

Allow org.jabref.model to access org.jabref.logic

Context and Problem Statement

- How to create a maintainable architecture?
- How to split model, logic, and UI

Decision Drivers

- Newcomers should find the architecture “split” natural
- The architecture should be a help (and not a burden)

Considered Options

- `org.jabref.model` uses `org.jabref.model` (and external libraries) only
- `org.jabref.model` may use `org.jabref.logic` in defined cases
- `org.jabref.model` and `org.jabref.logic` may access each other freely

Decision Outcome

Chosen option: “`org.jabref.model` may use `org.jabref.logic` in defined cases”, because comes out best (see below).

Pros and Cons of the Options

`org.jabref.model` uses `org.jabref.model` (and external libraries) only

The model package does not access logic or other packages of JabRef. Access to classes of external libraries is allowed. The logic package may use the model package.

- Good, because clear separation of model and logic
- Bad, because this leads to an [Anemic Domain Model](#)

`org.jabref.model` may use `org.jabref.logic` in defined cases

- Good, because model and logic are still separated
- Neutral, because each exception has to be discussed and agreed

- Bad, because newcomers have to be informed that there are certain (agreed) exceptions for model to access logic

`org.jabref.model` and `org.jabref.logic` may access each other freely

- Bad, because may lead to spaghetti code
 - Bad, because coupling between model and logic is increased
 - Bad, because cohesion inside model is decreased
-