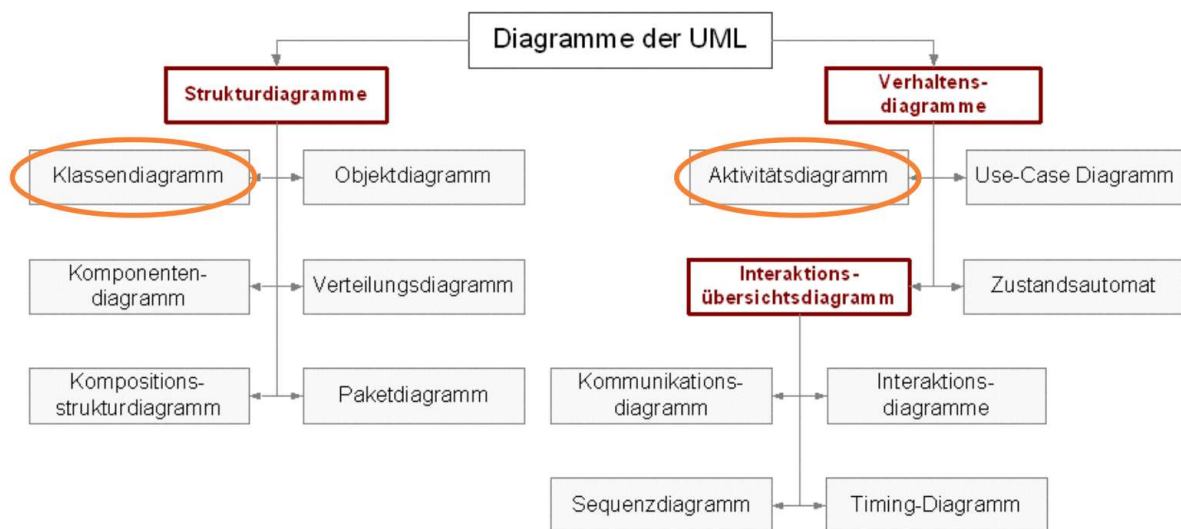


13. OOP UML

13.1. Definitionen

- UML = Unified Modeling Language
- bietet Unterstützung bei der Entwicklung von Software
- gliedert sich in die Phasen
 - OOA = objektorientierte Analyse
 - zur Analyse der Objekte und ihrer Beziehungen
 - OOD = objektorientiertes Design
 - Konzeption der entsprechenden Klassen und der Benutzeroberflächen
 - OOP = objektorientierte Programmierung
 - Implementierung der Klassen in der Programmiersprache
- Strukturdiagramme modellieren das zu entwickelnde System in statischer Hinsicht - sprich zeitunabhängig
- Verhaltensdiagramme modellieren in dynamischer Hinsicht



