

## Konzepte der Programmierung: Seminararbeit

### Simulation von Investitionsszenarien mit verschiedenen Anlageklassen

Sie sollen eine Anwendung entwickeln, die Anwendern ermöglicht, die Entwicklung von Investitionen in verschiedene Anlageklassen wie Aktien, Anleihen oder Immobilien zu simulieren und miteinander zu vergleichen. Die Anwendung soll den Anlagebetrag, den Investitionszeitraum und die Auswahl der Anlageklassen berücksichtigen, um zukünftige Renditen zu berechnen. Ziel ist es, realistische Szenarien abzubilden, indem eine stochastische Komponente wie Volatilität einbezogen wird.

Folgende *Mindestanforderungen* an die Anwendung müssen beachtet werden:

**Als Benutzer** möchte ich einen Anlagebetrag, eine Laufzeit und eine oder mehrere Anlageklassen eingeben können, damit ich die Renditeentwicklung meiner Investition simulieren kann.

**Als Benutzer** möchte ich eine übersichtliche tabellarische Ausgabe der jährlichen Entwicklung meiner Investition sehen, damit ich die Unterschiede zwischen den Anlageklassen nachvollziehen kann.

**Als Benutzer** möchte ich sicherstellen, dass meine Eingaben (z. B. positive Beträge und gültige Laufzeiten) validiert werden, damit ich keine falschen Ergebnisse erhalte.

**Als Benutzer** möchte ich mindestens die Anlageklassen Aktien, Anleihen und Immobilien wählen können.

**Als Benutzer** möchte ich, dass die Anwendung auch mit fehlerhaften Eingaben (z. B. negative Werte oder unbekannte Anlageklassen) sinnvoll umgeht, damit ich keine Abstürze erlebe.

**Als Benutzer** möchte ich, dass die Ergebnisse der Simulation verständlich und nachvollziehbar dargestellt werden, damit ich sie als Entscheidungsgrundlage nutzen kann.

**Als Entwickler** möchte ich, dass die Anwendung die grundlegenden Konzepte der objektorientierten Programmierung (z. B. Vererbung, Polymorphie) verwendet, damit sie modular, wartbar und erweiterbar bleibt.

**Als Entwickler** möchte ich, dass der Code der Anwendung gut dokumentiert und modular ist, damit er für andere verständlich und wartbar bleibt.

Eine (nicht verpflichtende) *Erweiterung* des Funktionsumfangs ist z.B. durch folgende Anforderungen möglich:

**Als Benutzer** möchte ich die Simulation über eine grafische Benutzeroberfläche (GUI) durchführen können, damit die Eingabe und Visualisierung der Ergebnisse einfacher und ansprechender ist.

**Als Benutzer** möchte ich, dass die Anwendung die Renditeentwicklung in einem Diagramm darstellt, damit ich die Ergebnisse visuell nachvollziehen kann.

**Als Benutzer** möchte ich historische Daten über eine Webschnittstelle abfragen können, damit ich die Entwicklung von Anlageklassen retrospektiv analysieren und mit der Simulation vergleichen kann.

## Beispiel für eine mögliche Eingabe und Ausgabe in der Konsolenanwendung

Willkommen zum Investitionsrechner!

Bitte geben Sie den Anlagebetrag ein: 10000

Bitte geben Sie die Laufzeit in Jahren ein: 5

Wählen Sie die Anlageklassen aus:

1. Aktien
2. Anleihen
3. Immobilien

Ihre Wahl: 1

Ergebnisse der Investitionssimulation:

Jahr	Startkapital	Rendite (%)	Endkapital
1	10.000,00 €	8,52 %	10.852,00 €
2	10.852,00 €	-3,15 %	10.509,82 €
3	10.509,82 €	12,34 %	11.804,14 €
4	11.804,14 €	7,89 %	12.736,87 €
5	12.736,87 €	5,67 %	13.460,09 €

Gesamtrendite nach 5 Jahren: 3.460,09 €

Endkapital: 13.460,09 €

## Berechnung des Kapitals nach einem Jahr

Eingaben:

