Aktienhandelssimulator

MLZ-Prüfung

Fach	C# 2 Programmierung
Klasse	17I/18NI
Lehrperson	Sagi Nedunkanal
Semester	Frühlingssemester 2019
Prüfungsbeginn	04.07.2019
Abgabetermin	13.07.2019, 24:00 Uhr

1. Ausgangslage

Es soll ein Aktienhandelssimulator realisiert werden. Aktien können gekauft und verkauft werden. Die Kurse aller Aktien im Portfolio ändern sich sekündlich. Das Portfolio wird mit dem Kaufwert und dem aktuellen Kurswert bewertet.

Die Aufgabenstellung ist nicht in allen Details ausspezifiziert. Bei Unklarheiten wird erwartet, dass beim Dozenten nachgefragt wird.

2. Anforderungen

2.1. Benutzeroberfläche

Schreibe eine WPF-Applikation, um die folgenden Aktiendaten darzustellen:

a) Abkürzung der Aktien (z.B. „GOOG“)
b) Firmenname (z.B. „Google“ oder „Alphabet Inc.“)
c) Kaufkurs [CHF]
d) Stück [Ganzzahl]
e) Aktueller Kurs [CHF]
f) Kursverlauf [Datum, Zeit, Kurs] (z.B. 05.01.2015 - 17:30:01 - CHF 763.51)

Verwende das MVVM-Entwurfsmuster.

2.2. Aktien kaufen

Eine neue Aktie soll gekauft werden können, indem die Aktiendaten

a) Abkürzung der Aktien (z.B. „GOOG“)
b) Firmenname (z.B. „Google“ oder „Alphabet Inc.“)
c) Kaufkurs [CHF]
d) Stück [Ganzzahl]

erfasst sind und ein „Kaufen“-Knopf gedrückt wird. Die Abkürzung soll eindeutig sein (unique identifier). Das heisst, es soll nicht möglich sein, eine zweite Aktie mit der gleichen Abkürzung zu kaufen.

Zur Vereinfachung:
Es kann kein zweiter Kauf mit der gleichen Abkürzung getätigt werden. Ist beispielsweise bereits ein Kauf von „GOOG“ getätigt worden, ist es nicht nötig, dass man zusätzliche „GOOG“-Aktien kaufen kann. Man kauft einfach „GOOG2“. Das vereinfacht später die Portfoliobewertung bei Punkt 2.7.

2.3. Aktien verkaufen

Einzelne Teile einer Aktienposition aus dem Portfolio sollen verkauft werden können, indem die Aktiendaten «Kürzel» und «Stückzahl» aus Punkt 2.1 erfasst sind und ein „Verkaufen“-Knopf gedrückt wird.

Zur Vereinfachung:
Sind alle Aktien einer Firma verkauft, verschwindet die Aktienposition aus dem Portfolio. Das vereinfacht später die Portfoliobewertung bei Punkt 2.7.

2.4. Portfolioauflistung

In einer Liste sollen alle gekauften Aktien ersichtlich sein. Ein Listeneintrag kann beispielsweise wie folgt aussehen: „GOOG, Alphabet Inc., 15 Stk., CHF 763.51“.

Die Aktien und das Portfolio sollen in geeigneten Klasse gespeichert werden.

2.5. Aktiendetailansicht mit Mutationsmöglichkeit

Wenn eine Aktie in der Portfolioauflistung selektiert wird, sollen ihre Details in einer Aktiendetailansicht angezeigt werden. Die Aktiendaten Abkürzung, Firmenname, Kaufkurs und Stück sollen mutiert werden können. Der aktuelle Kurs und der Kursverlauf werden nur angezeigt und können nicht mutiert werden. Die Mutation soll die Portfolioauflistung aktualisieren.

Zur Vereinfachung:

- Der Kursverlauf kann anhand einer Tabelle dargestellt werden
  02.07.2019 17:01 – CHF 10.00
  02.07.2019 17:02 – CHF 10.50
  02.07.2019 17:03 – CHF 09.40
- Falls ihr wollt, könnt ihr zusätzlich noch eine Grafik anzeigen:
  https://www.nuget.org/packages/LiveCharts/

2.6. Kursveränderungssimulation

Zeitgesteuert soll jede Sekunde der Kurs aller Aktien im Portfolio leicht geändert werden. Die Änderung wird durch einen Timer ausgelöst und im Kursverlauf jeder Aktie protokolliert. Die Änderung kann z.B. einfach mit += 0.1 oder -=0.1 oder einem geeigneten Zufallswert simuliert werden. Keine zu grossen Kurssprünge zulassen.

