**Dependency metrics**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | Cyclic | Dcy | Dcy\* | Dpt | Dpt\* | PDcy | PDpt |
| **org.jabref.JabRefPreferencesTest** | 0 | 2 | 1 334 | 0 | 0 | 1 | 0 |
| **org.jabref.TestIconsProperties** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **org.jabref.architecture.MainArchitectureTests** | 0 | 3 | 3 | 1 | 1 | 1 | 1 |

**Legend:**

Cyclic – number of cyclic dependencies

Dcy – number of dependencies

Dcy\* – number of transitive dependencies

Dpt – number of dependents

Dpt\* – number of transitive dependents

pDcy – number of package dependencies

pDpt – number of dependent packages

Analysis of the collected metrics:

In this data file, we have the example of three different classes with very different values regarding the code’s dependencies. As we can see, none of them has cyclic dependencies.

The first class, although it doesn’t have a lot of dependencies (2), it has 1334 transitive dependencies. A transitive dependency is any dependency that is induced by the components that the program references directly. It has no dependents or transitive dependents. This class has one package dependency.

The second class doesn’t have any dependencies or dependents.

The third one has 3 dependencies and 3 transitive dependencies, has 1 dependent and 1 transitive dependent. This class also has 1 package dependency and 1 dependent package.

The amount of transitive dependencies in this class could be a trouble spot in the code because a class shouldn’t depend so much on others.

The Dependency metrics is associated with inappropriate intimacy, which is a code smell that occurs when two classes depend too much on one another.