# Individualleistung Programmierung II

## Fakultät für Wirtschaft

**Studiengang**: Wirtschaftsinformatik

| | | | |
|---|---|---|---|
| **Kurs:** | WWI24SCA | **Semester:** | 2 |
| **Hilfsmittel:** | Keine Bibliotheken außer java.* erlaubt! | **Bearbeitungszeit:** | Bis 09.05.2025 |

*Die im Folgenden verwendete Bezeichnung „der User“ beinhaltet User aller Geschlechter.*

**Zusammenfassung:**

Für diese Individualleistung sollen die Studierenden eine Software bauen, die eine Aktien-Tradingseite abbildet. Der User soll die Möglichkeit haben, pro Tag die aktuellen Kurse angezeigt zu bekommen, Geld investieren und Aktien wieder verkaufen zu können. Nach 10 Tagen wird ein Resümee gezogen, wie viel Geld der User gewonnen oder verloren hat.

Für die Software werden die Aktienkurse im CSV-Format eingelesen und dem User in einer Konsolenoberfläche die Möglichkeit gegeben, die Investitionen manuell oder automatisch zu steuern.

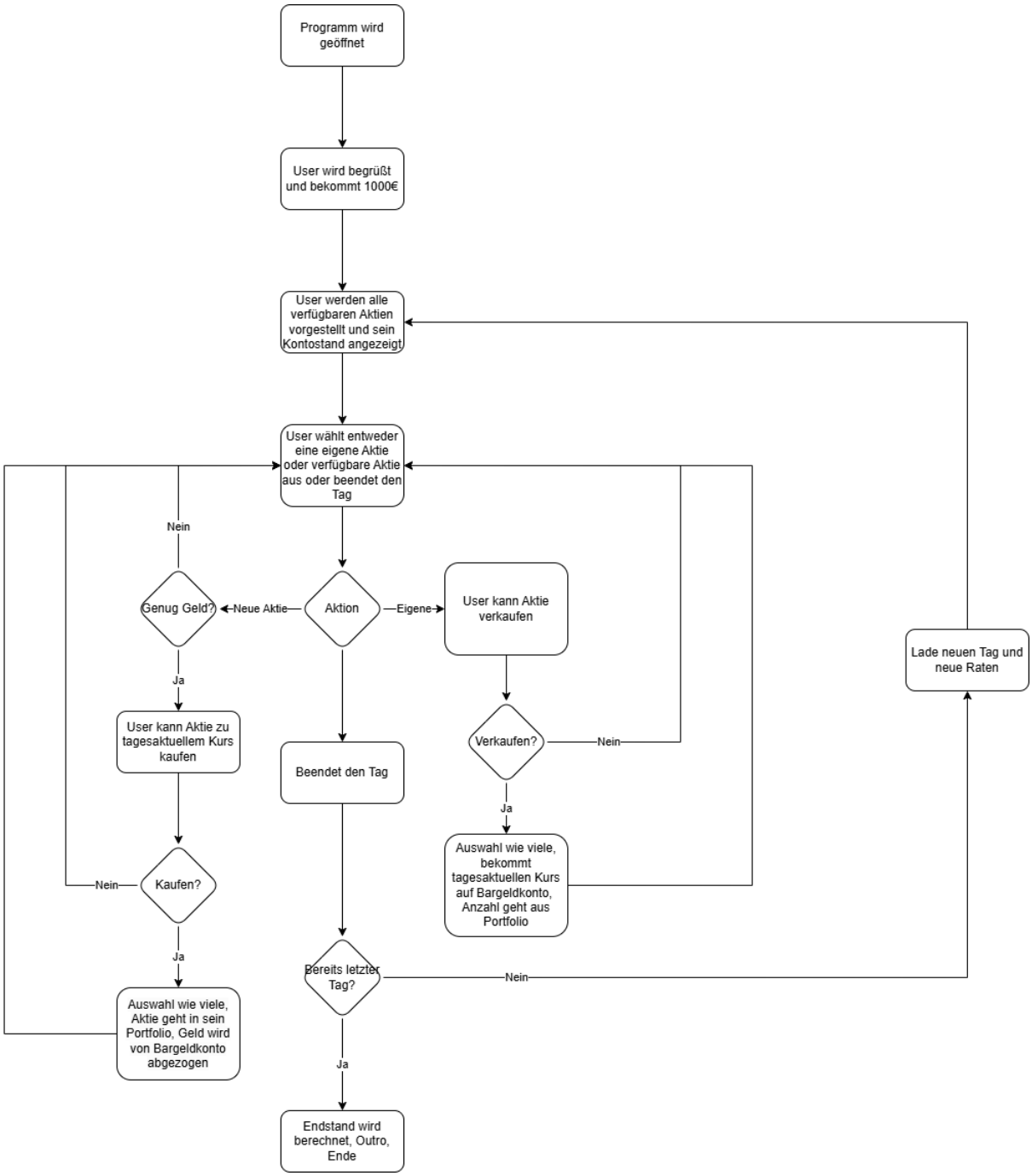
Zusätzlich muss der Code mindestens kommentiert werden – eine weiterführende Doku ist sinnvoll, sofern die Kommentierung nicht ausreicht.

