Protokoll Aufgabe\_1  
  
Das Programm implementiert eine Hashtabelle, um Informationen über Aktien zu verwalten. Die Hashtabelle verfügt über zwei Arrays von Zeigern auf Aktien, eines für den Namen der Aktie und eines für ihre Abkürzung.

Die Hashfunktion ordnet einen Zeichenketten-Schlüssel (Name oder Kürzelname) einem Integer-Index in der Tabelle zu.

Bei einer Kollision verwendet das Programm quadratische Probing, um einen freien Slot in der Tabelle zu finden.

Der Aufwand für das Einfügen, Suchen und Löschen von Datensätzen hängt von der Füllstufe der Hashtabelle und der Anzahl der Kollisionen ab, aber in optimalen Fällen ist konstant, O(1), im Vergleich zu O(n) bei einem Array und O(n/2) bei einer einfach verketteten Liste.

Einfügen: im schlimmsten Fall sind alle Slots in der Tabelle belegt und das Programm muss eine lineare Suche durchführen, um einen freien Slot zu finden. Wenn die Hashtabelle nicht vollständig belegt ist und es nicht viele Kollisionen gibt, O(1).

Suchen: hängt ebenfalls von der Füllstufe der Hashtabelle und der Anzahl der Kollisionen ab. Im schlimmsten Fall muss das Programm eine lineare Suche durch alle Slots in der Tabelle durchführen, was zu einem Aufwand von O(n) führt, wobei n die Anzahl der Datensätze in der Hashtabelle ist. Im besten Fall wird der Datensatz im ersten Slot gefunden, was zu einem Aufwand von O(1) führt.

Der Aufwand, einen Datensatz in der Hashtabelle zu löschen, ist konstant, O(1). Das Programm setzt einfach den Zeiger in der Tabelle auf NULL.

Für 1000 Aktien wäre der Aufwand, der für das Einfügen, Suchen und Löschen von Datensätzen erforderlich ist, konstant, O(1), wenn die Hashtabelle nicht bis zur Kapazität gefüllt ist und es nicht viele Kollisionen gibt.  
  
  
  
Faruk Imamovic if21b044  
  
Vadim Steshkov if22b262