Der Timer soll durch eine CheckBox aktiviert und deaktiviert werden können. Beim Start soll sie deaktiviert sein.

2.7. Portfoliobewertung

Die Portfoliobewertung soll folgende Werte anzeigen:

a) Kaufwert des Portfolios [CHF]
b) Aktueller Wert des Portfolios [CHF]
c) Gewinn/Verlust als absoluter Betrag [CHF]
d) Gewinn/Verlust als Prozentwert [%]

Punkte c) und d) sollen grün sein, wenn der Wert positiv ist. Sie sollen rot sein, wenn der Wert negativ ist.

Zur Vereinfachung:

- Wird eine Aktienposition komplett verkauft, muss ihr Gewinn/Verlust nicht in die Portfoliobewertung einfliessen, sondern kann ignoriert werden. Dann sieht es so aus, als wäre die Aktie nie gekauft worden. Das soll die Berechnung vereinfachen.
- Es müssen auch keine Teilverkäufe mit einkalkuliert werden. Beispiel: wenn ursprünglich 10 Aktien vorhanden waren und 6 verkauft wurden, reicht die Bewertung mit den übrigen 4 Aktien; es ist irrelevant, ob die 6 verkauften Aktien einen Gewinn oder Verlust eingefahren haben.
- Es gibt nur CHF als Währung. D.h. keine Wechselkursumrechnungen nötig.

3. Qualitätssicherung

3.1. Testdaten

Beim Applikationsstart sollen genau 2 Aktien erfasst sein.

3.2. Unittests

Es sollen sinnvolle Unittest geschrieben werden; mindestens 3.
Beispiel:

// Arrange	Es befinden sich 2 Google-Aktien im Portfolio, die zu je CHF 10.- gekauft worden sind.
// Act	Der Kurs steigt auf CHF 11.-.
// Assert	Das erwartet Ergebnis ist: 1. Aktueller Wert des Portfolios = CHF 22.- 2. Gewinn/Verlust als absoluter Betrag = CHF 2.- 3. Gewinn/Verlust als Prozentwert = 10%

4. Dokumentation

Es soll eine minimale Benutzerdokumentation erstellt werden, damit der Benutzer nachlesen kann, wie die Anwendung zu bedienen ist: pro Funktionalität ein Kapitel mit Screenshots und Sprechblasen oder mit Screenshots und einer stichwortartigen Anleitung reicht.

5. Bewertungskriterien

1. Umsetzung der funktionalen Anforderungen (1/4)
2. Bewertung des GUIs (1/4)
3. Bewertung des Codes (1/4)
4. Testen der Applikation (1/4)

- Es ist keine Persistierung in eine DB nötig.
- Erlaubt: Internet als Quelle, Beispiele aus dem Unterricht usw.
- Nicht erlaubt: Kopieren grösserer Codemengen, Mitarbeit Dritter usw.
  - Selbstkontrolle: «Kann ich jede Zeile Code erklären?»
- Missachtung der Prüfungsregeln führen zu einer Note 1.
- Verspätete Abgabe führt zu Notenabzug:
  - Bis 6h Verspätung → 0.5 Note Abzug
  - Bis 12h Verspätung → 1.0 Note Abzug
  - >12h Verspätung → Note 1.0

6. Abgabe

Die Applikation wird via GitHub abgegeben:

1. Den Quellcode ablegen unter
   `CS2\Prüfungen\Prüfung 3 MLZ\Quellcode\Aktienhandelssimulator`
2. Das Kompilat als ZIP ablegen unter
   `CS2\Prüfungen\Prüfung 3 MLZ\Kompilat\Aktienhandelssimulator.zip`
3. Die Dokumentation ablegen unter
   `CS2\Prüfungen\Prüfung 3 MLZ\Dokumentation`

4. Die existierende Solution verwenden
   `CS2\Prüfungen\Prüfung 3 MLZ\<Vorname Nachname> CS2 Prüfung 3 MLZ.sln`
5. Alles hochladen

Verwende die Versionsverwaltung, um regelmässig Sicherungen des aktuellen Standes zu erstellen (Commit) und hochzuladen (Push).