Abschlussklausur
Python Programmierung III

14. April 2025

Name: _____

Vorname: _____

Unterschrift: _____

Aufgabe	**Max. Punkte**	**Erreichte Punkte**
1	4	
2	6	
3	6	
4	8	
5	8	
Gesamt	32	

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 `venv` 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 den Unterschied zwischen **statischen** Methoden und **Klassenmethoden** in Python. (2 Punkte)

2. Nennen Sie ein Entwurfsmuster. (1 Punkte)

3. Ist die folgende Aussage wahr oder falsch? (1 Punkt)

   "Das Entwurfsmuster Singleton wird verwendet, um sicherzustellen, dass eine Klasse nur eine Instanz hat."

4. Der folgende Code enthält einen Fehler. Finden Sie den Fehler und korrigieren Sie ihn. (2 Punkte)

```python
from abc import ABC, abstractmethod

class Fahrzeug(ABC):
    def __init__(self, name, max_speed):
        self.name = name
        self.max_speed = max_speed

    @abstractmethod
    def fahren(self):
        pass

class Fahrrad(Fahrzeug):
    def __init__(self, name, max_speed, typ):
        super().__init__(name, max_speed)
        self.typ = typ

    def fahren(self):
        return f"Das Fahrrad {self.name} fährt mit maximal {self.
            max_speed} km/h."

class Auto(Fahrzeug): # Hinweis: Der Fehler liegt in dieser Klasse
    def __init__(self, name, max_speed, anzahl_sitze):
        super().__init__(name, max_speed)
        self.anzahl_sitze = anzahl_sitze
```

```python
auto = Auto("BMW", 220, 5)
```

Hinweis: Der Code enthält einen Fehler, der das Erstellen einer Instanz der Klasse `Auto` verhindert.

3 Algorithmen, Effizienz und Rekursion (6 Punkte)

1. Welche der folgenden Aussagen zur Zeitkomplexität von Bubble Sort ist **korrekt**? (1 Punkte)

   (a) Die Worst-Case-Zeitkomplexität von Bubble Sort ist $O(n^2)$, weil jedes Element mit jedem anderen verglichen werden kann.

   (b) Die Worst-Case-Zeitkomplexität von Bubble Sort ist $O(n \log n)$, weil der Algorithmus immer die Eingabe halbiert.

   (c) Bubble Sort ist in allen Fällen der effizienteste Sortieralgorithmus mit einer Laufzeit von $O(n)$.

   (d) Die Worst-Case-Zeitkomplexität von Bubble Sort ist $O(n)$, weil nur eine Schleife durchlaufen wird.

2. Der folgende rekursive Code berechnet die Summe aller geraden Zahlen von 1 bis n. Eine wichtige Bedingung fehlt, um sicherzustellen, dass nur gerade Zahlen addiert werden. Ergänzen Sie die fehlende Bedingung. (2 Punkte)

```python
def sum_even(n):
    if n <= 0:
        return 0
    if ____________________________:
        return n + sum_even(n-2)
    else:
        return sum_even(n-1)

print(sum_even(6))  # Erwartete Ausgabe: 12 (2 + 4 + 6)
```

Hinweis: Mit % können Sie den Rest einer Division berechnen.

3. Beschreiben Sie, warum es bei rekursiven Algorithmen wichtig ist, einen Basisfall zu definieren. (1 Punkt)

4. Der folgende Python-Code implementiert den Sortieralgorithmus `Selection Sort`. Analysieren Sie den Code und geben Sie die Zeitkomplexität des Algorithmus an. (2 Punkte)

```python
def selection_sort(arr):
    n = len(arr)
    for i in range(n):
        min_index = i
        for j in range(i + 1, n):
            if arr[j] < arr[min_index]:
                min_index = j
        arr[i], arr[min_index] = arr[min_index], arr[i]
    return arr
```

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

4 NumPy (8 Punkte)

1. Erklären Sie, warum NumPy-Arrays für numerische Berechnungen gegenüber Python-Listen bevorzugt werden. (2 Punkte)

2. Was ist die Aufgabe der folgenden NumPy-Funktionen? (2 Punkte)

   (a) `np.ones()`

   (b) `np.arange()`

3. Was ist die Ausgabe des folgenden Codes? (1 Punkt)

```
import numpy as np

a = np.eye(3)
print(a)
```

Hinweis: Die Funktion `np.eye()` erstellt eine Einheitsmatrix.

4. Was ist die Ausgabe des folgenden Codes? (3 Punkt)

```
import numpy as np

a = np.array([[1, 2, 3],
              [4, 5, 6]])

print(a.shape)
print(a[:, 2])
print(a[1, :].sum())
```

5 Pandas (8 Punkte)

1. Erklären Sie den Unterschied zwischen einer Series und einem DataFrame in Pandas. (2 Punkt)

2. Betrachten Sie den folgenden Codeausschnitt. Was ist die Ausgabe, wenn dieser Code ausgeführt wird? (1 Punkt)

```python
import pandas as pd

data = {'Name': ['Anna', 'Bernd', 'Clara', 'David'],
        'Alter': [28, 34, 29, 42],
        'Stadt': ['Berlin', 'München', 'Hamburg', 'Köln']}

df = pd.DataFrame(data)

print(df[df['Alter'] > 30])
```

3. Vervollständigen Sie den folgenden Code, um eine CSV-Datei einzulesen und nur die Rechnungen anzuzeigen, bei denen das Trinkgeld (`tip`) mehr als 15% der Gesamtrechnung (`total_bill`) beträgt. (2 Punkt)

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

Hinweis: Um den Prozentsatz zu berechnen, können Sie die folgende Formel verwenden: Prozentsatz = tip/total_bill × 100.

4. Was macht die methode `df.head(10)` in Pandas? (1 Punkt)

5. Was ist die Ausgabe des folgenden Codes? (2 Punkte)

```
import pandas as pd

data = {'Name': ['Anna', 'Bernd', 'Clara', 'David'],
        'Alter': [28, 34, 29, 42],
        'Stadt': ['Berlin', 'München', 'Hamburg', 'Köln']}
df = pd.DataFrame(data)

df['Geschlecht'] = ['w', 'm', 'w', 'm']
print(df)
```

Viel Erfolg bei der Klausur!