

Vorbereitungsklausur

Python 3

12. Februar 2025

Name: _____

Vorname: _____

Unterschrift: _____

Aufgabe	Max. Punkte	Erreichte Punkte
1	4	
2	6	
3	6	
4	6	
5	7	
Gesamt	29	

1 Python Umgebungen (4 Punkte)

1. Erklären Sie mit eigenen Worten, **was** ein virtuelles Environment ist und **warum** es nützlich ist. (**2 Punkte**)
 2. Welche Befehle verwenden Sie, um ein virtuelles Environment zu **erstellen** und zu **aktivieren** mit venv in Python? (**2 Punkte**)

2 Objektorientierte Programmierung (OOP) (6 Punkte)

1. Erklären Sie mit eigenen Worten, **was** sind abstrakte Klassen und **warum** sind sie nützlich? Geben Sie ein Beispiel für eine abstrakte Klasse in Python. (2 Punkte)

2. Der folgende Code definiert eine Klasse `Person` mit einem privaten Attribut `__password`. Ergänzen Sie die Methode `change_password`, sodass sie das Passwort nur dann ändert, wenn das alte Passwort korrekt eingegeben wurde. Falls das alte Passwort nicht stimmt, soll eine Fehlermeldung ausgegeben werden. (2 Punkte)

```
1 class Person:  
2     def __init__(self, name, password):  
3         self.name = name  
4         self.__password = password # Privates Attribut  
5  
6     def change_password(self, old_password, new_password):  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18 .
```

Listing 1: Methode `change_password` ergänzen

3. Erklären Sie den Zweck der Funktion `super()` in einer UnterkLASSE? (1 Punkt)

4. Nennen Sie zwei Entwurfsmustern. (1 Punkt)

3 Algorithmen, Effizienz und Rekursion (6 Punkte)

1. Erklären Sie den Unterschied zwischen linearer Suche und binärer Suche. Geben Sie die Zeitkomplexität für lineare Suche. (2 Punkte)

2. Der folgende Code soll die Summe aller natürlichen Zahlen von 1 bis n rekursiv berechnen. Ergänzen Sie die fehlende Zeile für die Basisfallbedingung. (2 Punkt)

```
1 def summe(n):
2
3
4
5
6
7     return n + summe(n-1)
8
9 print(summe(5)) # Erwartete Ausgabe: 15
```

Listing 2: Rekursiver Algorithmus zur Summenberechnung

Hinweis: Der Basisfall ist erforderlich, um eine unendliche Rekursion zu vermeiden.

3. Der folgende Python-Code implementiert den Sortieralgorithmus **Insertion Sort**. Analysieren Sie den Code und geben Sie die Zeitkomplexität des Algorithmus an. (2 Punkte)

```
1 def insertion_sort(arr):
2     for i in range(1, len(arr)):
3         key = arr[i]
4         j = i - 1
5         while j >= 0 and arr[j] > key:
6             arr[j + 1] = arr[j]
7             j -= 1
8         arr[j + 1] = key
9     return arr
```

Listing 3: Sortieralgorithmus analysieren

4 NumPy (6 Punkte)

1. Erklären Sie den Unterschied zwischen einem NumPy-Array und einer Python-Liste. Welche Vorteile bieten NumPy-Arrays gegenüber Python-Listen? (2 Punkte)

2. Betrachten Sie folgenden Code. Geben Sie die erwartete Ausgabe an und begründen Sie Ihre Antwort. (1 Punkt)

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

Listing 4: Broadcasting-Beispiel

3. Ergänzen Sie den folgenden Code, um eine Matrixmultiplikation zwischen den beiden NumPy-Arrays **a** und **b** durchzuführen. (1 Punkte)

```
1 import numpy as np
2
3 a = np.array([[1, 2], [3, 4]])
4 b = np.array([[5, 6], [7, 8]])
5
6 result = -----
7 print(result)
```

Listing 5: Matrixmultiplikation

4. Was versteht man unter Broadcasting in NumPy? Erklären Sie anhand eines Beispiels. (2 Punkte)

5 Pandas (7 Punkte)

1. Erklären Sie den Unterschied zwischen einer Series und einem DataFrame in Pandas. Wann würden Sie eine Series verwenden und wann ein DataFrame? (2 Punkte)

2. Betrachten Sie den folgenden Code. Was wird ausgegeben, wenn der Code ausgeführt wird? (1 Punkt)

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

Listing 6: DataFrame-Filterung

3. 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 7: Bedingte Filterung im tips-Datensatz

4. Beschreiben Sie, wie man in Pandas fehlende Werte erkennt und behandelt. Was ist der Unterschied zwischen `fillna()` und `dropna()`? (2 Punkte)

5. Ergänzen 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 8: Erstellung einer neuen Spalte in tips.csv