

## Aufgabe 1

### a) Was ist Typisierung?

Die Typisierung ist eine Form, wie Daten gehandhabt werden. Mittels Abstraktionsebenen kann das Schreiben von Software vereinfacht werden [1]. Typisierung teilt sich einmal in die statische, wie auch die dynamische Typisierung auf.

a) Welche Art von Typisierung nutzt Python und wie unterscheidet sich Python hier von anderen Programmiersprachen?  
Python nutzt die dynamische Typisierung. Gemäß der Universität Regensburg sei Python eine Programmiersprache, die eine Ganzzahl  $x$  an einer Stelle des Programms als Variable speichern und an späterer Stelle ein String als Variable speichern könne. Der in Python festgelegte Interpreter versuche daraufhin den geeignetsten Datentypen zu erraten [2].

b) Worin besteht der Unterschied zwischen einer kompilierten und einer interpretierten Programmiersprache?  
Compiler und Interpreter weisen strukturelle Ähnlichkeiten auf. Der entscheidende Unterschied liege vor allem darin, dass ein Compiler den erzeugten Code in einer Datei für spätere Ausführung abspeichere, während ein Interpreter den erzeugten Code unmittelbar in Echtzeit ausführe [3]. Das lässt sich auf die dazugehörigen Programmiersprachen ableiten.

c) Was ist der Unterschied zwischen Bit und Byte?  
Ein Byte entspricht genau 8 Bit, wobei ein Bit nur exakt „zwei gültige diskrete Signalzustände annehmen“ kann, beispielsweise Wahr oder Falsch, 1 oder 0. Beide Einheiten sind gleichzeitig auch „Mengeneinheiten für Daten“. Ein Tupel von 8 Bit (Oktett) entspricht somit einem Byte [4].

d) Erklären Sie das EVA-Prinzip. (Bedeutung und eine kurze Erklärung)  
Das EVA-Prinzip stellt ein Prinzip der Datenverarbeitung dar. Dabei beginnt es mit der Eingabe (E), in der Informatik in Form von z.B. Maus oder Tastatur, daraufhin folgt die Verarbeitung (V) im Prozessor und die Speicherung im Speicher eines Geräts, während zum Schluss die verarbeitenden Daten in einer definierten Form wieder ausgegeben werde, z.B. in Form eines Sensors [4].

## Aufgabe 2

### a) Type Error

Ein Type Error tritt dann auf, wenn eine Verarbeitung mit einem falschen Datentyp stattfindet, die nicht verarbeitet werden kann. Das Weitergeben von Argumenten des falschen Typs (zum Beispiel das Weitergeben einer Liste, wenn eine Ganzzahl erwartet wird), sollte zu einem Type Error führen [5].

### b) Index Error

Ein Index Error tritt dann auf, wenn auf einen Index außerhalb des gültigen Bereichs einer Sequenz zugegriffen werden möchte. Ein Beispiel wäre die Ausgabe des Elements an zweiter Stelle einer leeren Liste [5].

### c) Syntax Error

Dieser Fehler tritt dann auf, wenn gemäß den Vorgaben der Programmiersprache eine falsche oder unvollständige Angabe verwendet wurde. Dies kann bei vielen Eingaben, nicht

nur bei der Standardeingabe, sondern auch bei weiteren Syntax Fehlern auftreten, wie z.B. falschem Filenamen [5].

d) Name Error

Ein Name Error tritt dann auf, wenn auf eine Variable oder eine Funktion zugegriffen wird, die nicht definiert wurde.

Autor = Schenk Philipp, 7093700

#### Literaturverzeichnis:

[1] Willers, Darwin, „Typisierung in Programmiersprachen“, *Software-System-Entwicklung*, S. 10, 15. März 2020.

[2] Universität Regensburg Fakultät Physik, „Begleitscript zur Vorlesung im Wintersemester 2020/21“, *Programmieren in Python*, S. 21, 02. März 2021.

[3] Singer, Lukas, „Entwicklung und Konzeption einer interpretierten Programmiersprache“, *DIPLOMARBEIT*, S. 10, 10. Januar 2016.

[4] Kaschube, Matthias und Matthäus, Franziska und Tolle, Karsten, *Grundlagen der Programmierung. Teil 1 – Computer*. Frankfurt am Main: Goethe-Universität Frankfurt, 2023.

[5] Python Software Foundation. (23. Oktober 2023). *Built-in Exceptions*. Available URL: <https://docs.python.org/3/tutorial/errors.html>