

Klausur: Python-Grundlagen, OOP, Algorithmik, NumPy und Pandas

(Keine Hilfsmittel erlaubt)

12. Februar 2025

Anleitung: Beantworten Sie alle Fragen in deutscher Sprache. Schreiben Sie Ihre Lösungen sauber und nachvollziehbar. Begründen Sie bei Bedarf Ihre Antworten mit kurzen Erklärungen. Keine Hilfsmittel (Skripte, Internet etc.) sind erlaubt. Bearbeitungszeit: 90 Minuten.

1. Python Fundamentals & Environment Setup (7 Punkte)

1. Virtuelle Umgebungen (3 Punkte)

- Erklären Sie den Zweck einer virtuellen Umgebung (`venv` oder `conda`). Warum ist es sinnvoll, eine zu verwenden?
- Wie können Sie in `conda` eine neue Umgebung erstellen und aktivieren? Nennen Sie die wichtigsten Befehle.

2. Jupyter Notebook (2 Punkte)

- Beschreiben Sie kurz den Aufbau eines Jupyter Notebooks. Welche Zelltypen gibt es?
- Welche Vorteile bieten Jupyter Notebooks gegenüber einfachen Python-Skripten?

3. Code-Fehler finden (2 Punkte) Betrachten Sie folgenden Python-Code:

```
my_var = "Hello"

if my-var == "Hello":
    print("Yes, it's Hello")
else:
    print("No, it's not Hello")
```

- Der Code erzeugt einen Fehler. Nennen Sie die Ursache und schlagen Sie eine Korrektur vor. (1 Punkt)
- Nehmen Sie an, Sie möchten eine weitere Zeile hinzufügen, die die Länge von `my_var` ausgibt. Schreiben Sie die fehlende Codezeile. (1 Punkt)

2. Object-Oriented Programming (OOP) (12 Punkte)

1. Grundlagen und Konzepte (4 Punkte)

- Was versteht man unter einer Klasse und einem Objekt? Nennen Sie ein kurzes Beispiel in Python. (2 Punkte)
- Erklären Sie kurz *Encapsulation*, *Inheritance* und *Polymorphism*. (2 Punkte)

2. Vererbung (4 Punkte)

- Wie erstellt man eine Unterklasse in Python? Geben Sie ein kleines Beispiel. (2 Punkte)
- Welche Bedeutung hat die Funktion `super` bei der Vererbung? (2 Punkte)

3. Code-Frage: Fehler erkennen (4 Punkte) Betrachten Sie folgenden Code-Ausschnitt:

```
class Animal:  
    def __init__(self, name):  
        self.__name = name  
  
    def speak(self):  
        print("...")  
  
class Dog(Animal):  
    def __init__(self, name, breed):  
        super().__init__(name)  
        self.breed = breed  
  
    def speak():  
        print(f"{self.__name} barks!")  
  
my_dog = Dog("Rex", "Labrador")  
my_dog.speak()
```

- Der Code kompiliert, aber es tritt ein Fehler zur Laufzeit auf. Erklären Sie den Fehler und schlagen Sie eine Lösung vor. (2 Punkte)
- Nach der Korrektur möchten Sie das Attribut `breed` als privat kennzeichnen. Wie ändern Sie den Code? (1 Punkt)
- Warum könnte das Anlegen privater Attribute (z.B. `__name`) sinnvoll sein? (1 Punkt)

3. Algorithmic Thinking, Efficiency und Rekursion (10 Punkte)

1. Big-O und Suchen (3 Punkte)

- Was versteht man unter $O(n)$ und $O(\log n)$ in Bezug auf die Laufzeit? (1 Punkt)
- Warum kann Binary Search nicht auf unsortierten Daten angewendet werden? (1 Punkt)
- Geben Sie ein Beispiel für einen Anwendungsfall von Linear Search. (1 Punkt)

2. Sortieralgorithmen (3 Punkte)

- (a) Erklären Sie das Grundprinzip von Bubble Sort. (1 Punkt)
- (b) Warum ist Bubble Sort meistens nicht effizient? (1 Punkt)
- (c) Welche Komplexität hat Quick Sort im schlechtesten Fall? (1 Punkt)

