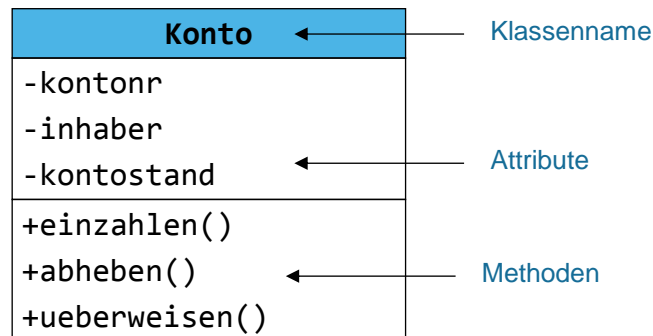


## Beispiel Konto-Klasse

- Zu entwickeln ist eine einfache Klasse für ein **Konto**. Dabei werden zunächst nur die einfachsten Konzepte der Objektorientierung umgesetzt. Später wird dieses Beispiel mehrfach ergänzt und erweitert werden.
- Klasse Konto
  - ⇒ Mögliche Attribute:
    - Kontonummer,
    - Inhaber,
    - Kontostand
  - ⇒ Mögliche Methoden:
    - abheben,
    - einzahlen,
    - überweisen

## Beispiel Konto-Klasse

### UML-Darstellung der Kontoklasse



UML = Unified Modelling Language

## Der Klassenblock

```
public class Konto {  
    ...  
}
```

- Mit dem Schlüsselwort **class** wird die Klassendefinition eingeleitet
- **public** bedeutet, dass die Klasse überall im System zugreifbar ist.
- Alle zu der Klasse gehörigen Merkmale wie Attribute und Methoden werden zwischen die beiden Klammern eingefügt.

## Attribute definieren

```
private int kontonummer;  
private String inhaber;  
private double kontostand;
```

- **private** bedeutet, dass nur innerhalb der Klasse auf das Attribut zugegriffen werden kann
- **int** und **double** sind Namen von elementaren Datentypen. (später dazu mehr)
- **String** ist der Name einer Standardklasse von Java. Objekte dieser Klasse können Zeichenketten (engl. Strings) speichern.

## Methoden definieren

```
/**
 * Einzahlen eines Betrags
 *
 * @param betrag    Einzuzahlender Betrag
 */
public void einzahlen(double betrag) {
    kontostand = kontostand + betrag;
}
```

## Methoden definieren

- Eine Methode zum Einzahlen eines Betrages.

<b>public</b>	Die Methode kann von anderen Klassen aus aufgerufen werden. Während <b>private</b> einen Methodenaufruf nur klassenintern erlaubt, ermöglicht public den Aufruf der Methode von anderen Klassen aus.
<b>void</b>	Die Methode hat keinen Rückgabewert.
<i>einzahlen</i>	Methodenname, mit dem die Methode aufgerufen werden kann.
<b>double betrag</b>	Der übergebene Parameter vom Typ double (Fließkommazahl). Der Name des Parameters soll auf den Inhalt schließen lassen.

## Javadoc-Kommentare

- Oberhalb der Methode steht der sogenannte Javadoc-Kommentar
- Javadoc Kommentare werden eingeleitet durch **/\*\*** und enden mit **\*/**
- **@param** gibt an, welche Parameter die Methode empfängt
- **@return** gibt an, welchen Rückgabewert die Methode zurückgibt.
- Javadoc Kommentare sind enorm wichtig. Mit Hilfe des zugehörigen Tools **javadoc** können so aus dem Quelltext Dokumentationen der Programme automatisch erstellt werden.

## get-Methoden

```
public double getKontostand() {  
    return kontostand;  
}
```

- Methode zum Abfragen des Kontostandes. Da die Objektattribute private sind, müssen sogenannte get-Methoden erstellt werden, um von anderen Klassen aus den Inhalt abfragen zu können.
- Rückgabetypen
  - ⇒ **double** Der Rückgabewert der Methode ist vom Typ Double.
  - ⇒ **return** Schlüsselwort zum Zurückgeben eines Wertes.