

Abschlussklausur

Python Programmierung III

14. Februar 2025

Allgemeine Hinweise

Beantworten Sie alle Fragen schriftlich. Es sind keinerlei Hilfsmittel (Computer, Bücher, Internet) erlaubt. Viel Erfolg!

Aufgabe	Max. Punkte	Erreichte Punkte
1. Python Fundamentals & Environment Setup	4	
2. Objektorientierte Programmierung (OOP)	6	
3. Algorithmen, Effizienz und Rekursion	6	
4. NumPy	6	
5. Pandas	7	
Gesamt	29	

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 `myenv`.

2 Objektorientierte Programmierung (OOP) (6 Punkte)

1. Erklären Sie den Unterschied zwischen statischen Methoden und Klassenmethoden in Python. Geben Sie ein Beispiel, wann eine statische Methode sinnvoll ist. (2 Punkte)
2. Der folgende Code enthält einen Fehler. Finden Sie den Fehler und korrigieren Sie den Code, sodass die Methode `beschreiben` den Namen der Person korrekt ausgibt. (2 Punkte)

```
1 class Person:  
2     def __init__(self, name):  
3         self.name = name  
4  
5     def beschreiben(self):  
6         print("Die Person heit " + Name)  
7  
8 p = Person("Laura")  
9 p.beschreiben()
```

Listing 1: Fehlersuche in einer Klassenmethode

3. Nennen Sie ein praktisches Beispiel, in welchem das Singleton-Muster sinnvoll eingesetzt wird. (2 Punkte)

3 Algorithmen, Effizienz und Rekursion (6 Punkte)

1. Vergleichen Sie die Zeitkomplexitäten von Bubble Sort und Quick Sort in Ihren eigenen Worten. (2 Punkte)
2. Der folgende rekursive Code berechnet die n-te Fibonacci-Zahl. Es fehlt eine Basisfallbedingung. Vervollständigen Sie den Code. (2 Punkte)

```
1 def fibonacci(n):
2     if _____:
3         return n
4     else:
5         return fibonacci(n-1) + fibonacci(n-2)
6
7 print(fibonacci(6)) % Erwartete Ausgabe: 8
```

Listing 2: Rekursiver Algorithmus zur Berechnung der Fibonacci-Zahl

3. Beschreiben Sie, warum es bei rekursiven Algorithmen wichtig ist, einen Basisfall zu definieren, und wie Memoization die Effizienz verbessern kann. (2 Punkte)

4 NumPy (6 Punkte)

1. Erklären Sie, warum NumPy-Arrays für numerische Berechnungen gegenüber Python-Listen bevorzugt werden. (2 Punkte)
2. Betrachten Sie folgenden Code. Geben Sie die erwartete Ausgabe an und begründen Sie, warum. (1 Punkt)

```
1 import numpy as np
2 a = np.array([1, 2, 3])
3 b = np.array([[10], [20], [30]])
4 result = a + b
5 print(result)
```

Listing 3: Broadcasting-Beispiel

3. Vervollständigen Sie den folgenden Code, um ein 3x3 Array mit Zufallszahlen zwischen 0 und 1 zu erzeugen. (1 Punkt)

```
1 import numpy as np
2 array_random = np.random.-----((3,3))
3 print(array_random)
```

Listing 4: Ergänzen Sie den Code

4. Was bedeutet der Begriff Broadcasting in NumPy? Geben Sie eine kurze Erklärung und ein Beispiel. (2 Punkte)

5 Pandas (7 Punkte)

- Erklären Sie den Unterschied zwischen einer Series und einem DataFrame in Pandas. Wann würden Sie welche Datenstruktur verwenden? (2 Punkte)
- Betrachten Sie den folgenden Codeausschnitt. Was ist die Ausgabe, wenn dieser Code ausgeführt wird? (1 Punkt)

```

1 import pandas as pd
2 data = {'Name': ['Anna', 'Bernd', 'Clara', 'David'],
3          'Alter': [28, 34, 29, 42],
4          'Stadt': ['Berlin', 'München', 'Hamburg', 'Köln']}
5 df = pd.DataFrame(data)
6 print(df[df['Alter'] > 30])

```

Listing 5: DataFrame-Filterung

- Vervollständigen Sie den folgenden Code, der eine CSV-Datei einliest und alle Zeilen ausgibt, in denen der total_bill gröSSer als 20 ist. (1 Punkt)

```

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

```

Listing 6: Bedingte Filterung im tips-Datensatz

- Beschreiben Sie, wie Sie in Pandas fehlende Werte erkennen und behandeln können. Erklären Sie den Unterschied zwischen `fillna()` und `dropna()`. (2 Punkte)
- Vervollständigen Sie den folgenden Code, um eine neue Spalte `Tip_Percentage` zu erstellen, die den Prozentsatz des Trinkgelds am Gesamtbetrag berechnet. (1 Punkt)

```

1 df = pd.read_csv("tips.csv")
2 df["Tip_Percentage"] = -----
3 print(df[["total_bill", "tip", "Tip_Percentage"]].head())

```

Listing 7: Ergänzen Sie den Code zur neuen Spalte in tips.csv

- Beschreiben Sie in eigenen Worten, wie Sie eine Zeitreihe in Pandas analysieren würden. Welche Schritte sind dabei notwendig? (1 Punkt)

Hinweis

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