

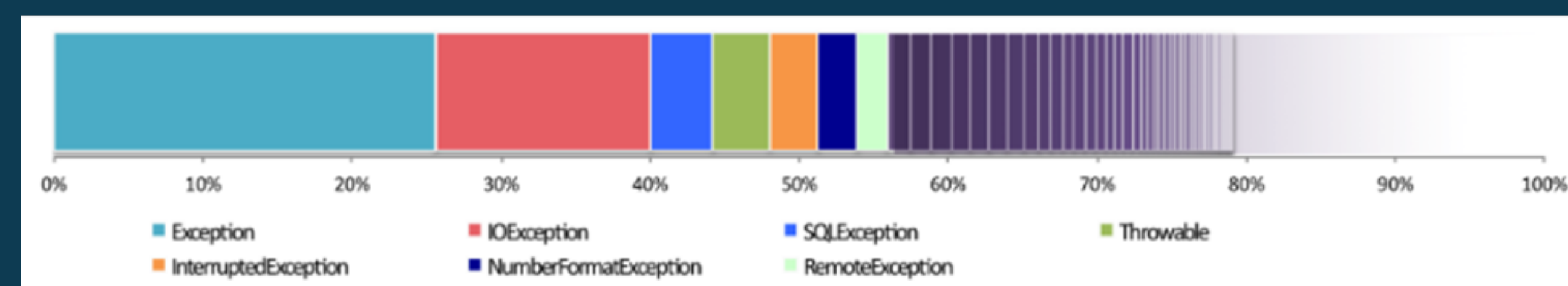
Moonstone

a tool to support exception handling code comprehension for developers

Mary Beth Kery
Steven Moore
Michael Puskas
Brad A. Myers

The Problem

Despite the importance of designing software that behaves well in error conditions, there is abundant evidence suggesting that error handling code is often poor quality. One study of Java repositories found that code using the exception mechanism to handle errors is three times as likely to contain faults when compared to other kinds of code. [1]



Exceptions caught by Java catch blocks on Github. Exceptions that occur more than 1% of the time are labeled. The rest, in purple, are thousands of exceptions that are rarely ever caught.

For example, a large-scale analysis of the exception handling in nearly 8,000,000 Java repositories on Github revealed that the general "Exception" class is caught 26% of the time. [2] In most cases, this is a poor practice that can carry unexpected consequences. Other bad practices include ignoring exceptions, not preserving original exceptions, and failing to close resources in exceptional situations. [3][4]

Our Solution

In order to help developers become more proficient at writing proper exception handlers, we have developed Moonstone: an Eclipse plugin that aims to teach novice developers how to write better exception handlers, and also provide expert developers with faster options for handling exceptions. Based on a survey of past literature and an interview study that we previously conducted on five developers, we have chosen four main features for Moonstone.

References

1. Sawadpong, Puntitra, Edward B. Allen, and Byron J. Williams. "Exception handling defects: An empirical study." In High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on, pp. 90-97. IEEE, 2012.
2. Kery, Mary Beth, Claire Le Goues, and Brad A. Myers. "Examining programmer practices for locally handling exceptions." Proceedings of the 13th International Workshop on Mining Software Repositories. ACM, 2016.
3. Asaduzzaman, Muhammad, Muhammad Ahasanuzzaman, Chanchal K. Roy, and Kevin A. Schneider. "How developers use exception handling in Java?." In Proceedings of the 13th International Workshop on Mining Software Repositories, pp. 516-519. ACM, 2016.
4. Weimer, Westley, and George C. Necula. "Mining temporal specifications for error detection." International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer Berlin Heidelberg, 2005.

