Protokoll Programmierbeispiel 1

Armin NASERI, Andor KURUCZ

Einleitung:

Das Protokoll dokumentiert die Struktur und Funktionsweise des Stock-Verwaltungsprogramms. Es bietet eine detaillierte Beschreibung der Datenstruktur, der Hashfunktion, eine Aufwandsabschätzung für verschiedene Operationen und eine Anleitung zur optimalen Nutzung des Programms.

Datenstruktur:

Das Programm verwendet eine Hash-Tabelle zur effizienten Verwaltung von Aktiendaten. Die Hash-Tabelle ist eine Datenstruktur, die schnellen Zugriff auf Elemente ermöglicht, indem sie Schlüsselwerte mithilfe einer Hashfunktion auf Indizes abbildet. Jeder Index der Tabelle verweist auf eine Liste von Aktien, die denselben Hashwert haben.

Die Datenstruktur umfasst die folgenden Hauptkomponenten:

- Stock: Eine Klasse, die Informationen über eine Aktie speichert, einschließlich Namens, WKN (Wertpapierkennnummer), Akronym und Historie von Aktiendaten.
- StockData: Eine Struktur, die Datenpunkte einer Aktie zu einem bestimmten Zeitpunkt enthält, wie Datum, Eröffnungskurs, Höchstkurs, Tiefstkurs, Schlusskurs, volumengewichteter Durchschnittskurs und Handelsvolumen.
- HashTable: Eine Klasse, die die Hash-Tabelle implementiert, um Aktienobjekte basierend auf ihren Akronymen zu speichern und zu verwalten.

Hashfunktion:

Die Hashfunktion wird verwendet, um den Hashwert für das Akronym einer Aktie zu berechnen. Die Implementierung der Hashfunktion ist recht einfach und summiert die ASCII-Werte der Zeichen im Akronym und nimmt dann das Modulo des Ergebnisses durch die Größe der Hash-Tabelle.

Aufwandsabschätzung:

- Hashfunktion (hash): Die Hashfunktion weist eine lineare Zeitkomplexität O(n) auf, da sie jeden Buchstaben im Akronym durchläuft, was bedeutet, dass ihre Leistung proportional zur Länge des Akronyms ist. Für eine große Anzahl von Aktien kann dies zu Kollisionen führen, was jedoch in diesem Programm nicht behandelt wird.
- Hinzufügen (add): Die Operation zum Hinzufügen einer Aktie zur Hash-Tabelle erfordert das Berechnen des Hashwerts und das Einfügen in die entsprechende Position in der Tabelle. Die durchschnittliche Zeitkomplexität beträgt O(1), aber in Fällen von Kollisionen kann sie bis zu O(n) steigen, wobei n die Anzahl der Elemente in der Kollisionsliste ist.
- Entfernen (remove): Die Operation zum Entfernen einer Aktie aus der Hash-Tabelle erfordert das Auffinden der Aktie und das Entfernen aus der entsprechenden Kollisionsliste. Die durchschnittliche Zeitkomplexität beträgt O(1), kann jedoch bis zu O(n) sein, wenn die Liste lang ist.
- Suchen (search): Die Operation zum Suchen einer Aktie in der Hash-Tabelle erfordert das Berechnen des Hashwerts und das Durchsuchen der entsprechenden Kollisionsliste. Der Aufwand beträgt O(n+3), wobei hier n die länge des Aktienkürzels ist und Drei für die Summierung der einzelnen Dezimalwerte(laut Hashfunktion), der Modulorechnung und dem Zugriff auf den bestimmten Index.
- Importieren (import): Die Operation zum Importieren von Aktiendaten aus einer CSV-Datei erfordert das Lesen der Datei, Parsen der Daten und Hinzufügen zur entsprechenden Aktienliste. Die Zeitkomplexität hängt von der Größe der Daten in der Datei ab und kann im schlimmsten Fall O(n) erreichen.
- Plotten (plot): Die Operation zum Plotten von Aktiendaten erfordert das Durchlaufen der Historie einer Aktie und das Ausgeben der Daten. Die Zeitkomplexität beträgt O(n), wobei n die Anzahl der Datensätze in der Historie ist.
- Speichern (save) und Laden (load): Die Operationen zum Speichern und Laden von Aktiendaten in eine Datei erfordern das Schreiben bzw. Lesen der Daten aus einer Datei. Die Zeitkomplexität hängt von der Anzahl der Aktien und der Größe der Historie ab und kann bis zu O(n) betragen.

Größe der Hashtabelle

Namenskonvention für aktienkürzel sind zwischen 2 und (den längsten den wir am NASDAQ gefunden hab) 6 Buchstaben. Der geringste Wert wär somit AA (also 65+65) und der größte ZZZZZZ(6*90) und diese nummerische Wert wird durch die Hash-Funktion durch die Tabellen-Größe "moduliert".

Aber aufgrund eines Zeitmangels ist eine schöne implementierung der Zuweisung auf die Hastabelle nicht unserer Erwartung vorhanden. Da unser Programm in jetziger Instanz auf externer Verkettung basiert, ist die Tabellen-Größe zugegebenerweise arbiträr, aber derzeit 100.
Geplant gewesen wäre eine lineare Sondierung zu implementieren:

Unsere Test-Aktie Microsoft(MSFT) gibt laut unserer Hashfunktion den Wert 14 heraus, wenn eine Aktie hinzugefügt wird mit identem Wert soll sie dementsprechend auf den Index 15 gesetzt werden.

Optimale Anwendung:

Beim Start des Programms wird das Hauptmenü angezeigt, welches folgende Optionen zur Auswahl bietet:

- ADD (a): Fügt eine neue Aktie hinzu. Sie werden aufgefordert, den Namen, die WKN und das Kürzel der Aktie einzugeben.
- DELETE (d): Löscht eine vorhandene Aktie anhand ihres Kürzels.
- IMPORT (i): Importiert Aktiendaten aus einer CSV-Datei für eine vorhandene Aktie.
- SEARCH (s): Sucht nach einer Aktie anhand ihres Kürzels und zeigt die neuesten Daten an.
- PLOT (p): Plottet die Schlusskurse der letzten 30 Tage einer Aktie.
- SAVE (v): Speichert die gesamten Aktiendaten in eine Textdatei.
- LOAD (l): Lädt zuvor gespeicherte Aktiendaten aus einer Textdatei.
- QUIT (x): Beendet das Programm.

Für die Auswahl der Option werden sie aufgerufen den jeweiligen Buchstaben einzugeben und mit der Eingabetaste zu bestätigen. Sie werden aufgefordert, Informationen einzugeben oder Aktionen durchzuführen, weshalb es wichtig ist den Anweisungen auf dem Bildschirm zu folgen.

WICHTIG

Es ist wichtig anzumerken, dass beim Aufruf der save-Funktion die Dateien im „saves“ Ordner gespeichert werden. Ebenso muss beim Importieren von Daten die csv-Datei sich im „files“ Ordner befinden und auch das gewünschte Kürzel der zugehörigen Aktie als Filenamen haben (Für die Aktie MSFT soll die CSV-Datei als MSFT.csv abgespeichert werden).