HWS 2024
Moritz Grävinghoff

STADS | Students' Association for Data Analytics & Statistics

# Probe-Zertifikatsklausur

# "Programmierkurs Python"

---

**Wichtige Hinweise:**

- Für die Bearbeitung der Klausur hast Du **45 Minuten** Zeit. Zusätzlich hast Du weitere **10 Minuten** vor Beginn der Klausur zum Download der Aufgaben und im Anschluss an die Klausur **10 Minuten** zum Upload der Lösungen. Wir werden die Klausuraufgaben um 18:50 an deine E-Mail Adresse zusenden. Bitte beachte:

  **Die Abgabe muss bis spätestens 19.55 Uhr an `pythonkurs@stads.de` erfolgen.**

- Die Klausur besteht aus 3 Aufgaben.

- Bitte erstelle für **alle Aufgaben insgesamt eine Python-Datei** mit dem Namen

  `<Nachname>_<Vorname>_exam.py` (z.B. `mustermann_max_exam.py`)

  und bearbeite ausschließlich in dieser Datei die Prüfung. Sende deine Lösungen anschließend als **ZIP-Datei** mit dem Namen

  `<Nachname>_<Vorname>_exam.zip` (z.B. `mustermann_max_exam.zip`)

  an die oben angegebene E-Mail Adresse.

- Der Code muss in Python $\geq$3.11.5 mit pandas $\geq$1.5.3, matplotlib $\geq$3.7.1 bzw. plotly $\geq$5.15.0 lauffähig sein. Falls weitere Pakete oder andere Versionen verwendet werden, muss die jeweilige Version angegeben werden.

**Viel Erfolg!**

---

**Aufgabe 1** (15 Punkte)

a) [2 Pkt] Erstellen Sie eine Variable namens `lieblingsfarbe` und weisen Sie der Variable Ihre Lieblingsfarbe zu. Drucken Sie den Text "Meine Lieblingsfarbe ist `lieblingsfarbe`".

b) [6 Pkt] Erstellen Sie ein Dictionary namens `resultat_b`, in welchem Sie für jeden Schlüssel "i" bis "vi" ein Tupel als Wert angeben. Jedes Tupel soll als erstes Element den Datentyp des Ausdrucks und als zweites Element den Wert des Ausdrucks enthalten. Drucken Sie das Dictionary.

(i) `7 % 4`

(ii) `"Python"[1:5]`

(iii) `bool(None)`

(iv) `{2, 4, 6, 8} - {4, 8}`

(v) `[z**3 for z in range(5)]`

(vi) `len("Welt") < 5`

c) [3 Pkt] Erstellen Sie eine Liste namens `fibonaccizahlen` mit den ersten 10 Fibonacci-Zahlen. Drucken Sie das vierte und vorletzte Element der Liste.

d) [4 Pkt] Sie haben folgenden Code gegeben, welcher eine Liste von 300 Zufallszahlen im Bereich von 10 bis 500 generiert.

```
import random
random.seed(42)
zufallszahlen = [random.randint(100, 500) for _ in range(300)]
```

Drucken Sie die Anzahl der Zahlen aus `zufallszahlen`, die durch 9 teilbar sind.

**Aufgabe 2** (15 Punkte)

a) [7 Pkt] Schreiben Sie eine Funktion namens `carbon_footprint_category`, die den jährlichen CO2-Ausstoß (in Tonnen) von mehreren Personen in einer Liste entgegennimmt und eine Einschätzung der Kategorie des CO2-Fußabdrucks zurückgibt.
Kategorien:

- "Niedrig": CO2-Ausstoß $< 5$ Tonnen/Jahr
- "Durchschnittlich": 5 Tonnen/Jahr $\leq$ CO2-Ausstoß $\leq 15$ Tonnen/Jahr
- "Hoch": CO2-Ausstoß $> 15$ Tonnen/Jahr

Die Rückgabe soll ein Dictionary sein, wobei der CO2-Wert der Schlüssel und die Kategorie der Wert ist.
Rufen Sie die Funktion `carbon_footprint_category` mit der Liste `[3, 20, 12, 16, 7]` auf und drucken Sie das Ergebnis.

b) [8 Pkt] Erstellen Sie

- eine Klasse namens `Buchsammlung`, die keine Attribute hat,
- einen leeren Konstruktor, der nur `self` initialisiert,
- eine Methode namens `buch_hinzufuegen`, die zwei Strings entgegennimmt (Titel des Buches und Autor) und eine Zeichenkette mit dem Format "Titel von Autor" zurückgibt,
- eine Methode namens `sammlung_auflisten`, die eine Liste von Tupeln (Titel, Autor) entgegennimmt und eine formatierte Zeichenkette mit allen Buchtiteln in der Sammlung ausgibt. Die Methode soll intern die Methode `buch_hinzufuegen` verwenden.

Rufen Sie die Methode `sammlung_auflisten` mit der Liste

```
[("Der Hobbit", "J.R.R. Tolkien"),
("1984", "George Orwell"),
("Die Verwandlung", "Franz Kafka")]
```

auf und drucken Sie das Ergebnis.

**Aufgabe 3** (15 Punkte)

Sie haben folgenden Code gegeben, der den zu bearbeitenden Datensatz importiert.

```
import pandas as pd
import numpy as np
np.random.seed(42)
df = pd.DataFrame({
    'sensor_id': range(1, 11),
    'messwert': np.random.uniform(50, 100, 10),
    'kalibrierung': np.random.uniform(0.8, 1.2, 10),
    'fehlerquote': np.random.uniform(0.01, 0.1, 10),
    'temperatur': np.random.uniform(20, 30, 10)
})
```

Das DataFrame enthält fünf Spalten:

- sensor_id $\mapsto$ Eindeutige ID des Sensors
- messwert $\mapsto$ Gemessener Wert des Sensors
- kalibrierung $\mapsto$ Kalibrierungsfaktor des Sensors
- fehlerquote $\mapsto$ Fehlerquote des Sensors in Prozent
- temperatur $\mapsto$ Temperatur während der Messung

a) [3 Pkt] Geben Sie den durchschnittlichen Messwert sowie die höchste Fehlerquote aus.

b) [4 Pkt] Erstellen Sie eine neue Spalte namens `justierter_wert`, die den justierten Messwert berechnet (Messwert $*$ Kalibrierung). Geben Sie die letzten fünf Zeilen des DataFrames aus.

c) [3 Pkt] Geben Sie alle Sensoren aus, deren Kalibrierungsfaktor größer als 1.0 ist.

d) [5 Pkt] Erstellen Sie einen Scatter Plot, der die Beziehung zwischen justiertem Messwert, Fehlerquote und Temperatur darstellt. Verwenden Sie den justierten Messwert als x-Achse, die Fehlerquote als y-Achse und die Temperatur zur Farbgebung der Punkte. Der Titel des Plots soll "Beziehung zwischen justiertem Messwert, Fehlerquote und Temperatur" sein. Benennen Sie die x-Achse mit "Justierter Messwert" und die y-Achse mit "Fehlerquote (%)". Zeigen Sie den Plot.