

## Objektorientierte Modellierung Implementierung einer Banksoftware

---



🔗 **Hinweis:** Auf der Rückseite seht ihr auf der rechten Seite das *Implementierungsklassendiagramm* zum *Entwurfsklassendiagramm* links.

### 🔗 Aufgabe 1

Analysiert die Diagramme und vergleicht sie mit dem Entwurf aus dem Unterricht.



### 🔗 Aufgabe 2

Implementiert die Klasse  Kunde entsprechend des *Implementierungsdiagramms*. Geht dazu so vor:

- Öffnet BlueJ und erstell ein neues Projekt. Speichert das Projekt in eurem Laufwerk (📁 N: ).
- Erstellt die Klasse  Kunde als leere Klasse, indem ihr auf  klickt und unten „Leere Klasse“ auswählt.
- Deklariert die Objektvariablen `name`, `geburtstag`, `adresse` und `konto` in der Klasse.
- Implementiert den *Konstruktor* der Klasse, der die Attribute initialisiert.
- Implementiert die *Getter* und *Setter*.

### 🔗 Aufgabe 3


Implementiert die Klasse  Konto entsprechend des *Implementierungsdiagramms*. Geht dazu so vor:

- Erstellt die Klasse  Konto als leere Klasse wie oben.
- Deklariert die Objektvariablen der Klasse (mit Ausnahme der Transaktionen).
- Implementiert den *Konstruktor* der Klasse, der die Attribute initialisiert. Dabei sollen `kontostand`, `dispo` und `zinssatz` zunächst null sein und die vierstellige `pin` soll zufällig generiert werden. (Siehe  Random.)
- Implementiert die *Getter* und *Setter*.
- Implementiert die Methode `einzahlen` und `auszahlen`. `auszahlen` soll `true` zurück gegeben werden, wenn die Auszahlung (unter Berücksichtigung des Dispos) erfolgreich war.
- Implementiert die Methode `ueberweisen`. Sie bekommt das Zielkonto als Parameter und ruft die Methode `einzahlen` des Zielkontos auf.

Testet die Klasse ausgiebig.

### 🔗 Aufgabe 4

Implementiert die Klasse  Transaktion entsprechend des *Implementierungsdiagramms*. Geht dazu ungefähr so vor, wie bei den anderen Klassen.

Modifiziert die Klasse  Konto dann so, dass bei jeder Einzahlung, Auszahlung und Überweisung ein neues Transaktion-Objekt mit den passenden Informationen erstellt wird. Das neue Objekt soll in einer Objektvariablen „letzteTransaktion“ gespeichert werden. Ergänzt auch einen entsprechenden *Getter*.

