

Method Execution Report

Date: 13/04/17

Type: FN / NF

System: draw

Session overall:

This session aims at gathering impression of the *Method Execution Report* from practitioners. The session is relatively short, only a few minutes long. The session is divided into three phases. The first phase is about yourself. The second phase present Method Execution Report, and the third phase is about the evaluation of the report.
Note that all your answers are treated anonymously.

Phase 1: About yourself

- Are you Female or Male ? F
- How many years of experience do you have in programming? ~44
- How long have you been programming in Java for? ~3y
- Which Java programming environments (IDE) are you familiar with? ~~E~~ Eclipse, IntelliJ
- Which other programming languages and programming environments do you use?
Python, C, Erlang, Javascript
- While programming, if your application does not behave as you expect, what do you usually do?
How do you usually debug an application?
Debuggers / Prints
- How do you usually do to improve the performance of a particular method?
Calculate $O()$ notation, optimize accordingly

Phase 2: Description of Method Execution Report

Method Execution Report is a textual and interactive report that summarizes the execution of a particular method for a given software execution. The report provides an overview of the dynamic calls and time consumption.

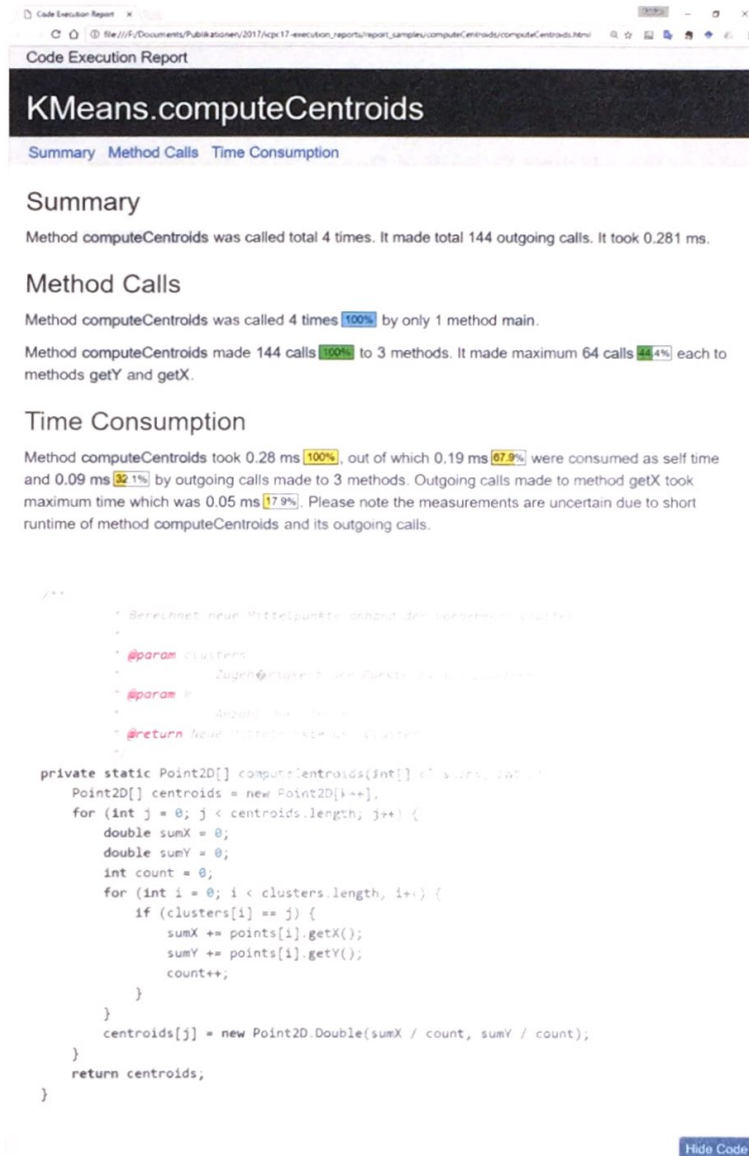


Figure on the left gives an example of the report. The report is structured into three sections, and lists the code of the method.

The summary gives first essential data, including the number of incoming and outgoing calls, and how long the summed executions of the method took.

The second section details the description of the method calls, including the most important caller and callees.

The third part provides further information about the timing.

Phase 3:

TIME BEGIN: 17:52

QUESTION 1:

What do you think about the content of the textual description?

- I find it easy to understand? (strongly agree, agree, neutral, disagree, strongly disagree)
Please, justify

Agree. The Summary is alright. The Method calls section shows partial ~~stats~~ statistics, I'd rather see them for all methods (maybe by toggling a button)

- I find it useful? (strongly agree, agree, neutral, disagree, strongly disagree)
Please, justify

Strongly Agree. It gives a very good idea about the time the method took to run. I'd like to see an example where the method was run more than 1 time.

QUESTION 2:

What do you think about the interaction and the visual elements offered by the report?

- I find them easy to understand? (strongly agree, agree, neutral, disagree, strongly disagree)
Please, justify

Neutral.
It took me a little bit to understand the percentage bars. They are more straight forward to understand when you can see all the corresponding parts: m1: 12.25% (in a vertical list)
m2: 17.5%

- I find them useful? (strongly agree, agree, neutral, disagree, strongly disagree)
Please, justify

Agree. Once I understood them, they provided me with a clear visual support.

QUESTION 3:

Overall, do you feel that such a report is useful?

(strongly agree, agree, neutral, disagree, strongly disagree)

Please, justify

Strongly agree.
However, as I said before, I would like to have the possibility to see a complete report, with all the methods. This is a highlight.

QUESTION 4:

In what scenarios and for solving which maintenance tasks would developers use Method Execution Reports?

Please, justify

Code profiling for ^{exewhon time} ~~time execution~~, wanting to decrease execution time.

QUESTION 5:

What tools would you use instead of Method Execution Reports to retrieve the same information?

Please, justify

I don't know of anything similar to this. I would measure time in the code and it would be ~~slower~~ slower to get the data:

```
t1 = time.now()
method
dt = time.now() - t1
print(method_name + dt)
```

QUESTION 6:

Do you have any suggestion on how to improve the report? Any critic?

Please, justify

Already said, the option to show a full report, with all the called methods and their time %, not just the larger / slower one.

TIME END: 18:08