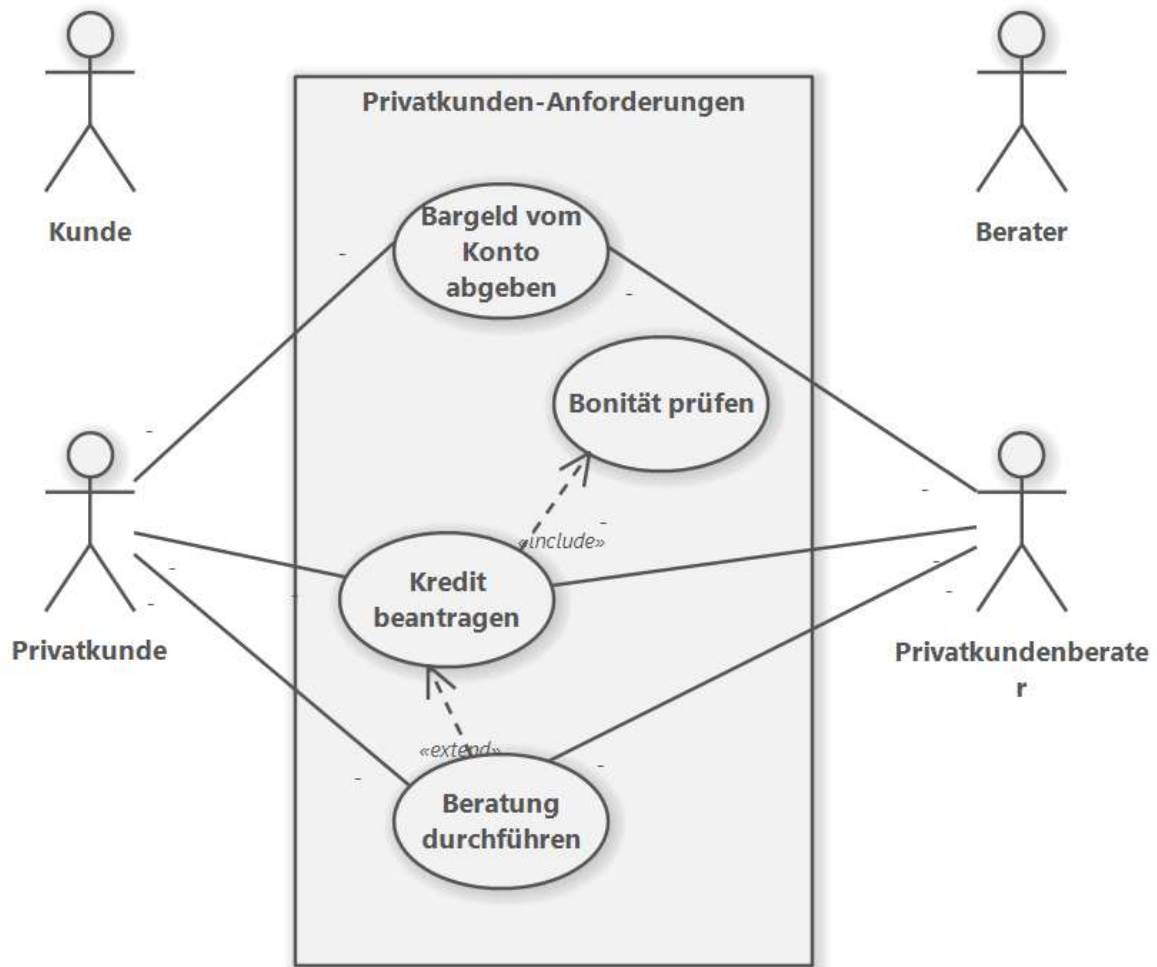
13.2. Verhaltensdiagramme

- Anwendungsfalldiagramm
 - ein Anwendungsfall (use case) beschreibt eine Funktionalität eines Systems – sie werden als Ellipse dargestellt – ein System umfasst meist mehrere Anwendungsfälle
 - die Kommunikation oder Interaktion mit den Anwendungsfällen erfolgt durch Akteure – entweder Personen (Strichmännchen) bzw. Systeme/Maschinen (Rechteck)
 - Akteure befinden sich immer außerhalb der Systemgrenzen
 - Akteure stehen mit den Anwendungsfällen in Beziehung (Assoziation) - sie werden durch Linien dargestellt
 - mit der Multiplizität werden die Beziehungen qualifiziert, sie gibt an, wie viele Akteure mit wie vielen Anwendungsfällen in Beziehung stehen
 - z.B. ein Bestellvorgang wird von einem oder mehreren Sachbearbeitern bearbeitet (1..*)
 - include wenn ein Anwendungsfall einen anderen zwingend einschließt
 - extend ein Anwendungsfall bindet unter bestimmten Umständen ein

- Sequenzdiagramm
 - es wird die Kommunikation zwischen Objekten eines Softwaresystems dargestellt
 - die Kommunikation besteht aus Nachrichten
 - im Interaktionsrahmen werden Objekte, deren Lebenszeit und die Kommunikation untereinander in einem zeitlichen Ablauf betrachtet

- Aktivitätsdiagramm
 - stellt das Verhalten des Softwaresystems dar
 - es zeigt die Prozesse und deren Reihenfolge
 - es ähnelt einem Programmablaufplan
 - es gibt
 - Aktionen (Rechteck mit abgerundeten Ecken)
 - Steuerungsfluss

