

Übung: Programmierung – Datenstrukturen, Einlesen von Dateien

Programmierung

Verwaltung eines Dateisystems mit Laufwerken und Ordnern

In modernen Dateisystemen gibt es eine hierarchische Struktur aus Laufwerken und Ordnern. Jedes Laufwerk besitzt eine Wurzel (Root-Ordner), unter der sich weitere Ordner befinden. Jeder Ordner kann wiederum Unterordner enthalten. Dateien sollen in dieser vereinfachten Modellierung als Speichergrößen innerhalb von Ordnern abgebildet werden.

Anforderungen:

1. Klasse Ordner

- Ein Ordner besitzt:
 - einen Namen (name)
 - eine Speichergröße (groesse, Standardwert: 0 MB)
 - eine Liste von Unterordnern
- Ein Ordner kann Unterordner enthalten.

2. Klasse Laufwerk

- Ein Laufwerk besitzt:
 - einen Namen (name)
 - eine Root-Struktur mit dem Hauptordner Root
- Folgende Methoden sollen in der Klasse Laufwerk implementiert werden:
 - `struktur_anzeigen()`: Gibt die gesamte Ordnerstruktur des Laufwerks aus.
 - `gesamtgröße_berechnen()`: Berechnet die gesamte belegte Größe des Laufwerks.
 - `ordner_suchen(name)`: Sucht einen Ordner anhand seines Namens innerhalb des Laufwerks und gibt ihn zurück.
 - `ordner_hinzufuegen(parent_name, neuer_ordner)`: Fügt einen neuen Ordner in einen bestehenden Ordner ein.

- `zip_ordner(name)`: Komprimiert einen bestimmten Ordner und reduziert dessen Speichergröße um die Hälfte.

Beispielhafte Nutzung

- Ein Laufwerk C: wird erstellt.
- Die Ordnerstruktur wird aufgebaut:
 - Dokumente (100 MB)
 - Musik (200 MB) mit Unterordner Lieder (150 MB)
 - Backup (50 MB) innerhalb von Dokumente
- Die gesamte Laufwerksgröße wird berechnet und ausgegeben.
- Der Ordner Musik wird komprimiert (Größe halbiert).
- Die neue Struktur sowie die geänderte Gesamtgröße werden angezeigt.

Erwartetes Verhalten

- Die Methoden innerhalb der Klasse Laufwerk übernehmen die Suche nach Ordnern und fügen neue Ordner an der richtigen Stelle ein.
- Die Methode `zip_ordner(name)` findet den entsprechenden Ordner und reduziert seine Größe sowie die seiner Unterordner.
- Die Methode `gesamtgröße_berechnen()` liefert die Gesamtgröße des Laufwerks korrekt zurück.
- Die Methode `struktur_anzeigen()` gibt die Struktur in einer lesbaren hierarchischen Form aus.

Aufgabe

Entwickle eine Python-Klassenstruktur, die es ermöglicht, ein Laufwerk mit Ordnern zu verwalten. Die Methoden zur Verwaltung und Suche nach Ordnern sollen innerhalb der Klasse Laufwerk realisiert werden, nicht direkt in der Klasse Ordner.

Einlesen von Dateien

Gegeben ist der Inhalt eines Ordners als Dateinamen und der Größe der Datei. Erstelle eine .csv Datei mit dem folgenden Inhalt:

Dateiname	Größe in MB
Foto1	200
Foto3	120
Video1	2000
Textdokument1	8
PDF1	27
ordner1.zip	254

Ein Programm zur Auswertung des Speicherplatzes soll folgendes können:

- Die Datei soll eingelesen werden können.
- Die Daten in einer passenden Struktur abgespeichert werden.
- Die Funktion anzeigen() soll alle Dateinamen und die Größe der Dateien anzeigen.
- Die Funktion ordner_ansicht() soll alle Dateinamen anzeigen.
- Die Funktion max_groesse(name) soll den Dateinamen und die Größe der größten Datei ausgeben.
- Die Funktion gesamtgroesse() soll die gesamte Größe des Ordners anzeigen.
- Die Funktion abweichung() soll die Differenz des Speicherplatzes zum Speicherplatzbedarf der größten Datei im Ordner ausgeben (Foto1: 1800 MB kleiner, ...)
- Die Funktion ordner_zip() soll alle Dateien mit einer Größe von 50% der aktuellen Größe in eine neue Datei "zip_ordner.csv" speichern.

Implementiere eine Lösung für diese Aufgabenstellung.