Gisany Ravichandran

Java Exception Handling Analyzer
Tasks:

- Read the main README.md to get familiar with features of the analyzer
- Install the extension following the instruction in vscode-extension/README.md
- Open the provided Java files from resources/sample-java
- Run analyzer by opening VSCode command palette and run command: Analyze Java
- Config risk levels and colors by opening VSCode command palette and run command: Configure Java Analyzer
- Try to fix issues in ExampleWithIssues.java with the analyzer and check ExampleWithIssuesSolution.java

Questions:

1. **How easy do you feel the installation process?**

   Yea it was nice, the installation of the VSCode extension was pretty straightforward. No issues there.

2. **Are the introduction and instructions clear enough? What might be missing?**

   The instructions were mostly clear. I think the really short example or walkthrough you gave me (especially for people who aren't that familiar with Java) was helpful just to bridge some knowledge gap between seeing an error and knowing how to approach fixing it.

3. **Are the error messages and suggestions helpful when issues are found?**

   They're helpful overall especially because it highlights exactly where the problem is. But some of the messages were a bit confusing like one warning said something about "number of functions that might encounter the error," and wasn't totally sure what that meant. The only reason i figured it out was because the function name in the examples helped me figure it out, literally called "alwaysTriggered()". This is probably just my lack of familiarity with java though.

4. **Are the highlighted issues easy to understand? How do you feel about the user-customized risk level feature?**

   The highlights are good but at first, when they were red it made me think the code wouldn't compile but it was just a flag. So being able to customize the colors is really nice for my preference. The risk levels are cool too especially if you want to focus on certain kinds of issues

5. **Is there any other feature you hope to include in the analyzer?**

   Yeah, it would be really helpful if the extension could offer quick fixes or some kind of example. Like, when I hover over a warning, it could suggest and apply the fix automatically or give me an example of how to fix it. Because like sometimes when I try to fix one thing, I accidentally cause another warning, so having a "one-click" fix or example would make the experience smoother. But again, probably my lack of familiarity with java.

6. **Overall, how do you feel about the analyzing tool? Is it helpful? Would you use it in your Java programming projects?**

   I think it's pretty cool. I didn't even know some of these exception-handling issues were a thing, so it definitely taught me something. I'd probably use it if I was writing more Java especially during debugging or some kind of code cleanup