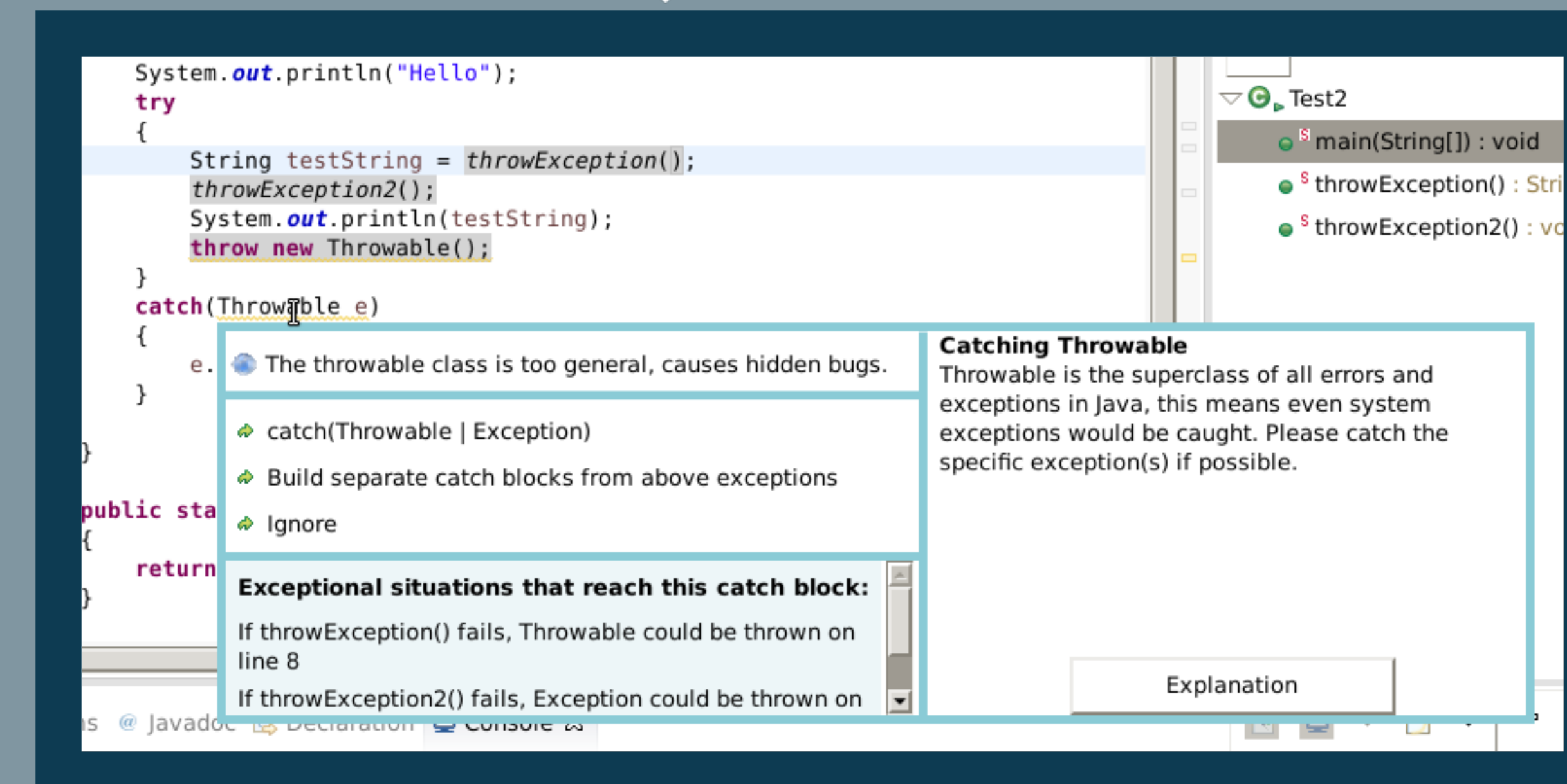
Acknowledgements

We would like to acknowledge Joshua Sunshine, Claire Le Goues, and Charlie Garrod for their coordination of the excellent REUSE program at CMU.

