Use Plain JUnit5 for advanced test assertions

Context and Problem Statement

How to write readable test assertions? How to write readable test assertions for advanced tests?

Considered Options

- Plain JUnit5
- Hamcrest
- AssertJ

Decision Outcome

Chosen option: “Plain JUnit5”, because comes out best (see below).

Positive Consequences

- Tests are more readable
- More easy to write tests
- More readable assertions

Negative Consequences

- More complicated testing leads to more complicated assertions

Pros and Cons of the Options

Plain JUnit5

Homepage: https://junit.org/junit5/docs/current/user-guide/ JabRef testing guidelines: <../testing.md>

Example:

```
String actual = markdownFormatter.format(source);
assertTrue(actual.contains("Markup<br />"));
```

```
assertTrue(actual.contains("<li>list item one</li>"));
assertTrue(actual.contains("<li>list item 2</li>"));
assertTrue(actual.contains("> rest"));
assertFalse(actual.contains("\n"));
```

- Good, because Junit5 is “common Java knowledge”
- Bad, because complex assertions tend to get hard to read
- Bad, because no fluent API

Hamcrest

Homepage: https://github.com/hamcrest/JavaHamcrest

- Good, because offers advanced matchers (such as `contains`)
- Bad, because not full fluent API
- Bad, because entry barrier is increased

AssertJ

Homepage: https://joel-costigliola.github.io/assertj/

Example:

```
assertThat(markdownFormatter.format(source))
        .contains("Markup<br />")
        .contains("<li>list item one</li>")
        .contains("<li>list item 2</li>")
        .contains("> rest")
        .doesNotContain("\n");
```

- Good, because offers fluent assertions
- Good, because allows partial string testing to focus on important parts
- Good, because assertions are more readable
- Bad, because not commonly used
- Bad, because newcomers have to learn an additional language to express test cases
- Bad, because entry barrier is increased
- Bad, because expressions of test cases vary from unit test to unit test

Links

- German comparison between Hamcrest and AssertJ: https://www.sigs-datacom.de/uploads/tx_dmjournals/philipp_JS_06_15_gRfN.pdf