- **Startkapital:  $K_0$**   
Das Kapital, dass zu Beginn investiert wird (z.B. 10.000€)
- **Durchschnittliche Rendite:  $r$**   
Die erwartete jährliche Rendite für die gewählte Anlageklasse (z.B. 15% für Aktien)
- **Volatilität:  $\sigma$**   
Die Schwankungsbreite der Rendite, die die Unsicherheit im Markt simuliert (z. B. 15 % für Aktien)
- **Zufällige Schwankung:  $z$**   
Ein Zufallswert, der normalerweise aus einer Normalverteilung gezogen wird (z. B.  $z=-0.05$  für eine leicht negative Abweichung)

Das Kapital  $K_1$  nach einem Jahr wird berechnet als:

$$K_1 = K_0 * (1 + (r + \sigma * z))$$

## Beurteilungskriterien

### 1. Formales

Teilbenotung	1,0	1,3	1,7	2,0	2,3	2,7	3,0	3,3	3,7	4,0	4,3	4,7	5,0
<b>Gestaltung</b> (gem. Leitfaden zur formalen Gestaltung von Seminar- und Abschlussarbeiten)													
<b>Rechtschreibung und Grammatik</b> (Korrektheit; inklusive Interpunktion)													
<b>Umfang</b> (Problemangemessenheit - gem. Leitfaden zur formalen Gestaltung von Seminar- und Abschlussarbeiten)													
<b>Zusammenfassende Bewertung</b>													

### 2. Konzept

Teilbenotung	1,0	1,3	1,7	2,0	2,3	2,7	3,0	3,3	3,7	4,0	4,3	4,7	5,0
<b>Kohärenz und Modularität</b> (Logische und klare Programmdesign-Struktur)													
<b>Lesbarkeit</b> (Verständlicher, gut dokumentierter und wiederverwendbarer Code)													
<b>Erweiterbarkeit</b> (Leichte Anpassung und Erweiterung des Programms)													
<b>Robustheit und Fehlerbehandlung</b> (Fehlerbehandlung und stabile Funktionalität)													
<b>Angemessenheit der Abstraktion</b> (Sinnvolle und benutzerfreundliche Umsetzung von Konzepten)													
<b>Effektivität der Polymorphie</b> (Unterschiedliche Implementierungen werden durch einheitliche Schnittstellen sinnvoll genutzt)													
<b>Kapselung</b> (Daten und Funktionen sind sinnvoll verborgen und nur über klar definierte Schnittstellen zugänglich)													
<b>Vermeidung von Redundanzen</b> (Code-Duplikate werden durch abstrakte Strukturen oder Wiederverwendung vermieden)													
<b>Benutzerfreundlichkeit</b> (Die Bedienung der Anwendung ist intuitiv und effizient gestaltet)													
<b>Zusammenfassende Bewertung</b>													

## 3. Inhalt

Teilbenotung	1,0	1,3	1,7	2,0	2,3	2,7	3,0	3,3	3,7	4,0	4,3	4,7	5,0
<b>Gliederung</b> (Logik, Problembezogenheit, Stringenz)													
<b>Problemstellung und Zielsetzung</b> (nachvollziehbare Herleitung)													
<b>Argumentation</b> (Abgewogenheit, Logik, Neutralität, Problembezogenheit, Stringenz)													
<b>Sprachlicher Ausdruck</b> (Nachvollziehbarkeit, Verständlichkeit, Verwendung von Fachtermini)													
<b>Darstellung</b> (Nutzung von UML Diagrammen, sinnvolle Abstraktion der Konzepte)													
<b>Problemlösung</b> (Breite u. Tiefe der Problemdurchdringung, Eigenständigkeit der Problemlösung, kritische Reflexion des Vorgehens)													
<b>Zusammenfassende Bewertung</b>													

**Hinweis zur Gesamtbewertung:**

Die Gesamtnote muss nicht aus dem arithmetischen Mittel der Einzelnoten gebildet werden. Auch Defizite in einzelnen Kriterien können zu einer nicht mehr ausreichenden Gesamtbewertung führen.