

Feedback from the developers of jOcular

We emailed to the authors of each of the applications sending them the details of two sample opportunities (the ones with highest number of cond-Stmts). For each of those opportunities, we asked the authors the following two *key* questions:

(KQ_A) “Is the refactoring opportunity intuitive?”

(KQ_B) “If the answer to KQ_A is ‘Yes’, then will you be open to performing the refactoring?”

We now discuss the feedback from the developers of jOcular.

Feedback from the developers of jOcular. Fig. 1 shows the details of the two refactoring opportunities that were mailed to the developers of jOcular. Each refactoring opportunity is represented by a control-field. In the table, number of cond-Stmts is shown as #uses, number of subclasses that will be created after refactoring is shown as #v, and RO represents the type of the refactoring opportunity (SC or ST). In the first opportunity (*O1*), the `log` field of the class `GraphAxis` is used to identify if we are using a *log*-graph axis or not, and accordingly different computations are carried out. Suggested refactorings: two subclasses of `GraphAxis`, say: `LogGraphAxis` and `NonLogGraphAxis`. Summary of the author response: 1) *Yes.* 2) *I will probably do this. I have added a ticket: <https://sourceforge.net/p/jocular/tickets/139>.* In the second opportunity (*O2*), the `m_value` field of class `BooleanProperty` tells if the optical object absorbs the photons (and hence suppressed). In this case, the cond-Stmts are not present in the class `BooleanProperty`, but outside (in the classes of the optical objects). After refactoring, the optical objects can be in two states: suppressed or non-suppressed. Summary of the author response: 1) *You are right, it does not seem right to check in each method whether the object is suppressed.* 2) *I will not do this, as this an obsolete piece of code.* The feedback from developers illustrates that the opportunities identified by Auto-SCST are intuitive. However, the decision to apply refactoring depends on many other factors (such as, obsolete code) which can be specific to each individual application.

Sl.	Refactoring Opportunity	RO	#uses	#v
<i>O1</i>	<i>⟨GraphAxislog⟩</i>	SC	3	2
<i>O2</i>	<i>⟨BooleanProperty m_value⟩</i>	SC	8	2

Figure 1: Two of the refactoring opportunities in jOcular.