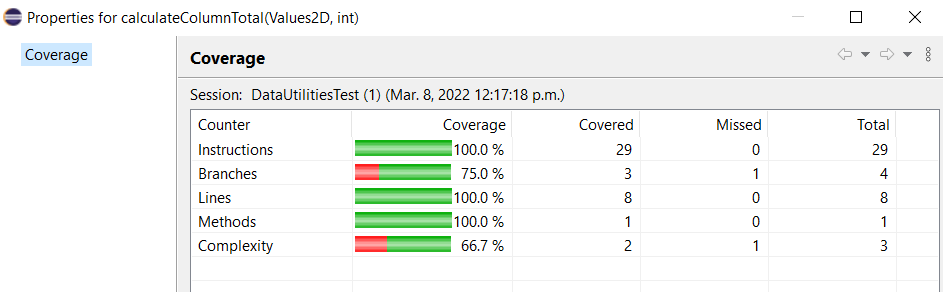


The result is that the coverage for branch is also 100%:



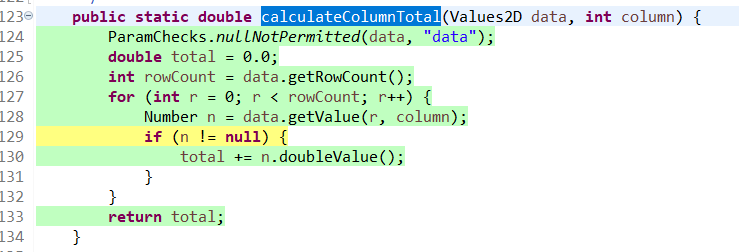
## Improving Coverage for calculateColumnTotal(Values2D, int)

Using EclEmma, the coverage for instructions, branches, lines, and methods is:



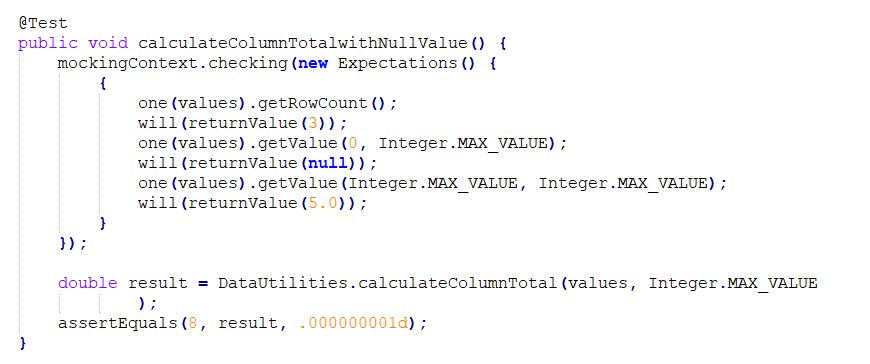
Although we have reached the minimum coverage of 70% coverage for branch, we can still find ways to improve this coverage.

When we analyze our code using branch coverage:

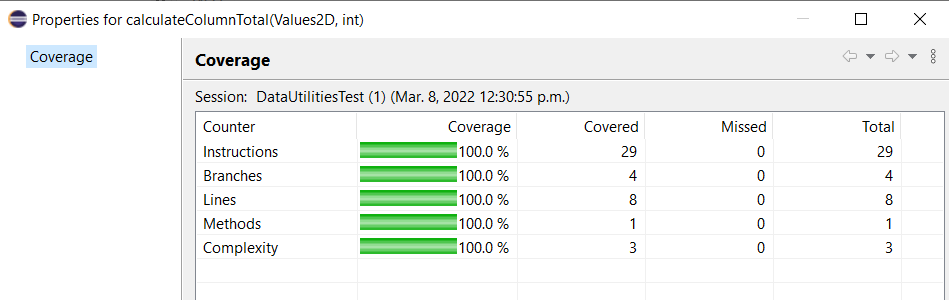


We have found that the branch that we did not cover is when the variable n is equal to null.

To improve on the coverage, we have added the following test case to add a null value.

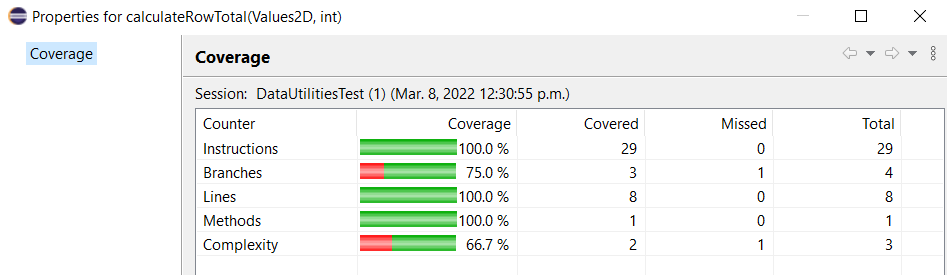


The result is that the coverage for branch is also 100%:



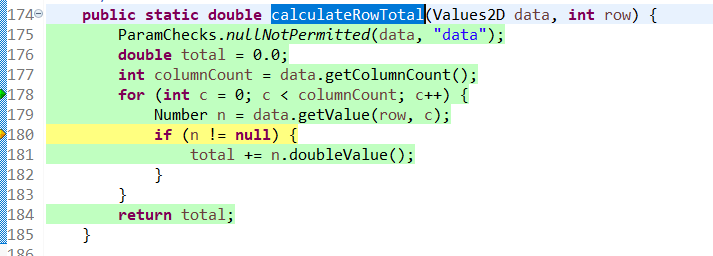
## Improving Coverage for calculateRowTotal(Values2D, int)

Using EclEmma, the coverage for instructions, branches, lines, and methods is:



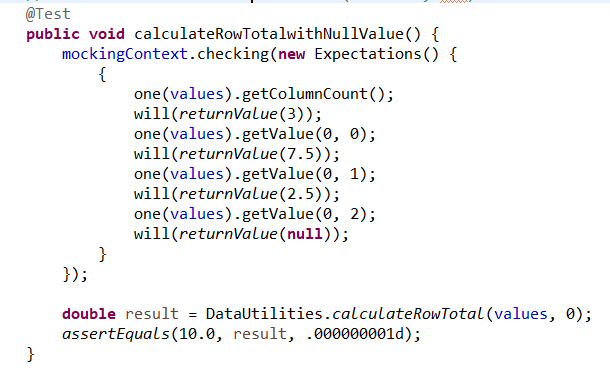
Although we have reached the minimum coverage of 70% coverage for branch, we can still find ways to improve this coverage.

When we analyze our code using branch coverage:



We have found that the branch that we did not cover is when the variable n is equal to null.

To improve on the coverage, we have added the following test case to add a null value.



The result is that the coverage for branch is also 100%:

