

Klausur: Python, OOP, Algorithmik, NumPy und Pandas

Keine Hilfsmittel erlaubt

12. Februar 2025

Anleitung: Bitte beantworten Sie alle Fragen in deutscher Sprache.

Bearbeitungszeit: 90 Minuten. Keine Skripte, kein Internet, keine Hilfsmittel.

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-Frage: Fehler finden (2 Punkte)

Betrachten Sie folgenden Python-Code:

```
1 my_var = "Hello"
2
3 if my_var == "Hello":
4     print("Yes, it's Hello")
5 else:
6     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 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? Geben Sie ein kurzes Beispiel in Python.
- Erklären Sie kurz *Encapsulation*, *Inheritance* und *Polymorphism*.

2. Vererbung (4 Punkte)

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

3. Code-Frage: Fehler erkennen (4 Punkte)

Betrachten Sie folgenden Code-Ausschnitt:

```
1 class Animal:
2     def __init__(self, name):
3         self.__name = name
4
5     def speak(self):
6         print("... ")
7
8 class Dog(Animal):
9     def __init__(self, name, breed):
10        super().__init__(name)
11        self.breed = breed
12
13    def speak():
14        print(f"{self.__name} barks!")
15
16 my_dog = Dog("Rex", "Labrador")
17 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 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?
- Warum kann Binary Search nicht auf unsortierten Daten angewendet werden?
- Geben Sie ein Beispiel für einen Anwendungsfall von Linear Search.

2. Sortieralgorithmen (3 Punkte)

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

3. Rekursion (4 Punkte)

Unten sehen Sie eine unvollständige rekursive Funktion `factorial`:

```

1 def factorial(n):
2     # TODO: Basisfall ergänzen
3     if n < 0:
4         return None
5     else:
6         # 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?
- (b) Welche Informationen liefern `.shape` und `.dtype`?
- (c) Geben Sie ein Beispiel für die Verwendung von `np.linspace`.

2. Broadcasting und Operationen (4 Punkte)

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

```

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

```

Was ist das Ergebnis von `A + B`? Notieren Sie die resultierende 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.
- (b) Wie transponiert man eine Matrix `M` in NumPy?
- (c) Vervollständigen Sie den Code, der das Matrixprodukt von `A` und `C` berechnet:

```

1 A = np.array([[1,2],[3,4]])
2 C = np.array([[5],[6]])    # 2x1 Matrix
3
4 # TODO: Matrixprodukt
5 result = ?

```

5. Pandas (12 Punkte)

1. Grundlagen (4 Punkte)

- (a) Was ist der Unterschied zwischen `Series` und `DataFrame` in Pandas?
- (b) Wie liest man eine CSV-Datei `data.csv` in ein DataFrame ein?
- (c) Nennen Sie zwei wichtige Attribute eines DataFrame und erklären Sie diese kurz.
- (d) Wie speichert man ein DataFrame namens `df` in eine neue CSV-Datei namens `output.csv` (ohne Index)?

2. Selektion und Datenbereinigung (4 Punkte)

- (a) Wie finden Sie fehlende Werte in einer Spalte `col_x`? Schreiben Sie einen beispielhaften Pandas-Code.
- (b) Was ist der Unterschied zwischen `df.loc` und `df.iloc`?
- (c) Betrachten Sie den Code:

```

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

```

Welche Zeilen werden in `df_new` enthalten sein, und wie ist die Sortierreihenfolge?

- (d) Wie fügen Sie einer existierenden DataFrame-Spalte `df['Price']` z.B. 3 hinzu (um alle Preise zu erhöhen)? Geben Sie ein Code-Beispiel.

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

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

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!