

Labyrinth Profiles

Autoren: POS

Inhaltsverzeichnis

1 Voraussetzung	2
2 Durchführung	2
2.1 Java	2
2.2 Python	3

1 Voraussetzungent

- **Startpunkt:** X-Coord.: 1; Y-Coord: 1
- **Labyrinth:** l3.txt

2 Durchführung

2.1 Java

Nachdem die vorausgesetzten Variablen gesetzt wurden kann die Zeitmessung mit dem Profiler gestartet werden. Dazu drücke ich in der IntelliJ IDE einfach nur auf die 3 Punkte oben rechts und wähle dann „Profile ‚Labyrinth.java‘ with ‚IntelliJ Profiler‘“

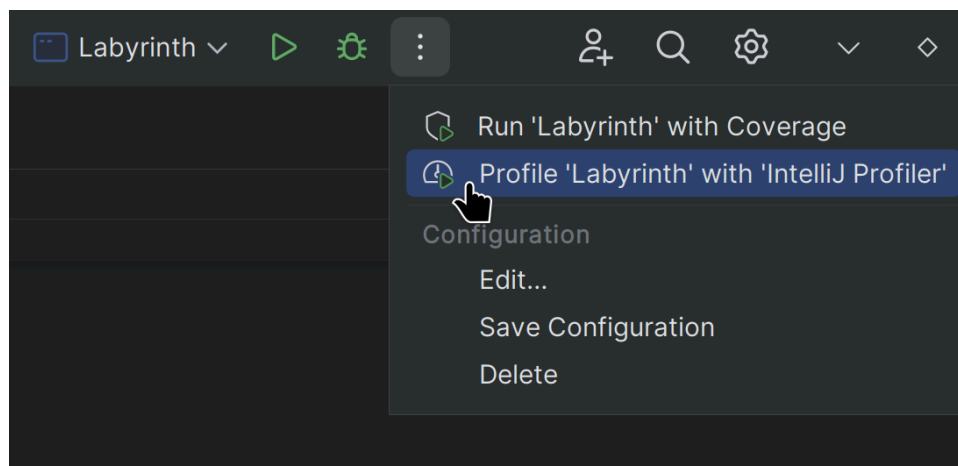


Abbildung 1: Aufrufen des IntelliJ Profilers

Nun können wir das Ergebniss einsehen. Es ist zu sehen das Java etwa **10ms** benötigt hat um den Weg zu finden.

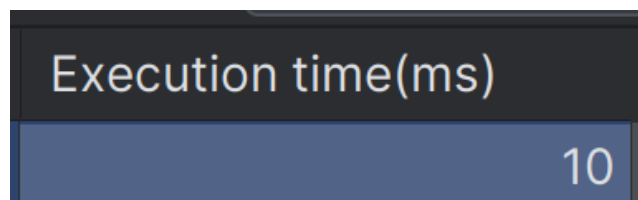


Abbildung 2: Ergebniss des IntelliJ Profilers

2.2 Python

Nun machen wir das selbe mit dem Python Programm. Dazu müssen wir aber zuerst noch die „Run-Paramter“ setzten. Das kann ich in der „Run-Configuration“ machen.

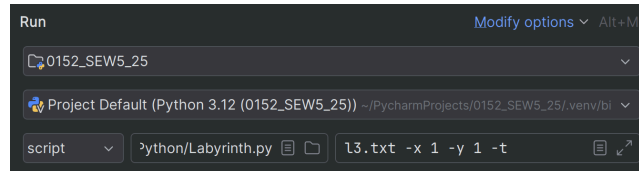


Abbildung 3: Einstellen der Python „Run-Configuration“

Danach rufe ich den Profiler auf.

Profiler 0152_SEW5_25	
Flame Graph	Call Tree
Method List	Statistics
Call Graph	
Name	Time (ms)
alleSuchen	6287.3%

Abbildung 4: rgebniss des PyCharm Profilers

Man kann sehen das Python deutlich langsamer läuft.