**Unterlagen:**

Die folgenden Dateien werden zur Verfügung gestellt und sollen für die Software verwendet werden. Weitere Quellen werden nicht benötigt.

| | |
|---|---|
| ^GDAXI | Der DAX-Kurs in dem entsprechenden Zeitraum |
| Diverse Aktiendateien | Der entsprechende Aktienkurs über den Zeitraum unter Angabe von abschließendem Wert und Volumen. |
| Namings.csv | Match zwischen Aktienkürzel und korrekte Name der Aktien / des Unternehmens. |

**Der Programm-Ablauf:**



**Anforderungen:**

Die Anforderungen unterteilen sich in MUSS, SOLL und KANN-Anforderungen.
Alle MUSS-Anforderungen müssen erfüllt sein, sonst gilt die Prüfungsleistung als nicht bestanden.
SOLL-Anforderungen machen den Großteil der Note aus – sind alle Anforderungen zufriedenstellend erfüllt, steht einer sehr guten Note wenig im Weg.
KANN-Anforderungen können ggf. Schwächen in der Erfüllung der SOLL-Anforderungen ausbessern – es macht aber nichts, wenn keine KANN-Anforderungen im Programm erfüllt sind. MUSS-Anforderungen können damit nicht ausgebessert werden!

MUSS:

| **Anforderung** | **Beschreibung** |
|---|---|
| M1 | Das Programm ist in Java geschrieben und lässt sich ausführen |
| M2 | Das Programm ist in Eigen- und Einzelarbeit geschrieben |
| M3 | Das Programm verwendet die o.g. CSV-Dateien unverändert als Datenbasis. |
| M4 | Das Programm hat den unter „Der Programm-Ablauf“ beschriebenen Ablauf. |
| M4.a | Der User startet mit einem Guthaben von 1000€. |
| M4.b | Dem User wird zu Beginn des jeweiligen Tages sein Kontostand angezeigt, inklusive der aktuellen Werte seiner Aktien. |
| M4.c | Der User kann aus allen Aktien, die als Dateien mitgegeben wurden, Investitionen tätigen – heißt kaufen und verkaufen. |
| M4.d | Sobald der User den Tag beendet, werden die Aktienkurse des nächsten Tags geladen und der Kreislauf beginnt von Vorne. |
| M4.e | Nach 10 Tagen endet der Kreislauf und der User wird über Gewinne und Verluste informiert. |
| M5 | Der User soll sich die Kursentwicklungen seit gestern von allen verfügbaren Aktien anzeigen lassen können. |
| M6 | Der User soll sein Portfolio (welche Aktien er besitzt) inklusive aktuelle Werte anzeigen lassen können. |
| M6.a | Dem User werden für seine Aktien die Kursentwicklungen seit gestern in Prozent angezeigt. |
| M6.b | Dem User werden für seine Aktien die Kursentwicklungen seit dem Kauf in Prozent gezeigt. |
| M7 | Wenn der User kein Geld mehr hat, soll auch keine Investition mehr möglich sein – darauf soll das Programm ihn hinweisen. |
| M8 | Der komplette Code muss sinnstiftend kommentiert werden. Reicht die Kommentierung nicht zum Verständnis aus, kann auch eine weiterführende Dokumentation angelegt werden |

