

ALGODAT – Programmierbeispiel 1 – Protokoll

Klassenübersicht

```
class stockLine:
    def __init__(self, date, closeLast, volume, open, high, low):
        self.date = date
        self.closeLast = closeLast
        self.volume = volume
        self.open = open
        self.high = high
        self.low = low
```

Ein Objekt repräsentiert eine Zeile aus einer CSV-Datei.

Übersicht der Hash-Funktionen

- `hash_function(word)`
Berechnet den Hash-Wert eines Strings, indem die ASCII-Werte der Buchstaben multipliziert und das Ergebnis modulo 1009 berechnet werden.
- `shiftIndex(hash_index, list)`
Prüft, ob der Index `list[hash_index]` bereits belegt ist. Falls ja, wird der Index mithilfe von **quadratischer Sondierung** weiter verschoben, bis ein freier Platz gefunden wird.
- `search_wkn(stockSymbolList, stockNameList, search_input)`
Sucht die WKN einer Aktie basierend auf dem Hash-Wert von `search_input`. Je nach Länge der Eingabe wird entweder in `stockSymbolList` oder `stockNameList` gesucht. Aufgrund der quadratischen Sondierung kann es vorkommen, dass der gefundene Wert nicht zur gesuchten Aktie passt.
- `compare(list, search_input, search_index)`
Überprüft, ob `search_input` mit `list[search_index][0]` übereinstimmt.
- `search_stock(wkn, stockList)`
Erstellt den Hash-Wert der WKN, prüft mit `compare()`, ob die Werte übereinstimmen, und gibt den `stock_index` zurück.
- `deleteStock(stockNameList, stockSymbolList, stockList)`
Findet die WKN einer Aktie anhand des Namens und Symbols und ersetzt die entsprechenden Einträge in allen drei Tabellen durch 1 (Tombstone).
- `load(stockList, stockLines, stockNameList, stockSymbolList)`
Lädt Daten aus drei verschiedenen JSON-Dateien und fügt sie in die entsprechenden Tabellen ein.

- `importStock(stockLines, stockList, stockSymbolList, stockNameList)`
Liest eine CSV-Datei ein, erstellt für jede Zeile ein Objekt und speichert die Daten mit der zugehörigen WKN in `stockList`.
- `save(stockList, stockSymbolList, stockNameList)`
Speichert die drei Tabellen in separate Dateien.
- `plot(stockSymbolList, stockNameList, stockList, search_input)`
Findet anhand der Eingabe (`search_input`) die zugehörigen Aktienkurse aus `stockList`. Diese Werte werden in **Floats konvertiert**, in eine Liste eingefügt und anschließend mit **matplotlib.pyplot** als Graph dargestellt.

Übersicht der Hash-Tabellen

Alle Hash-Tabellen haben maximal **1009 Einträge**:

1. **StockList:**

- Enthält [WKN, StockLine-Objekte (30 Stück)]
- Position = **Hash-Wert des Symbols der Aktie**

2. **StockNameList:**

- Enthält [WKN, Name der Aktie]
- Position = **Hash-Wert des Namens der Aktie**

3. **StockSymbolList:**

- Enthält [WKN, Symbol der Aktie]
- Position = **Hash-Wert der WKN der Aktie**

Funktionalität

Funktion	Beschreibung
ADD	Fügt eine Aktie hinzu (Eingabe: Name, Symbol, WKN).
DELETE	Löscht eine Aktie (Eingabe: Name, Symbol).
IMPORT	Importiert Aktien aus einer CSV-Datei (Eingabe: Dateiname, Name oder Symbol).
SEARCH	Sucht eine Aktie nach Name oder Symbol.
PLOT	Erstellt ein Diagramm des Aktienkurses (Eingabe: Name oder Symbol).
SAVE	Speichert die aktuellen Tabellen in Dateien (Eingabe: Dateinamen).
LOAD	Lädt Aktieninformationen aus Dateien.
QUIT	Beendet das Programm.

Das Protokoll ist nun klarer und einfacher zu verstehen. Lass mich wissen, falls du noch Änderungen brauchst! 🚀

AUFWANDABSCHÄTZUNG

Einfügen:

Best Case $O(1)$ (direkt Einfügen)

WorstCase $O(n)$ (wenn es die gesamte Liste durchgehen muss)

Suchen:

Best Case $O(1)$ (direkt Einfügen)

WorstCase $O(n)$ (wenn es die gesamte Liste durchgehen muss)

Löschen:

Best Case $O(1)$ (direkt Einfügen)

WorstCase $O(n)$ (wenn es die gesamte Liste durchgehen muss)