

## ÜBUNG 1 - HASHTABELLE

Arbeiten Sie in 2er Gruppen an folgendem Programmierbeispiel. Die Wahl der Programmiersprache (C, C++, C#, Java, Python) bleibt Ihnen überlassen, jedoch muss die Datenstruktur selbst implementiert werden (keine fertige Hashtable Klasse verwenden!). Weitere Programmiersprachen nur nach Absprache!

### AUFGABENSTELLUNG

Implementieren Sie ein Programm zur Verwaltung von Aktienkursen. Die Daten sollen dabei mittels Hashtabellen gespeichert werden.

Um schnell nach einer bestimmten Aktie suchen zu können, sollen die Aktien in einer Hashtabelle verwaltet werden. Pro Aktie sollen Name, Wertpapierkennnummer (WKN als String) und Kürzel (auch ein beliebiger String) gespeichert werden. Weiters sind pro Aktie die Kursdaten der vergangenen 30 Tage zu verwalten mit folgenden Informationen:  
(Date,Close,Volume,Open,High,Low)

Aktienkursdaten dieser Form können z.B.

<https://www.nasdaq.com/market-activity/quotes/historical>  
als csv Datei heruntergeladen werden.

Das Programm wird über folgende Menüpunkte gesteuert:

1. ADD: Eine Aktie mit Namen, WKN und Kürzel wird hinzugefügt.
2. DEL: Aktie wird gelöscht.
3. IMPORT: Kurswerte für eine Aktie werden aus einer csv Datei importiert
4. SEARCH: Eine Aktie wird in der Hashtabelle gesucht und der aktuellste Kurseintrag (Date,Close,Volume,Open,High,Low) wird ausgegeben. Man soll sowohl nach Name als auch nach Kürzel gesucht werden können.
5. PLOT: Die Schlusskurse der letzten 30 Tage einer Aktie werden als ASCII Grafik ausgegeben, Format ist frei wählbar.
6. SAVE <filename>: Programm speichert die Hashtabelle in eine Datei ab
7. LOAD <filename>: Programm lädt die Hashtabelle aus einer Datei
8. QUIT: Programm wird beendet

**Hinweise Datenstruktur**

Überlegen Sie sich eine geeignete Hashfunktion, die aus dem Namen der Aktie bzw. dem Kürzel einen Hashwert generiert (Kriterien für gute Hashfunktionen siehe Vorlesung).

Implementieren Sie als Kollisionsbehandlung für die Hashtabelle quadratische Sondierung. Überlegen Sie sich außerdem ein effizientes Verfahren für das Löschen aus der Hashtabelle!

Überlegen Sie sich weiters eine geeignete Größe für die Hashtabelle unter der Annahme, dass maximal 1000 Aktien verwaltet werden sollen. Definieren Sie auch eine geeignete Datenstruktur für die Speicherung der 30 Kursdaten pro Aktie, um diese effizient anzeigen zu können.

**Hinweise Fileformat**

Für die Kursdaten ist das csv Fileformat von [finance.yahoo.com](http://finance.yahoo.com) zu verwenden, Beispiel siehe Moodle Kurs.

Überlegen Sie sich ein geeignetes Dateiformat, um die Daten der Hashtabelle abzuspeichern. Recherchieren Sie im Internet den Begriff Serialisierung in diesem Zusammenhang.

**Aufwandsabschätzung**

Vergleichen Sie den Aufwand zum Einfügen, Suchen und Löschen eines Datensatzes in ihrer Hashtabelle mit dem Aufwand zum Einfügen, Suchen und Löschen in einem normalen Array und in einer einfach verketteten Liste. Beachten Sie auch den Füllgrad der Hashtabelle. Wie viele Operationen sind jeweils bei 1000 Aktien notwendig und wie ist der Aufwand allgemein nach O-Notation definiert?

**Abgabe**

Im Abgabesystem ist ein .zip oder .tgz File abzugeben. Dieses soll beinhalten:

- Alle Sourcen inkl. Code Kommentaren!
- ausführbares Programm
- Protokoll mit Beschreibung der Datenstruktur (Hashfunktion, Kollisionserkennung, Verwaltung der Kursdaten, Löschalgorithmus und Aufwandsabschätzung)

Abgabe und Deadline siehe Moodle.

Die Abgabe muss beim ersten Code Review präsentiert werden, es können für diese Übung maximal 18 Punkte erreicht werden (15 Punkte für das Programm und 3 Punkte für das Protokoll).