

Vorlesung 10

Aufgabe 1

Discuss Donald Knuth's original quote about premature optimization (page 16). What is your opinion?

Ich stimme Knuths Meinung größtenteils zu. Sie passt gut zum Pareto-Prinzip: mit 20 % Aufwand erreichst du 80 % des Ergebnisses – aber um die letzten 20 % zu bekommen, brauchst du 80 % mehr Mühe. Ich glaube, das macht Sinn, nicht vorzeitig alles sofort perfekt zu machen

Ich glaube, es macht Sinn datengetrieben und nicht intuitionsgetrieben zu denken, denn egal wie gut und erfahren ein Programmierer sein mag - kann es nicht wie Computer das Große und Ganze sehen - dafür muss man zu Profilen greifen um genau zu verstehen, wo die Ressourcen wirklich verbraucht werden. Es macht auf jeden Fall Sinn zuerst funktionalen, sauberen Code zu schreiben und dann datengestützt zu optimieren, statt blind nach Effizienz zu streben

Wieso stimme ich seiner Meinung nur „größtenteils“ zu? Ich glaube es gibt richtige Code-Freaks (im guten Sinne des Wortes), die genau dadurch so „besonders“ sind, weil sie von Anfang an an absolut alles denken. Give it all you got! Aber für die breiten Massen ist die Knuths Regel schon sehr sinnvoll :)

Aufgabe 2

Select one slide from the lecture, research more about the topic, and report on it.

Ich fand cool, dass einige der besten Programmierer keine Debugger nutzen, sondern Probleme durch „old-school“ Methoden aufdecken. Linus Torvalds finde ich sehr beeindruckend und er meint, dass man den Code so gut verstehen sollte, dass man Fehler durch Nachdenken und gezielte Tests findet, anstatt sich auf ein externes Werkzeug zu stützen. Ich kann leider die Quelle nicht angeben, das habe ich irgendwo auf YouTube gesehen.

Als ich angefangen habe zu coden, gab es noch kein ChatGPT, ich habe mich nicht getraut nach Hilfe zu bitten (Corona Semester, Wirtschaftsinformatik, absolut keine Vorstellung von Informatik) und musste alleine klarkommen. Da habe ich fast nach jeder Zeile Prints gesetzt, um jeden kleinsten Schritt nachvollziehen zu können. Das hat mir extrem geholfen, den Datenfluss besser zu verstehen. Aber dann kam ChatGPT raus und dann war's das mit Nachdenken =)))) Ich sehe das als sehr-sehr großes Problem bei mir, denn ich bin extrem faul geworden und habe sehr verlernt Code zu schreiben, deswegen kann ich auf jeden Fall nachvollziehen, wieso die berühmten Programmierer sogar auf Debugger verzichten - viele Tools sind eine Art „Krücke“, die einen daran hindert, selber tiefes Verständnis zu erlangen. Wenn ich die Zeit zurückdrehen könnte, würde ich ChatGPT zum Coden NIE verwenden. Es ist einfach wie Droge, die dein Geist bricht. „Ah komm schon, so viel Zeit musst du gar nicht verschwenden, zeig doch mal wo es klemmt, dann ist es schnell vorbei“ :D