

## 1.1

### (a)

Die von-Neumann-Architektur beschreibt eine Computer Architektur und wurde 1945 von dem Mathematiker und Physiker John von Neumann erstellt. Die Grundprinzipien sind die Aufteilung eines Computers in 5 Funktionseinheiten (Speicher, Steuerwerk, Rechenwerk, Eingabe und Ausgabe), dass im Speicher sowohl das Programm als auch Daten abgelegt werden und einen Befehls-Ausführungszyklus.[2] Die entscheidende Schwachstelle ist der sogenannte "Von Neumann Flaschenhalz"[5]. Da sowohl Daten als auch Programme auf dem selben Speicher liegen, wird bei einem Zugriff auf eines der beiden immer der gleiche Bus verwendet, was den "Durchfluss" von Daten stark einschränkt.

In der Harvard Architektur kann die CPU gleichzeitig auf Daten und Programme zugreifen, da diese auf unterschiedlichen Speichermedien liegen. [1]

### (b)

Die Typisierung in einer Programmiersprache schränkt den Wertebereich einer Variable ein. Dabei unterscheidet man insbesondere zwischen starker und dynamischer Typisierung. [4] Python nutzt hierbei eine dynamische Typisierung.

## 1.2

### (a)

Die Funktion `sum(iterable,[,start])` summiert *start* und alle Elemente des *iterable* auf und gibt das Ergebnis zurück. [3]

Beispiel:

```
1 print(sum([1,2,3])) # 6
```

### (b)

- **TypeError** Wird geworfen, wenn eine Operation oder eine Funktion auf ein Objekt von unpassendem Typ ausgeführt werden soll. Beispiel:

```
1 print(sum(1,2,3))
```

- **IndexError** Wird geworfen, wenn ein Index "out of range" ist. Beispiel:

```
1 print([1,2,3][3])
```

- **SyntaxError** Wird geworfen, wenn der Parser einen Syntaxerror findet. Beispiel:

```
1 print("Hello World)
```

- **NameError** Wird geworfen, wenn ein Name (von einer Variablen, Funktion etc.) nicht gefunden wird. Beispiel:

```
1 print(Hello)
```

[3]

## 1.3

### (a)

Das linke Fenster ist ein Terminal bzw. eine Shell, mit dem eine Vielzahl unterschiedlicher Funktionen in einer Linux-Umgebung ausgeführt werden können, z.B. können Programme gestartet werden. Auf der rechten Seite ist eine Python Console zu sehen, auf der Python Befehle ausgeführt werden können.

### (b)

In einer Shell sind z.B. die Befehle `ls`, `cd` oder `echo` typisch, in einer Python Console sind natürlich alle Python-Funktionen wie `print()`, `import` oder `input()` möglich.

## 1.4

### (a)

- **Determiniertheit** Da in diesem Algorithmus kein Zufall eine Rolle spielt, wird bei jedem Durchlauf (mit der gleichen Liste) auch das gleiche Ergebnis herauskommen.
- **Deterministisch** Auch dies ist gegeben, da an jedem Punkt nur die Optionen "Betrachte das nächste Element" oder "Beende den Algorithmus" gibt.
- **Statische Finitheit** Dies ist gegeben, da sich der Algorithmus in vier Zeilen beschreiben ließ.
- **Dynamische Finitheit** Es wird nur die kleinste Zahl, die Liste, und ein Index gespeichert.
- **Terminiertheit** Solange die Liste nicht unendlich lang ist, wird der Algorithmus nach einer endlichen Zeit kontrolliert beendet sein.

[2]

### (b)

**Eingabe:** Eine Liste aller 8 Karten.

**Ausgabe:** Eine sortierte Liste aller 8 Karten.

**Vorschrift (Bubblesort):**

Vergleiche die ersten beiden Karten miteinander, ist der Wert der zweiten kleiner, vertausche sie. Betrachte nun Karten zwei und drei, ist der Wert der dritten Karte kleiner, vertausche sie. Führe dies weiter durch einschließlich der letzten beiden, die Karte mit dem höchsten Wert hat nun Index 7. Führe diesen Algorithmus wieder durch für die ersten 7, dann für 6, 5 ... bis nur noch eine Karte übrig ist. Wenn zwei Karten den gleichen Wert haben, sortiere entsprechend der angegebenen Reihenfolge.

## Literatur

- [1] *Harvard architecture*. URL: [https://en.wikipedia.org/wiki/Harvard\\_architecture](https://en.wikipedia.org/wiki/Harvard_architecture) (besucht am 25.10.2017).
- [2] Prof. Dr. Detlef Krömker. *Vorlesung 1: Computer - Algorithmus - Programm*.
- [3] *Python Docs*. URL: <https://docs.python.org/3/> (besucht am 25.10.2017).
- [4] *Typisierung*. URL: [https://de.wikipedia.org/wiki/Typisierung\\_\(Informatik\)](https://de.wikipedia.org/wiki/Typisierung_(Informatik)) (besucht am 25.10.2017).
- [5] *Von Neumann architecture*. URL: [https://en.wikipedia.org/wiki/Von\\_Neumann\\_architecture](https://en.wikipedia.org/wiki/Von_Neumann_architecture) (besucht am 25.10.2017).