Method Trace Analyser

The challenge is aimed at developing an application to assist developers in debugging code. Method tracing is one of the commonly used post-mortem diagnostic methods to identify problems. It consists of timestamp of entry and exit points for each method invocation. They may also contain stack-trace for each invocation. Depending upon the time for which trace data is collected, the file-size for these traces can be huge. Parsing them manually is a timeconsuming and error-prone task.

Instructions:

Methodtrace.java

Run

Graph.java

Log2.java file

Log3.java

This Run a Factorial Code using recursion with smaller input

Minimize the window (Don’t close it)

This shows a file choosing window

This shows the graphical representation of code- includes “execution time, performance, error, exception, line of code”

Enter the smaller input in popup window and click “OK”

Select the “execptions.log” trace file from browse menu:-C:\Users\Lenovo\eclipse\java-photon-2\eclipse\ibm\src\ibm\execptions.log

This show the contents of trace file in text field area

Graph.java

Log2.java file

Log3.java

This shows a file choosing window

This shows the graphical representation of code- includes “execution time, performance, error, exception, line of code”

Enter the input in popup window and click “OK”

This Run a Factorial Code using recursion with input creating exception

This show the contents of trace file in text field area

Select the “execptions.log” trace file from browse menu:-C:\Users\Lenovo\eclipse\java-photon-2\eclipse\ibm\src\ibm\execptions.log

Graph.java

Main2.java file

Log4.java

This Run a Factorial Code using for loop

This shows a file choosing window

Select the “execptions.log” trace file from browse menu

This shows the graphical representation of code- includes “execution time, performance, error, exception, line of code”

This show the contents of trace file in text field area

**Prerequisite:**

* Eclipse Photon
* Libraries: Java AWT , Java swing, org.apache.log4j, jcommon, jfreechart
* You can add a jar file in eclipse by right clicking on the project 🡪 Build Path 🡪 configure build path🡪under libraries tab 🡪 click add jar or add external jar and give the jar.
* execptions.log file at C:\Users\Lenovo\eclipse\java-photon-2\eclipse\ibm\src\ibm\execptions.log

Initially execptions.log fill will be empty

Running the test:-

Step 1):

Main Code:--Methodtrace.java, This file runs a GUI application in which there are four buttons one of which is “Open Trace File” button and a text area which displays the contents of trace file.

Step 2):Button:-- “Run Code 1”, on clicking this button log2.java file runs in the background which accesses two more files graph.java (For graphical Representation of code) andlog3.java (The input code using recursion). We give smaller value as input and it will not have an exception in it (PASS CASE).

Step 3):

Button:-- “Rerun code 1 ”, on clicking this button log2.java file runs in the background which accesses two more files graph.java (For graphical Representation of code) and log3.java (The input code using recursion) but this time we give an input which creates an exception (FAILURE CASE) which is shown in our graphical representation of code. For eg:-larger input or a mismatched data type to check the exception/error case.

Step4):

Button:-- “Run Code 2”, on clicking this button main2.java file runs in background accessing two more files log4.java (Input code using for loop) and graph.java (graphical representation).this contains the same code with different semantics.

After the graphical representation displays on clicking button :

🡪minimize the graph window🡪click on the “open trace file ” button which popup a file choosing window 🡪choose the “execptions.log” file from C:\Users\Lenovo\eclipse\java-photon-2\eclipse\ibm\src\ibm\execptions.log

* This displays the content of each execution of code(IT WILL APPEND THE TRACES OF NEXT EXECUTION IN THE SAME TRACE FILE[SCROLL TILL END])

Graphical representation:

The Graph thus generated will be the proof of the analysed code and will show us

1. Performance of the code
2. Line of Code
3. Execution Time
4. Error or not
5. Exception or not

Trace file:

While execution of code, trace file will be generated by itself showing:- Date, Time, line number, class/method name, the output at each input in code(eg:- in loop etc), and the exceptions /errors occur if any, total execution time.

Build With:-

Eclipse Photon

**Code of Conduct**

**Our Pledge**

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

**Our Standards**

Contributes to creating a positive environment include:

* Using welcoming and inclusive language
* Being respectful of differing viewpoints and experiences
* Gracefully accepting constructive criticism
* Focusing on what is best for the community
* Showing empathy towards other community members

Unacceptable behavior by participants include:

* The use of sexualized language or imagery and unwelcome sexual attention or advances
* Trolling, insulting/derogatory comments, and personal or political attacks
* Public or private harassment
* Publishing others' private information, such as a physical or electronic address, without explicit permission
* Other conduct which could reasonably be considered inappropriate in a professional setting

**Our Responsibilities**

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

Versioning:-

Eclipse Photon: https://www.eclipse.org/downloads/packages/release/photon

Eclipse Mars: http://www.eclipse.org/downloads/packages/release/mars/2