The folder MySQL (Stored Procedures) is comprised of 3 sub-folders:

* CLASSES & MESSAGES: referring to those stored procedure queries used for Chapter 3, to retrieve the bulk (unfiltered) data of the most problematic runtime exception *classes*, and then analogously within those identified most problematic 'runtime exception classes' errors, the most prevailing 'exception errors *messages*' categories[[1]](#footnote-1) that encompassed the largest percentage share of constructs' errors.
* TOTAL & USERS INSTANCES: referring to those stored procedure queries used for Chapter 3, to retrieve the ArrayIndexOutOfBoundsException, StringIndexOutOfBoundsException and IndexOutOfBoundsException classes '*Total Number*' of Instances and '*Users Number*' values; for those queries (unlike the cases of the 'CLASSES & MESSAGES' queries above), a different time-period (termed as 'User Lifetime') was used[[2]](#footnote-2), and more specifically ‒ the time-difference between the time that each individual participant subject user had opted-in to participate in Blackbox, until the time that this participant subject user’s exception error occurred (was thrown). In addition, these queries (unlike the cases of the 'CLASSES & MESSAGES' queries above) were further filtered to include: i) those Blackbox participant subject users whose lifetime-interaction with the BlueJ IDE had more than 1 session, and ii) those errors which derived from standard Java libraries, and from exception-message type categories that were previously indicated to have the highest number of instances (under appropriate thresholds).
* CONTENT: referring to those stored procedure queries used for Chapter 5, to retrieve the sample of those participants' actual error-cases, whose packages' source-files upon loading in BlueJ contained the complete source-code text *content* which led to the ArrayIndexOutOfBoundsException errors; for those queries (like in the case of the 'TOTAL & USERS INSTANCES' queries above), the same 'User Lifetime' time period had been used, as additionally the same filtering pattern (i.e., instances of Blackbox participant subject users whose lifetime-interaction with the BlueJ IDE had more than 1 session, and whose instances derived from exception-message type categories having the highest number of errors with appropriate thresholds, stemming only from standard Java libraries). However, one thing to note in these queries is the fact that, unlike the 'CLASSES & MESSAGES' and the 'TOTAL & USERS INSTANCES' folders, each query in this folder contains an extra step; and that is, that the researcher upon running these queries will receive a source\_file id and a master\_event id ‒ which should be subsequently fed to an external Blackbox tool (/tools/nccb/bin/print-source-state), as with the example below:

source\_fileid master\_eventid

/tools/nccb/bin/print-source-state297887841117615934

(first one being the source\_file id number, and second the master\_event id number) in order to retrieve the complete source code output for the class at the point where the participant subject user received the error. Finally, irrespectively of the sp\_ArrayIndexOutOfBoundsException\_contents\_more\_one\_sessions.sql file, queries for the NullPointerException, StringIndexOutOfBoundsException and IndexOutOfBoundsException classes have been additionally provided, for those researchers interested in further exploiting the actual source-file contents, leading Blackbox’s participant subject users' to succumb under those types of errors.

1. The term exception message refers to the short descriptive message text following the name of the exception, contained in the first line of information displayed in response to the invalid input (stack-trace) that led to the exception. The idea for this subsequent data analysis over the exception message types, stemmed from the logic that the procurement of these supplementary elaborated data, would provide *more information about the reasons* why the most problematic 'runtime exception-classes' augmented error instances had occurred; and consequently, similarly why the associated numerous constructs' errors (encompassed within those exceptions) had been also generated. [↑](#footnote-ref-1)
2. Not simply the time period when the exception error had occurred (was thrown). [↑](#footnote-ref-2)