SOLL:

| Anforderung | Beschreibung |
|---|---|
| S1 | Die Daten pro Aktie werden in einem optisch übersichtlichen und angenehmen Format präsentiert. |
| S2 | Es sollen zu den im Programm-Ablauf definierten Schritten auch automatische Lösungen angeboten werden: |
| S2.a | Das Programm soll „die Aktie des Tages“ vorschlagen, die voraussichtlich eine gute Investition darstellt. |
| S2.b | Der User soll an einem Tag auswählen können, dass das Programm für ihn seine Entscheidungen (Kauf/Verkauf) übernimmt. |
| S2.c | Alternativ soll das ganze Programm auch automatisch (aka. Bot) durchlaufen können, bei dem die Logik die besten vorauszusehenden Investitionen trifft. |
| S3 | Nachdem das Programm durchlaufen ist, wird die rentabelste genutzte Aktie gesucht und präsentiert. |
| S4 | Der User wird sprachlich sauber durch das Menü geführt, sodass er immer weiß, wo er sich befindet und was von ihm verlangt wird. |
| S5 | Das Programm stürzt bei unerwarteten Eingaben nicht ab |
| S6 | Es wird mindestens eine Hashmap verwendet |
| S7 | Es wird mindestens eine eigenprogrammierte Datenstruktur auf Basis von Nodes zur Datenspeicherung verwendet |
| S8 | Es wird mindestens ein generischer Datentyp verwendet |
| S9 | Es wird mindestens ein Interface oder eine abstrakte Klasse verwendet |
| S10 | In der Kommentierung steht bei jeder Funktion deren Softwarelaufzeit korrekt beschrieben. Die O-Notation genügt hierfür |

KANN:

| Anforderung | Beschreibung |
|---|---|
| K1 | Der User kann zu einer Aktie einen Kommentar hinterlegen, der jedes Mal angezeigt wird, wenn die Aktie aufgerufen wird. |
| K2 | Nachdem das Programm durchlaufen ist, wird der unglücklichste Verkauf gesucht und mit dem entgangenen Gewinn präsentiert. |
| K3 | Die Userein- und -Ausgabe soll sowohl in Englisch als auch Deutsch verfügbar sein. Dies wird am Anfang auf Englisch abgefragt |
| K4 | Das Programm wird zusätzlich zum Code als ausführbare Datei abgegeben |
| K5 | Das Programm lädt die Daten in einem separaten Thread vor, während der User durch das initiale Menü navigiert |
| K6 | Es gibt im kompletten Code keine Funktion mit der Laufzeit $O(n^2)$ oder langsamer |
| K7 | Die Daten sollen normalisiert angezeigt werden, also bei Namen nur der erste Buchstabe groß, keine unnötigen Sonderzeichen, etc. |
| K8 | Das Programm erstellt beim ersten Öffnen eine CSV-Datei, die die bisherigen Durchläufe dokumentiert, welche Investitionen wann getroffen wurden und wie viel Gewinn/Verlust gemacht wurde. |
| K9 | Statt der Abkürzung wird der korrekte Name der Aktie (zu finden in der Naming.csv) angezeigt, alternativ werden beide Bezeichnungen angezeigt. |