

Responsibility driven design, sprint 3.

Because we are mainly testing for this sprint, we can't really do responsibility driven design. Instead, we will describe the hard choices we've made during testing and why we didn't fix some checkstyle warnings.

First the checkstyle warnings. We didn't fix some of the warnings which told us to make all methods final which the tool thought should be final. We didn't do this in all cases, because during testing you can't mock final methods and some of the methods didn't really have to be final, as it would not be completely illogical to override them in a subclass. We didn't want to use powermock to mock final methods, because it works pretty badly together with javafx (getting invalid stack frame exceptions) and travis often failed to build powermocked test classes while they succeeded when building maven on our own pcs.

During testing we only had difficulties with mocking, because we had to mock a lot of the dependencies. This wasn't a big problem though.

The integration tests were a lot harder to make however. You can't fire button events from a different thread than from the javafx thread, so we had to make a junit runner which launched the application and then ran the tests while keeping the application running. We used a big part of this code from someone else (sources are specified in the comments), but we had to do the fine-tuning for our own applications ourselves.