

Abgabe am Freitag, den 10. Januar 2025, 12 Uhr

Dieser Zettel enthält nur Zusatzaufgaben. Das KdP-Team wünscht frohe Weihnachten und ein gutes neues Jahr.

Aufgabe 1 Wiederholung I

10 Zusatzpunkte

- (a) Was sind die wesentlichen Komponenten der von-Neumann-Architektur?
- (b) Was ist das zentrale Konzept der imperativen Programmierung?
- (c) Was ist der Zusammenhang zwischen dem imperativen Programmierparadigma und der von-Neumann-Architektur?
- (d) Welche Varianten gibt es bei der Interpretation von Zuweisungen in imperativen Programmiersprachen? Wodurch zeichnen sich diese Varianten aus?
- (e) Was sind die Herausforderungen, wenn es darum geht, Ein-Ausgabe Anweisungen für eine imperative Programmiersprache zu entwerfen?

Aufgabe 2 Autocomplete

10 Zusatzpunkte

Erinnern Sie sich an das Autocomplete-Programm aus der Methodenwoche. Hier wollen wir zwei Möglichkeiten betrachten, um die Funktionalität zu verbessern:

- (a) Man könnte statt nur einem Vorgängerwort die Häufigkeiten für *zwei* Vorgängerwörter berechnen.
- (b) Man könnte beim Kontext häufige Wörter wie *the*, *of*, *by*, usw. ausfiltern.

Modifizieren Sie das Autocomplete-Programm so, dass es diese beiden Ideen umsetzt. Testen Sie Ihre Implementierung und diskutieren Sie Vor- und Nachteile.

Aufgabe 3 Wiederholung II

10 Zusatzpunkte

- (a) Nennen Sie mindestens drei verschiedene Programmierparadigmen.
- (b) Was ist das zentrale Konzept der funktionalen Programmierung?
- (c) Wozu dienen Datentypen?
- (d) Nennen Sie einen Vor- und einen Nachteil von statisch typisierten Programmiersprachen gegenüber dynamisch typisierten Programmiersprachen.
- (e) Nennen Sie einen Vor- und einen Nachteil von call-by-reference gegenüber call-by-value.

Aufgabe 4 ASCII-Banner

10 Zusatzpunkte

Der ASCII-Zeichensatz besteht aus 255-Zeichen, die häufig auf Rechnern verwendet werden. Die ASCII-Zeichen werden typischerweise durch 16 Zeilen von jeweils 8 Pixeln dargestellt. Die Datei `glyph.txt` enthält die Darstellung der Zeichen des ASCII-Codes als Textdatei.

Schreiben Sie ein Python-Programm, das die Glyphs aus `glyph.txt` einliest und die folgende Funktionalität bereitstellt: Gegeben eine Zeichenkette `s`, stelle die Zeichen von `s` dar als Konkatenation der entsprechenden Glyphs aus `glyph.txt` (als Text), und gib das Ergebnis in eine Datei aus. Das Ergebnis wird auch als