Abschlussklausur
Python Programmierung III (Zweite Wiederholung)

28. Februar 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 `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)

```python
from abc import ABC, abstractmethod

class Fahrzeug(ABC):
  def __init__(self, marke, modell):
    self.__marke = marke
    self.__modell = modell

  @abstractmethod
  def beschreibung(self):
    pass

class Auto(Fahrzeug):
  def __init__(self, marke, modell, ps):
    super().__init__(marke, modell)
    self.ps = ps

  def auto_beschreibung(self):
    return f"{self.__marke} {self.__modell} mit {self.ps} PS"

# Objekt erstellen
auto1 = Auto("BMW", "X5", 300)
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)

```python
def power(a, b):
    if ____________________________:
        return ___________
    return a * power(a, b - 1)

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)

```python
def find_max(arr):
    max_value = arr[0]
    for i in range(1, len(arr)):
        if arr[i] > max_value:
            max_value = arr[i]
    return max_value

# Beispiel:
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)

```
import numpy as np

array_linspace = np.linspace(____, ____, ____)

print(array_linspace)
```

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

```python
import numpy as np

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

print(a * 3)
```

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

5 Pandas (8 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 anzuzeigen? (1 Punkt)

3. Vervollständigen Sie den folgenden Code, um den `Median` der Spalte `total_bill` im `tips`-Datensatz zu berechnen. (1 Punkt)

```
import pandas as pd

df = pd.read_csv("tips.csv")
median_total = df["total_bill"].____________()

print(median_total)
```

4. Vervollständigen Sie den folgenden Code, um eine CSV-Datei einzulesen und nur die Rechnungen anzuzeigen, bei denen das Trinkgeld (`tip`) weniger als 10% 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.

5. Gegeben ist ein Pandas-DataFrame `df` mit einer Spalte `"Datum"`, die Datumswerte als Strings enthält. Ergänzen Sie den Code, um die Spalte in ein `datetime`-Format zu konvertieren und als Index zu setzen. (2 Punkt)

```
import pandas as pd

df["Datum"] = ____________________________
df = df.set_index("Datum")
```

Hinweis: Verwenden Sie die Methode `pd.to_datetime()`, um die Spalte in das richtige Format zu bringen.

Viel Erfolg bei der Klausur!