

Übungsblatt 11

Abgabe: 26.1.2008

Datenverwaltung

Wenn es gilt, eine grosse Anzahl an Datensätzen zu verwalten, haben die bisher vorgestellten Organisationsstrukturen Vor- und Nachteile. Nehmen wir beispielsweise die Kundenverwaltung der Firma *Raff & Zahn Ltd, Cayman Isl.*, die kostenpflichtige Repetitorien zur praktischen Informatik anbieten will.

Hier werden als Daten von (potentiellen) Kunden folgende Informationen vorgehalten: Der Name in der Form **Nachname**, **Vorname**, die Kundennummer, der Name der Universität, die e-mail Adresse, und eine Referenz auf den eigentlichen Datensatz, in dem die bisherigen Geschäftsbeziehungen dokumentiert werden.

Ihr sollt nun die Verwaltung dieser Daten implementieren, damit *Raff & Zahn* Datensätze aufnehmen kann, bei Vorlage von **Nachname**, **Vorname** auf die entsprechenden Datensätze zugreifen kann und den kompletten Datenbestand durchlaufen kann – etwa um die e-mail Adressen rauszusammeln und zu verkaufen. Es gelingt Euch sogar, den Auftrag gegen einen geringen Aufpreis um das Löschen von Datensätzen zu erweitern.

Die häufigsten Tätigkeiten der Firma betreffen die Bearbeitung einzelner Datensätze – diese müssen also schnell zugreifbar sein. Hier soll ein *String-Hashing* verwendet werden, mit dem in einem Array von Listen von Kundendatenverweisen die zu durchsuchende Liste schnell gefunden wird. Daneben sollen die Daten alphabetisch sortiert vorgehalten werden, um z.B. regelmässig den Kunden „wichtige Informationen!“ zukommen zu lassen.

Die Aufgabe

Es soll sowohl eine Listenstruktur für die sortierte Ausgabe der Daten als auch ein Hash-Array von Listen implementiert werden. Dabei sollen die Kundendaten **nicht** doppelt im Speicher gehalten werden, sondern nur die *Referenzen* auf die Einträge sollen in den zwei erwähnten Ordnungen organisiert werden! Benutzt eine geeignete Hash-Funktion, für die weitere Suche dürft Ihr annehmen, dass **Nachname**, **Vorname** jeweils eindeutig sind.

Dabei soll bei der Wahl der Datenstruktur zusätzlich berücksichtigt werden, dass das Löschen eines Eintrags möglichst effizient implementiert wird.

Insgesamt sollen die folgenden vier Methoden mit geeignet zu definierenden Signaturen implementiert werden:

reingefallen fügt einen neuen Kunden in die Hash-Struktur sowie in die alphabetisch sortierte Liste ein.

gerichtlichAngeordnet löscht einen Eintrag zu einem gegebenen Namen sowohl aus der Hash – Struktur als auch aus der Liste.

abzocken sucht zu einem gegebenen Namen über die Hash-Liste die dazugehörige Daten heraus und liefert diese als Rückgabewert.

`spammen` gibt zeilenweise (nach Nachname sortiert) Vorname Nachname und e-mail Adresse auf den Bildschirm aus.

Zeigt mit geeigneten Testfällen, dass Eure Methoden die gewünschte Funktionalität besitzen.