Idea 1: race condition analyzer

Use case: Analyze asynchronous functions and highlight possible race conditions

Idea 2: Exception Handling Analyzer for Java

Use-Case: Identify poorly handled exceptions in Java codebases, increasing the code performance.
Static analysis:

- Examples:
    - try-catch statements with empty catch(Exception e) {}, leading to silent failures.
    - Redundant throws when the code doesn't throw an specific exception (e.g. IOException)
    - Only handle the exception by logging (System.out.println()), without stack trace (e.printStackTrace()) or recovery action
    - Preventing expensive exception-based control flow, check if try-catch blocks in loops or any in-loop methods.
    - Redundant try-catch blocks (e.g. we can use Files.exists() to check file existence rather than catching FileNotFoundException)
    - Unclosed resources in catch blocks:
      FileInputStream fis = new FileInputStream(//);
      try {//}
      catch (Exception e) {//}
      finally {//} // No code to close the FileInputStream
    - And more…

Visualization: Assign risk score to catch block and throw exception statement, high-risk blocks are highlighted in red, and low or medium-risk blocks are highlighted in yellow.

Idea 3: Runtime Analysis Visualizer for Performance Optimization

Use-Case: We can time how long the program spends on various function execution calls, provide different inputs, and visualize the results. This can help the developer prioritize what areas to optimize, if they are trying to make their code more performant.

Dynamic Analysis:
The user would specify a list of different inputs to run on a target program. They might also provide the function names, or line numbers, of the sections they want to time. We run the program using all the different inputs, timing how long the program spends in each section.

Visualization:

We will visualize the results in a (bar) graph, or a call stack. We can show the output for each set of inputs, or average results across all inputs.

Follow-up tasks:

- Brainstorming more ideas within our team
- Confirm our ideas with the TA
- Writing proposal by next Friday