- Anwendungsfalldiagramm am Beispiel Geschäftsprozesse für das Privatkunden-Geschäft



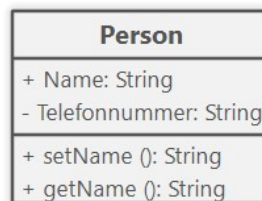
13.3. Strukturdiagramme

- Klassendiagramm
 - eine statische Sicht auf das zu entwickelnde Softwaresystem
 - Klassen und ihre Beziehungen werden dargestellt
 - einzelne Objekte und ihre Attributwerte (Zustände) werden nicht angezeigt
- Darstellung einer Klasse

Attribut mit Datentype

Sichtbarkeit:

+ öffentlich (public)
- privat (private)
geschützt (protected)



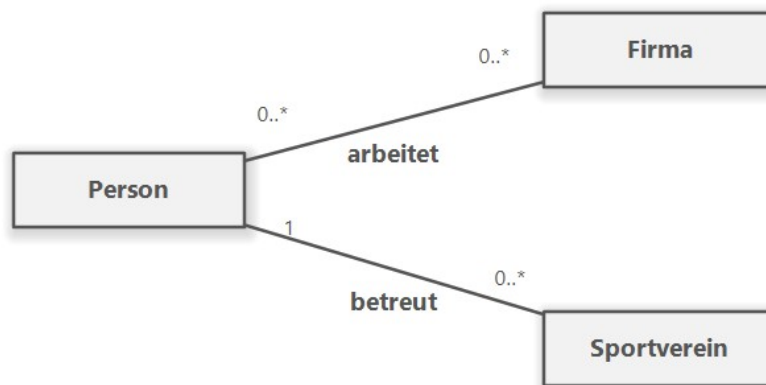
Attribute

Methoden

- Beziehungen zwischen Klassen
 - **Generalisierung bzw. Spezialisierung** ist eine Beziehung zwischen einer Basis-Klasse und einer abgeleiteten Klasse bzw. umgekehrt – sie wird auch **Ist-Beziehung** genannt
 - z.B. Person --- Kunde
 - von **Polymorphie** wird gesprochen, wenn Methoden in mehreren Klassen den gleichen Namen haben, aber unterschiedliche Aufgaben erfüllen
 - **Assoziation** ist eine Beziehung zwischen Klassen, die miteinander verknüpft sind
 - die eine Klasse kennt die andere Klasse oder
 - die eine Klasse ruft eine Methode der anderen Klasse auf
 - Multiplizität gibt an wie viele Objekte der einen Klasse mit wie vielen Objekten der anderen Klasse in Verbindung stehen
 - z.B. Person --- Geschäft oder Kunde --- Rechnung

