

Programmieren – Praktikum 4

Ingenieurinformatik Teil 1, Wintersemester 2026/27

David Straub

Protokolldaten

Nach dem simulierten Pack aus Praktikum 3 bekommt Ihr Labor heute echte Textprotokolle statt einzelner Werte – Zeilen aus einer Log-Datei, die Sie zerlegen und auswerten:

```
protokoll = """01;3.812;22.4
02;3.798;23.1
03;2.951;31.5
04;3.833;22.6
05;3.109;23.0
06;3.041;24.2
07;3.772;30.9
08;3.839;22.5"""
```

Jede Zeile: Zellnummer;Spannung in V;Temperatur in °C – getrennt durch Semikolon.

Aufgabe 1: Eine Zeile zerlegen

1. Nehmen Sie die erste Zeile als eigenen String: `zeile = "01;3.812;22.4"`
2. Zerlegen Sie sie mit `split(";")` in drei Teile
3. Wandeln Sie Spannung und Temperatur in `float` um und geben Sie beide aus (`print`)

Aufgabe 2: Das ganze Protokoll

1. Zerlegen Sie `protokoll` mit `split("\n")` in einzelne Zeilen
2. Durchlaufen Sie alle Zeilen mit einer Schleife und zerlegen Sie jede wie in Aufgabe 1
3. Geben Sie für jede Zeile Zellnummer, Spannung und Temperatur formatiert aus (`print`, Feldbreite)

Aufgabe 3: Auffällige Zellen zählen

1. Zählen Sie mit dem Akkumulator-Muster, wie viele Zeilen eine Spannung unter 3,2 V haben
2. Zählen Sie separat, wie viele Zeilen eine Temperatur über 30 °C haben
3. Geben Sie beide Anzahlen aus (`print`)

Aufgabe 4: Zeichen zählen

Verschachtelte Schleifen: die äußere über die Zeilen, die innere über die Zeichen jeder Zeile.

1. Zählen Sie, wie oft die Ziffer `3` im gesamten Protokolltext vorkommt
2. Zum Gegenchecken gibt es eine eingebaute Methode: `protokoll.count("3")` – stimmt Ihr Ergebnis überein?

Aufgabe 5 (falls Sie Funktionen 2 schon hatten): Die Zeile als Funktion

1. Schreiben Sie eine Funktion `zeile_zerlegen(zeile)`, die Zellnummer, Spannung und Temperatur als drei Werte zurückgibt (`return`) – nutzen Sie Ihre Lösung aus Aufgabe 1
2. Bauen Sie Ihre Protokoll-Auswertung aus Aufgabe 2 so um, dass sie diese Funktion verwendet
3. Was ist an dieser Version besser lesbar – was ist gleich geblieben?

Zusatz für Schnelle

- Seriennummern haben bei Ihrem Hersteller das Format `LB240815-06-A` : Hersteller (2 Zeichen), Jahr (2), Monat (2), Tag (2), dann `-`, Zellnummer (2), `-`, Charge (1). Extrahieren Sie alle sechs Felder per Slicing – die Trennstriche stehen an festen Positionen
- Schreiben Sie eine Funktion, die aus dem Protokoll die **ganze Zeile** mit der niedrigsten Spannung zurückgibt (`return`) – nicht nur die Zahl
- Testen Sie `"3.812".isdigit()` – das Ergebnis überrascht. Erklären Sie es, und überlegen Sie, wie man stattdessen prüfen könnte, ob ein String eine Kommazahl darstellt