Funded by NSF Grant # CNS-1423054

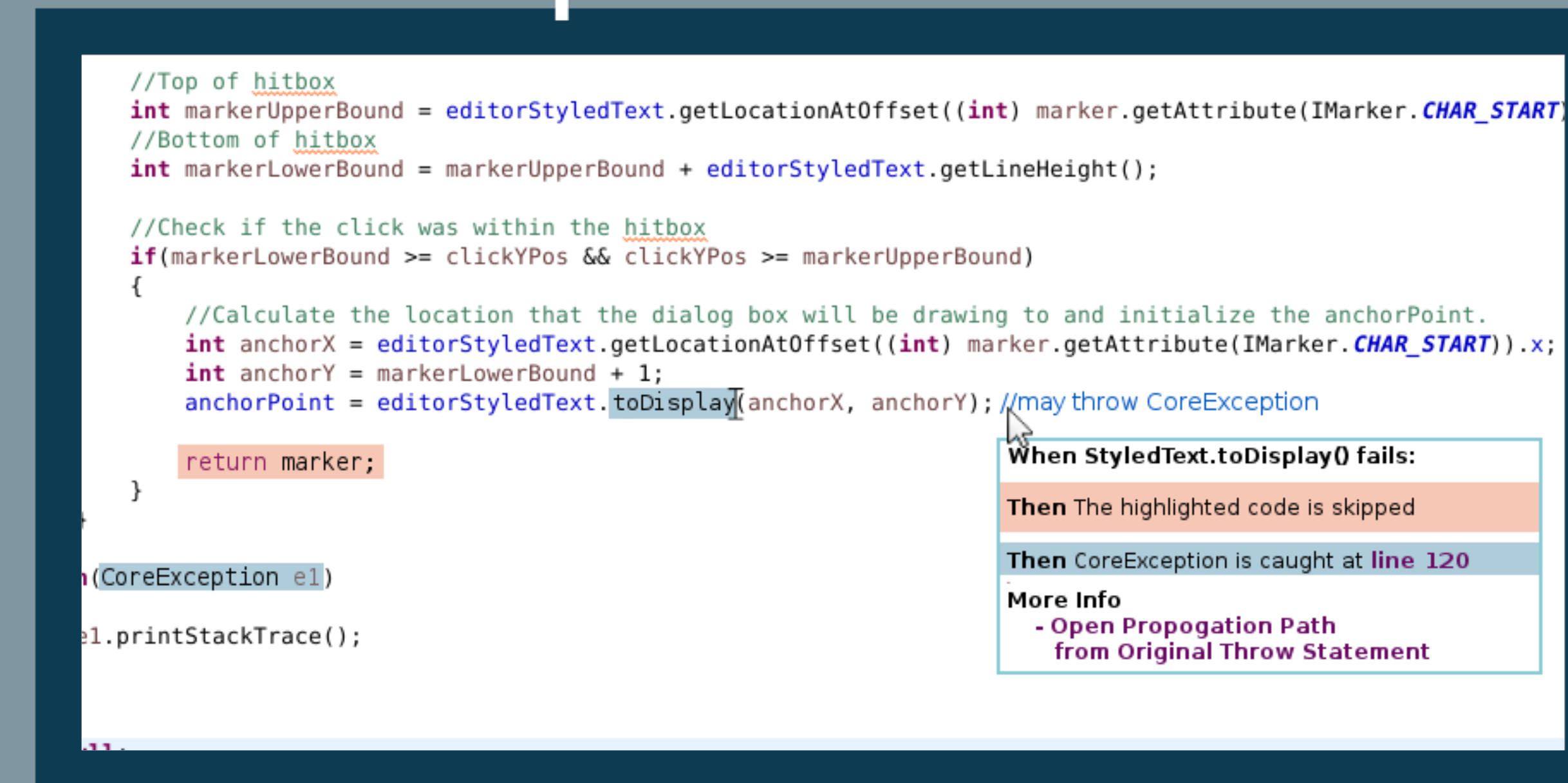
Features

Bad Practice Quick Fixes



Moonstone detects bad practices in exception handling code and presents the developer with a warning. Upon hovering over the relevant code, a dialog appears which offers potentially better ways of handling the situation. The developer can click on any of the solutions to automatically implement them.

