

IT-Berufe

Modul Systementwicklung - Lernauftrag 06

6. Dateiformate und Dateiverarbeitung

Ausgangssituation

Ihr Tool soll in der Lage sein, Dateien in verschiedenen Formaten zu speichern und zu verarbeiten. Dies ermöglicht es dem Benutzer, Daten wie Aufgabenlisten oder Dateiinformationen in einer strukturierten Form zu exportieren und später wieder zu laden. Zusätzlich soll das Tool doppelte Dateien erkennen und bei Bedarf löschen können.

Ziel ist es eine Funktion zu implementieren, die doppelte Dateien identifiziert und löscht.

Arbeitsauftrag

- 1) **Grundlagen von Dateien und Ordnern** - Machen Sie sich mit dem Modul *pathLib* vertraut. Recherchieren Sie, wie Ordner und Dateien mit diesem Modul erstellt, bearbeitet und überprüft werden können.
- 2) **Dateien erstellen und schreiben** - Entwickeln Sie eine Funktion, die eine neue Datei erstellt und mit einem vom Benutzer angegebenen Text füllt. Die Datei soll dabei den Namen *daten.txt* haben. Stellen Sie sicher, dass vorhandene Dateien überschrieben werden können.
- 3) **Dateien öffnen und lesen** - Schreiben Sie eine Funktion, die die zuvor erstellte Datei öffnet und deren Inhalt in der Konsole ausgibt. Fügen Sie eine Fehlerprüfung hinzu, die sicherstellt, dass die Datei existiert, bevor sie gelesen wird.
- 4) **Hashes erstellen** - Informieren Sie sich über den Sinn und Zweck von Hash-Funktionen. Entwickeln Sie eine Funktion, die den SHA256-Hash einer Datei berechnet und das Ergebnis in der Konsole anzeigt. Verwenden Sie dazu die Bibliothek *hashLib*.
- 5) **Dateien mit gleichen Inhalten identifizieren** - Schreiben Sie eine Funktion, die überprüft, ob in einem Ordner Dateien mit identischen Inhalten existieren. Vergleichen Sie dazu die SHA256-Hashes aller Dateien im Ordner und listen Sie Dateien mit gleichen Hashes auf.

Arbeitsmittel	Internet
Zeit in Stunden	10 h
Sozialform	Kollektive Einzelarbeit
Ergebnissicherung	Besprechung im Plenum

Anlagen – LA 6

Pseudocode

```
FUNKTION finde_duplikate(ordner_pfad):  
    WENN ordner_pfad nicht existiert ODER kein Verzeichnis ist:  
        GIB ein leeres Wörterbuch zurück  
  
    INITIALISIERE hash_verzeichnis als leeres Wörterbuch  
  
    FÜR jede Datei im ordner_pfad:  
        BERECHNE datei_pfad als vollständigen Pfad der Datei  
        BERECHNE datei_hash mit berechne_datei_hash(datei_pfad)  
  
        WENN datei_hash bereits in hash_verzeichnis existiert:  
            FÜGE datei_pfad zu hash_verzeichnis[datei_hash] hinzu  
        SONST:  
            SETZE hash_verzeichnis[datei_hash] auf eine neue Liste mit datei_pfad  
  
    FILTERE hash_verzeichnis, um nur Einträge mit mehr als einem Pfad zu behalten  
    GIB das gefilterte hash_verzeichnis zurück  
  
FUNKTION berechne_datei_hash(datei_pfad, blockgröße=4096):  
    INITIALISIERE hash_objekt mit SHA256  
  
    ÖFFNE datei_pfad im binären Lese-Modus:  
        SOLANGE ein block = Lese blockgröße Bytes aus der Datei:  
            AKTUALISIERE hash_objekt mit block-Daten  
  
    GIB den Hexadezimalwert von hash_objekt zurück
```