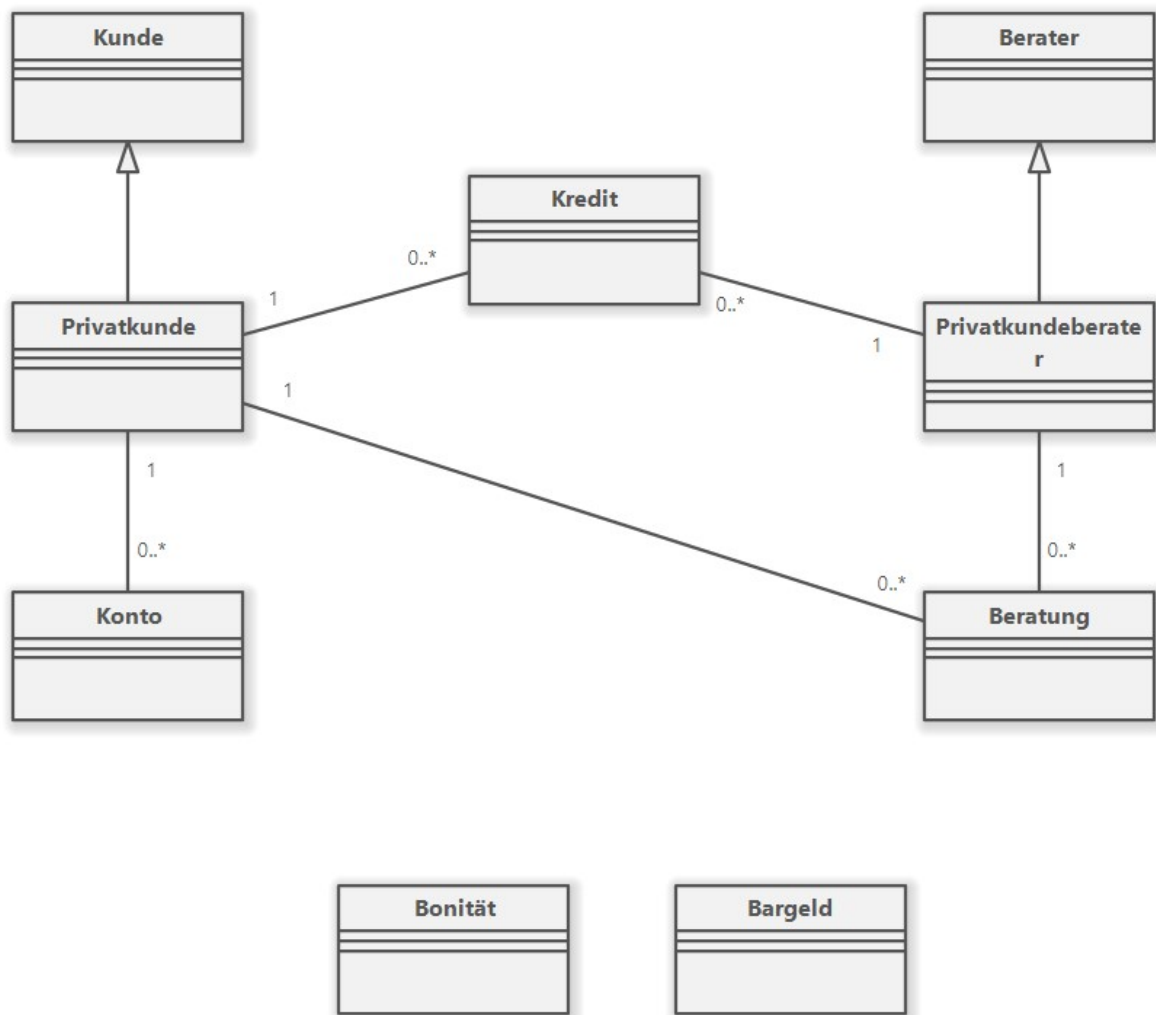
0	keins
1	genau eins
*	beliebig viele
0..*	keins oder beliebig viele
1..*	eins oder beliebig viele
1..3	eins, zwei oder drei
4..20	4 bis 20
1, 5, 7	eins, fünf oder sieben

- **Aggregation** ist eine spezielle Aggregation
- die verknüpften Klassen beschreiben eine Ganzes-Teile-Beziehung – sie wird auch **Hat-Beziehung** genannt
- z.B. Notebook --- Festplatte
- **Komposition** ist eine Aggregation, die eine zusätzliche besondere Eigenschaft hat
- es gibt starke Abhängigkeiten zwischen dem Ganzen und den Teilen
- z.B. Kunde --- Konto oder Firma --- Mitarbeiter
-



- Objektdiagramm
 - ist eine Art Schnappschuss des Softwaresystems und zeigt damit konkrete Objekte und ihre aktuellen Zustände

- Klassendiagramm am Beispiel Geschäftsprozesse für das Privatkunden-Geschäft



- Entwurfshinweis:
 - alle Akteure als Kandidaten für Klassen aufnehmen
 - Anwendungsfälle nach Substantiven durchsuchen – Substantive sind mögliche Klassenkandidaten
 - Kandidaten, die mehrere Attribute haben, sind mögliche Klassen
 - Kandidaten, die keine Attribute haben, sind oftmals selbst nur Attribute einer Klasse