

1. Struktur **TL** (TrendLine)

Diese Struktur speichert alle relevanten Daten einer Trendlinie, darunter Start- und Endzeitpunkte, Preise, die berechnete Steigung und der Y-Achsenabschnitt. Zusätzlich enthält sie Informationen zur Richtung der Linie (Long oder Short) sowie den Namen und die Anzahl der Tage zwischen den beiden Punkten.

2. Klasse **TrendLinePrice**

Diese Klasse verwaltet und berechnet Trendlinien für einen Chart.

Wichtige Attribute:

- **time_1, time_2, price_1, price_2**: Speichern die Zeit- und Preispunkte der Trendlinie.
- **trendSteigung, trendAchsenAbschnitt**: Berechnete Steigung und Y-Achsenabschnitt der Linie.
- **TrendLine[]**: Ein Array zur Speicherung mehrerer Trendlinien.
- **currentPrice**: Speichert den aktuellen Preis der Trendlinie, die dem Marktpreis am nächsten liegt.
- **updateFrequency & nextUpdate**: Bestimmen, wann die nächste Aktualisierung erfolgen soll.

Methoden der Klasse:

a) **Init()** - Initialisiert alle Trendlinien

1. Löscht alle vorhandenen Trendlinien aus dem Speicher.
2. Durchsucht den Chart nach Trendlinien-Objekten.
3. Falls keine Trendlinie existiert, wird die Methode beendet.
4. Geht alle Trendlinien durch und prüft, ob sie gültig sind.
 - Ist der Start- und Endzeitpunkt identisch? → Fehler
 - Ist einer der beiden Preise **0.0**? → Fehler
 - Liegt das Enddatum in der Zukunft oder weit in der Vergangenheit? → Fehler
5. Falls die Trendlinie gültig ist, wird sie gespeichert und ihre Steigung berechnet.

b) **FindClosestTrendLine()** - Findet die dem aktuellen Marktpreis nächstgelegene Trendlinie

1. Durchläuft alle gespeicherten Trendlinien.
2. Berechnet für jede Linie den erwarteten Preis zum aktuellen Zeitpunkt.
3. Vergleicht den Abstand zwischen dem berechneten Trendpreis und dem aktuellen Bid-Preis.
4. Speichert die nächstgelegene Trendlinie.

c) `UpdateTrendLines()` - Aktualisiert Trendlinien, falls nötig

1. Durchläuft alle Trendlinien.
2. Berechnet den Abstand zwischen dem aktuellen Bid-Preis und dem Trendpreis.
3. Passt die Update-Frequenz an:
 - Falls der Abstand klein ist → Updates häufiger durchführen.
 - Falls der Abstand groß ist → Updates seltener durchführen.
4. Falls ein schnelles Update nötig ist, wird die Trendlinie neu berechnet.

d) `BerechneTrendlinie()` - Berechnet die Steigung und den Y-Achsenabschnitt

Die Steigung der Linie ergibt sich aus der Differenz der beiden Preise, geteilt durch die Differenz der Zeitpunkte.

Der Y-Achsenabschnitt wird anhand des Startpunkts und der Steigung berechnet.

e) `BerechnePreisZumZeitpunkt()` - Gibt den Preis zu einem bestimmten Zeitpunkt zurück

Setzt die berechnete Steigung und den Y-Achsenabschnitt in die lineare Gleichung ein, um den Preis zu einem gegebenen Zeitpunkt zu berechnen.

f) `Ausgabe()` - Gibt Informationen zu allen Trendlinien aus

Durchläuft das Array der Trendlinien und gibt für jede Trendlinie folgende Daten aus:

- Name der Linie
- Anzahl der Tage zwischen den Punkten
- Entfernung zum aktuellen Bid-Preis