In-line Exception Information

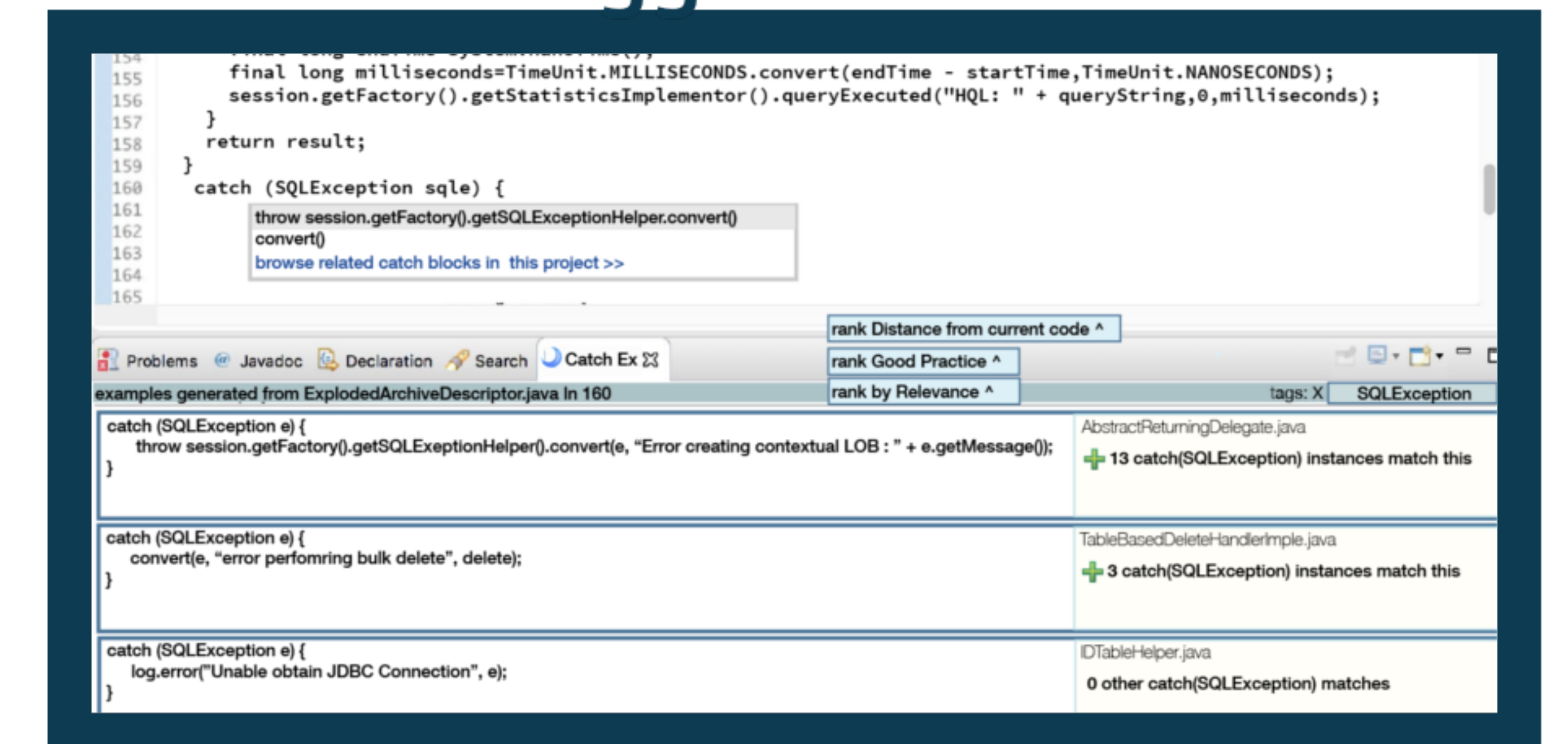


While reading code, the particular line that throws an exception is often hard to locate. This forces developers to check the documentation of each method within a try statement for its exceptional behaviour. Moonstone surfaces this exceptional behaviour in the form of in-line comments that appear while the text cursor is within an exception handler.