3. Rekursion (4 Punkte)

Unten sehen Sie eine unvollständige rekursive Funktion, die die Fakultät einer Zahl berechnet. Vervollständigen Sie den Code und erklären Sie den Basisfall.

```
def factorial(n):  
    # TODO: Basisfall ergänzen  
    if n < 0:  
        return None  
    else:  
        # TODO: Rekursionsfall ergänzen
```

- (a) Ergänzen Sie die fehlenden Zeilen, sodass `factorial(4)` den Wert 24 liefert. (2 Punkte)
- (b) Erklären Sie, was passiert, wenn Sie `factorial(-1)` aufrufen und warum. (1 Punkt)
- (c) Welche Rolle spielt Memoization in Rekursionsverfahren? (1 Punkt)

4. NumPy (10 Punkte)

1. Arrays erstellen und Attribute (3 Punkte)

- (a) Wie erstellt man in NumPy ein Array mit Werten von 0 bis 9 (inklusive) und Schrittweite 1? (1 Punkt)
- (b) Welche Informationen liefern `.shape` und `.dtype?` (1 Punkt)
- (c) Geben Sie ein Beispiel für die Verwendung von `np.linspace`. (1 Punkt)

2. Broadcasting und Operationen (4 Punkte)

- (a) Warum ist Broadcasting ein Vorteil in NumPy? (1 Punkt)
- (b) Gegeben:

```
import numpy as np  
A = np.array([[1, 2, 3],  
             [4, 5, 6]])  
B = np.array([10, 20, 30])
```

Was ist das Ergebnis von `A + B`? Notieren Sie die entstehende Matrix. (2 Punkte)

- (c) Nennen Sie zwei Aggregationsfunktionen und deren Zweck. (1 Punkt)

3. Form und Matrixmanipulation (3 Punkte)

- (a) Geben Sie ein Beispiel, wie man ein 1D-Array der Länge 6 in ein 2D-Array der Form (2,3) umwandelt. (1 Punkt)
- (b) Wie transponiert man eine Matrix `M` in NumPy? (1 Punkt)

- (c) Vervollständigen Sie den Code, der das Matrixprodukt von **A** und **C** berechnet:

```
A = np.array([[1,2],[3,4]])
C = np.array([[5],[6]]) # 2x1 Matrix

# TODO: Matrixprodukt
result = ?
```

(1 Punkt)

5. Pandas (12 Punkte)

1. Grundlagen (4 Punkte)

- Was ist der Unterschied zwischen **Series** und **DataFrame**? (1 Punkt)
- Wie liest man eine CSV-Datei **data.csv** in ein DataFrame **df** ein? (1 Punkt)
- Nennen Sie zwei wichtige Attribute eines DataFrame und erklären Sie diese. (2 Punkte)

2. Selektion und Datenbereinigung (4 Punkte)

- Wie finden Sie fehlende Werte in einer bestimmten Spalte **mycol**? Schreiben Sie einen beispielhaften Pandas-Code. (1 Punkt)
- Was ist der Unterschied zwischen **df.loc** und **df.iloc**? (1 Punkt)
- Betrachten Sie den Code:

```
df_new = df[df['column'] > 100]
df_new.sort_values(by='column', ascending=False)
```

Welche Zeilen werden in **df_new** enthalten sein und wie ist deren Sortierreihenfolge? (2 Punkte)

3. Gruppierung, Pivot-Tabellen und Plotten (4 Punkte)

- Wie berechnen Sie den mittleren Wert einer Spalte **Value** für jede Kategorie in **Category** mithilfe von **groupby**? (1 Punkt)
- Erstellen Sie eine Pivot-Tabelle, die den Durchschnitt der Spalte **Sales** anzeigt, gruppiert nach **Store** (Zeilen) und **Product** (Spalten). Geben Sie ein kurzes Code-Beispiel. (1 Punkt)
- Welche Schritte sind nötig, um eine Zeitspalte in das Datetime-Format zu konvertieren und die Daten anschließend wöchentlich zu resampeln? (1 Punkt)
- Schreiben Sie einen Einzeler-Code, der ein Liniendiagramm der Spalte **Revenue** zeichnet. (1 Punkt)

Punkteschlüssel:

- Teil 1: 7 Punkte
- Teil 2: 12 Punkte

- Teil 3: 10 Punkte
- Teil 4: 10 Punkte
- Teil 5: 12 Punkte

(Gesamt: 51 Punkte)

Viel Erfolg bei der Klausur!