

Abschlussklausur

Python Programmierung III (Alternative)

27. Februar 2025

Name: _____

Vorname: _____

Unterschrift: _____

Aufgabe	Max. Punkte	Erreichte Punkte
??	4	
??	6	
??	6	
??	8	
??	7	
Gesamt	31	

1 Python Fundamentals & Environment Setup (4 Punkte)

1. Erklären Sie, **was** ein virtuelles Python-Environment ist und **warum** es in Projekten sinnvoll ist. (2 Punkte)

2. Welche Befehle verwenden Sie, um ein virtuelles Environment mit **conda** zu **erstellen** und zu **aktivieren**? Schreiben Sie den entsprechenden Befehl. (2 Punkte)

Hinweis: Verwenden Sie z. B. den Befehl zur Erstellung eines neuen Environments namens `python_kurs_3`.

2 Objektorientierte Programmierung (OOP) (6 Punkte)

1. Erklären Sie das Konzept der **Kapselung** in der objektorientierten Programmierung. Warum ist sie nützlich? Nennen Sie ein Beispiel, wie Kapselung in Python umgesetzt wird. (2 Punkte)

2. Nennen Sie zwei Entwurfsmuster und erklären Sie einen von ihnen. (2 Punkte)

3. Der folgende Code enthält einen Fehler in Bezug auf **private Attribute** und **abstrakte Methoden**. Finden Sie die zwei Fehler und erklären Sie, warum sie problematisch sind. (2 Punkte)

```
1 from abc import ABC, abstractmethod
2
3 class Fahrzeug(ABC):
4     def __init__(self, marke, modell):
5         self.__marke = marke
6         self.__modell = modell
7
8     @abstractmethod
9     def beschreibung(self):
10        pass
11
12 class Auto(Fahrzeug):
13     def __init__(self, marke, modell, ps):
14         super().__init__(marke, modell)
15         self.ps = ps
16
17     def auto_beschreibung(self):
18         return f"{self.__marke} {self.__modell} mit {self.ps} PS"
19
20 # Objekt erstellen
21 auto1 = Auto("BMW", "X5", 300)
22 print(auto1.beschreibung())
```

Hinweis: Sie müssen keinen Code schrieben, sondern nur die Fehler finden und erklären.

3 Algorithmen, Effizienz und Rekursion (6 Punkte)

1. Erklären Sie, warum ein iterativer Ansatz manchmal gegenüber einem rekursiven bevorzugt wird. Nennen Sie jeweils einen Vorteil. (2 Punkte)

2. Implementieren Sie eine rekursive Funktion zur Berechnung von a^b , wobei a die Basis und b der Exponent ist. Ergänzen Sie die Basisfallbedingung. (2 Punkte)

```
1 def power(a, b):
2     if _____:
3         return _____
4     return a * power(a, b - 1)
5
6 print(power(2, 3)) # Erwartete Ausgabe: 8 (2^3 = 2 * 2 * 2)
```

3. Der folgende Python-Code sucht das größte Element in einer Liste. Analysieren Sie den Code und geben Sie die **Zeitkomplexität** des Algorithmus an. (2 Punkte)

```
1 def find_max(arr):
2     max_value = arr[0]
3     for i in range(1, len(arr)):
4         if arr[i] > max_value:
5             max_value = arr[i]
6     return max_value
7
8 # Beispiel:
9 print(find_max([3, 7, 2, 9, 5])) # Erwartete Ausgabe: 9
```

Hinweis: Bitte geben Sie nur die Zeitkomplexität in Big-O-Notation an.

4 NumPy (8 Punkte)

1. Nennen Sie zwei Funktionen zur Erstellung von NumPy-Arrays und erläutern Sie kurz, worin sie sich unterscheiden. (2 Punkte)
2. Was bewirkt der Befehl `np.random.randint(0, 100, (3,4))`? (1 Punkt)
3. Vervollständigen Sie den folgenden Code, um ein Array mit 10 gleichmäSSig verteilten Werten zwischen 5 und 15 zu erzeugen. (1 Punkt)

```
1 import numpy as np
2 array_linspace = np.linspace(____, ____, ____)
3 print(array_linspace)
```

Listing 1: Ergänzen Sie den Code

4. Was gibt folgender Code aus? Begründen Sie kurz Ihre Antwort. (2 Punkte)

```
1 import numpy as np
2 a = np.array([2, 4, 6])
3 print(a * 3)
```

Listing 2: Ausgabevorhersage

5. Erklären Sie, was Broadcasting in NumPy bedeutet, und nennen Sie ein typisches Anwendungsbeispiel. (2 Punkte)

5 Pandas (7 Punkte)

1. Erklären Sie den Unterschied zwischen einer **Series** und einem **DataFrame** in Pandas. Wann würden Sie welche Datenstruktur verwenden? (2 Punkte)
2. Welche Methode verwenden Sie, um die ersten 5 Zeilen eines DataFrames anzugeben? (1 Punkt)
3. Vervollständigen Sie den folgenden Code, um den Median der Spalte `total_bill` im `tips`-Datensatz zu berechnen. (1 Punkt)

```
1 import pandas as pd
2 df = pd.read_csv("tips.csv")
3 median_total = df["total_bill"].                ()
4 print(median_total)
```

Listing 3: Berechnung des Medians

4. Schreiben Sie ein Code-Snippet, das den `tips`-Datensatz einliest und alle Zeilen anzeigt, bei denen das Trinkgeld mehr als 10% des Gesamtbetrags beträgt. (2 Punkte)

```
1 df = pd.read_csv("tips.csv")
2 df_filtered = df[                            ]
3 print(df_filtered.head())
```

Listing 4: Filterung im tips-Datensatz

5. Beschreiben Sie die wesentlichen Schritte, um in Pandas eine Zeitreihe zu analysieren (Umwandlung einer Datumsspalte in ein Datetime-Objekt, Setzen als Index, Resampling). (1 Punkt)

Hinweis

Überprüfen Sie Ihre Antworten sorgfältig. Ihre Erklärungen und Code-Snippets sollten klar und vollständig formuliert sein.