Ongoing Work

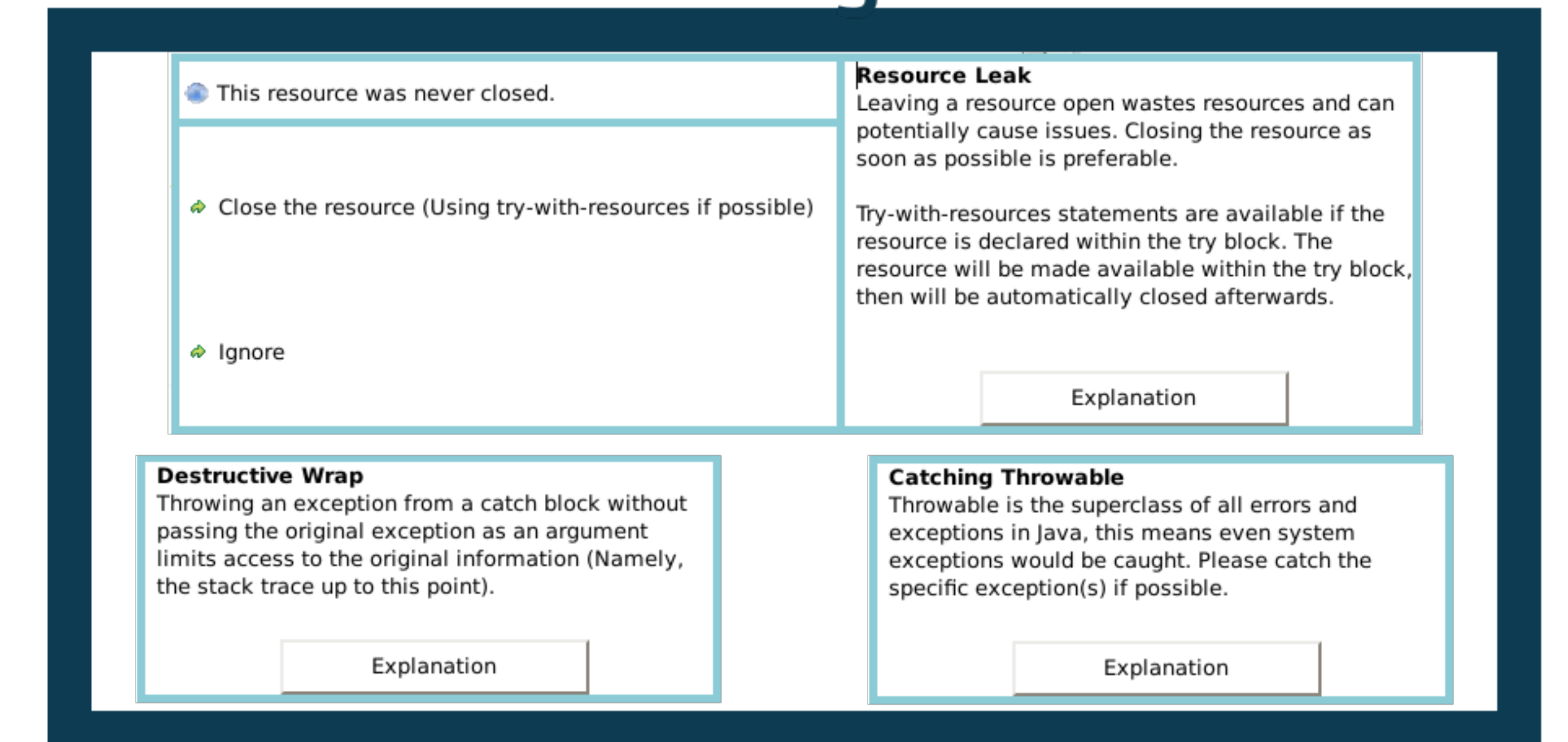
The evaluation of Moonstone will be the next step in our work. First, we will deploy a survey to test at a larger scale our findings from a previous interview study. This survey will be targeted at developers familiar with any language that contains exception handling constructs. Then, we will conduct user studies to test multiple facets of the effectiveness of the tool.

Past Code Suggestion



Professional coding projects often have established conventions for how to handle exceptions. When writing an exception handler, Moonstone offers examples of relevant past code from within the project. This code is ranked based on similarities between the past and present situations.

Accessible Learning Materials



Moonstone presents developers with easily accessible, concise explanations of why a section of code is considered a bad practice. If the developer wishes to learn more about the bad practice, they are able to press a button to open a more detailed explanation in their browser.

Survey

Developers will be asked a broad range of questions focused around exception handling code and the issues that they perceive in writing it. They will also be asked how they typically handle various exceptional situations.

User Study

The main focuses of the user study will be in determining the usability of the tool, and the extent to which Moonstone is successful in teaching developers proper exception handling techniques.