

Bank Account

Vorbereitung:

Erstellen Sie eine Klasse *BankAccount* mit folgenden Eigenschaften:

- Privates Feld *balance* (double), welches den aktuellen Kontostand repräsentiert.
- Konstruktor, welcher ein neues Bankkonto mit einem initialen Kontostand einrichtet.
- Methode *deposit*, die die Einzahlung eines bestimmten Betrags auf das Konto vornimmt. Implementieren Sie dabei folgenden Vorgang:
 - Zunächst soll eine Wartezeit von einigen Sekunden initiiert werden, die den Aufbau zu anderen Systemkomponenten simuliert.
 - Dann soll der aktuelle Kontostand zwischengespeichert werden (*oldBalance*).
 - Anschließend soll der Kontostand aktualisiert werden.
 - Schließlich soll eine Meldung in der Konsole ausgegeben werden:
Alter Kontostand: € xx.xxx, Einzahlung: € xx.xxx, Neuer Kontostand: € xx.xxx
- Methode *withdraw*, die die Auszahlung eines bestimmten Betrags vom Konto vornimmt. Implementieren Sie diesen Vorgang in gleicher Form, wie die Einzahlung.

Aufgabenstellung:

Erstellen Sie nun eine Klasse *Main*, welche folgende Vorgänge abbildet:

- Erstellen Sie ein neues Bankkonto mit einem initialen Kontostand von € 10.000.
- Erstellen Sie 4 Threads, welche folgende Vorgänge abbilden:
 - Auszahlung von € 2.500
 - Einzahlung von € 5.000
 - Auszahlung von € 2.500
 - Auszahlung von € 5.000
- Starten Sie alle Threads.
- Warten Sie, bis alle Threads abgearbeitet sind und drucken Sie den endgültigen Kontostand.
- Starten Sie das Programm mehrmals und analysieren Sie die Ergebnisse.
- Beurteilen Sie, inwieweit das Programm so funktioniert, wie Sie das erwartet haben.
- Führen Sie gegebenenfalls notwendige Änderungen hinsichtlich Synchronisation der Vorgänge durch, sodass die Klassen *BankAccount* entsprechend Ihrer Erwartungen funktioniert.
- Beobachten und vergleichen Sie in diesem Zusammenhang die unterschiedliche Wirkung von *synchronized methods* und *synchronized blocks*.