

Übung 7

Hashtabelle

Themen: Klassen, geschachtelte Listen, hash, for...else, itertools.chain, __str__, __getitem__, __setitem__

Python-Dictionaries sind als Hashtabellen implementiert.

Schreiben Sie eine Klasse `HashTabelle`, welche eine Hashtabelle implementiert. Die Daten der Hashtabelle speichern Sie in einer Liste von Buckets. Ein Bucket ist eine Liste von Key-Value-Paaren.

Die `init`-Methode der Klasse erhält optional die Größe der Hashtabelle mit einem Defaultwert von 10 als Argument und initialisiert die Hashtabelle mit leeren Buckets.

Die Methode `__getitem__` erhält einen Key als Argument ruft die Funktion `hash` für diesen Key auf und teilt den Hashwert per Integer-Division (`//`) durch die Größe der Hashtabelle. Wenn das Divisionsergebnis 13 ist, wird anschließend in Bucket 13 nach dem übergebenen Key gesucht. Wenn er gefunden wird, wird der entsprechende Value zurückgegeben, andernfalls `None`.

Analog wird eine Methode `__setitem__` definiert, welche einen Key und einen Value als Argument erhält, in der Tabelle nach dem Key sucht und seinen Wert auf Value setzt. Wenn der Key nicht gefunden wurde, wird er zu dem Bucket hinzugefügt.

Die Methode `__str__` gibt die Hashtabelle in dem Format aus, in dem Python den Inhalt von Dictionaries ausgibt.

Eine Methode `resize` wird aufgerufen, wenn die Zahl der Elemente in der Hashtabelle die Größe der Hashtabelle übersteigt. `resize` verdoppelt die Größe der Hashtabelle und speichert dann alle Key-Value-Paare in der neuen Tabelle.