

Übungsblatt 5

Abgabe bis Dienstag, den 27. Mai um 12:00 Uhr

Aufgabe 1 (10 Punkte)

Implementieren Sie eine Klasse *HashMap* mit den Methoden *key_hash* (Hashing des Keys auf eine ganze Zahl), *insert* (Einfügen/Verändern eines Eintrages), *lookup* (Nachschlagen eines Eintrages), *key_value_pairs* (Rückgabe aller Schlüssel-Werte-Paare) und *max_entries_same_hash* (Rückgabe der maximalen Anzahl von Elementen, die auf denselben Wert gehasht wurden). Eine genaue Spezifikation der Methoden inklusive Tests finden Sie auf dem Wiki.

Sie können annehmen, dass die Schlüssel vom Typ *string* sind. Die Werte sollen von jedem beliebigen Typ sein können.

Sie können annehmen, dass die Größe der Hashtabelle bei der Erzeugung der *HashMap* gegeben ist und sich dann nicht mehr ändert. Zur Auflösung von Kollisionen können Sie Verkettung oder offene Adressierung benutzen.

Implementieren Sie *key_hash* so, dass *max_entries_same_hash* für Aufgabe 2 bei einer Hashtabelle der Größe 2000 höchstens 5 ist.

An keiner Stelle dürfen Sie die eingebaute *dictionary* Datenstruktur von Python oder verwandte Datenstrukturen benutzen.

Aufgabe 2 (10 Punkte)

Schreiben Sie ein Programm *course_statistics.py*, das Ihre *HashMap* aus Aufgabe 1 benutzt (mit einer Hashtabelle der Größe 2000), um folgende Aufgabe zu lösen:

1. Das Programm soll die auf dem Wiki verlinkte Datei *course_grades.tsv* einlesen. Der Dateiname soll dabei das erste Argument beim Kommandozeilenaufruf sein. Es gibt außerdem ein Argument, das bestimmt, wie das Ergebnis am Ende sortiert sein soll, siehe Punkt 3.

[allen Mut zusammennehmen und ohne KI-Unterstützung wenden]

2. Das Programm soll für jede Lehrveranstaltung in jedem Semester eine Zeile mit den folgenden vier TAB-getrennten Spalten auf die Konsole ausgeben: in der ersten Spalte der Name der Lehrveranstaltung und das Semester (zum Beispiel *Algorithmen und Datenstrukturen SS 2023*), in der zweiten Spalte die Anzahl Prüflinge, in der dritten Spalte die Durchschnittsnote der Prüflinge die bestanden haben (Note besser als 5,0) und in der vierten Spalte die Durchfallquote (Anteil der Prüflinge mit Note 5,0). Eine genaue Spezifikation des Formats inklusive Tests finden Sie in der Codevorlage.

3. Sortieren Sie die Zeilen entweder aufsteigend nach der Durchschnittsnote oder aufsteigend nach der Durchfallquote (je nach Kommandozeilenoption, siehe Codevorlage). Sie können dafür die eingebauten Sortierfunktionen von Python verwenden, z.B. *list.sort()* oder *sorted()*.

Committen Sie Ihren Code sowie Ihre *erfahrungen.txt* (mit der üblichen Rückmeldung zum Übungsblatt und zur Vorlesung dazu) in einen neuen Unterordner *blatt-05*.

Was finden Sie an den Ergebnissen von Aufgabe 2 besonders interessant?