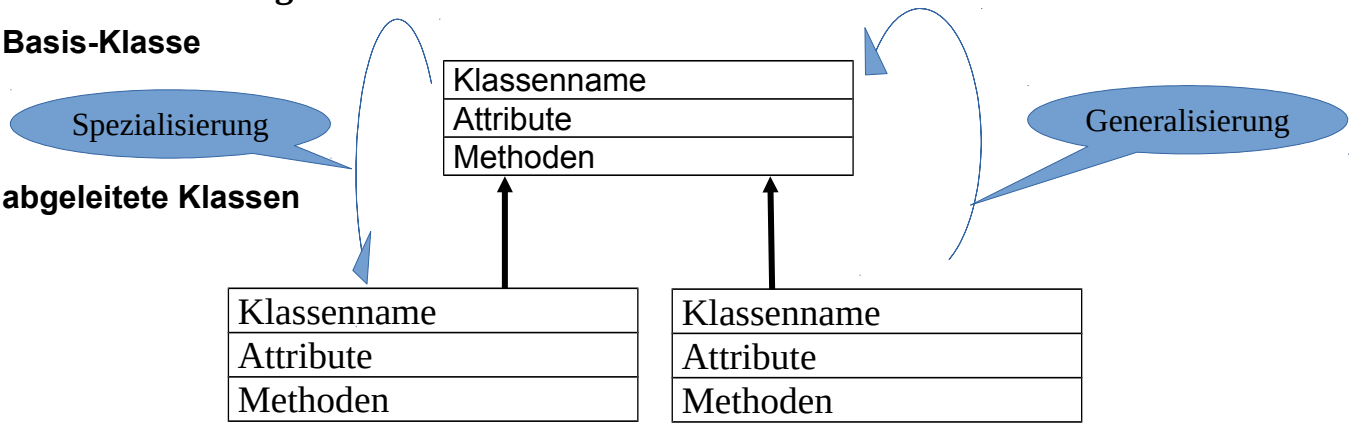
Das Prinzip der Vererbung

Eines der wichtigsten Konzepte im Zusammenhang mit Klassen ist die Möglichkeit, eine Klasse durch Vererbung von einer anderen Klasse abzuleiten. Sinn der Vererbung ist es, Klassenhierarchien aufzubauen, in denen die Gemeinsamkeiten von Klassen einer Ebene in die Klassen der vorangehenden Ebene ausgelagert werden. Klassenhierarchien zeichnen sich daher üblicherweise dadurch aus, dass die oberen Ebenen ein immer höheres Abstraktionsniveau haben, während die nachfolgenden Ebenen sich durch zunehmende Spezialisierung auszeichnen.

Einfache Vererbung

- **Basis-Klasse**
- **abgeleitete Klassen**



Es gelten folgende Regeln:

- alle Attribute werden an die abgeleiteten Klassen weitergegeben (vererbt)
- alle Methoden werden an die abgeleiteten Klassen weitergegeben (vererbt)
- Zugriffssteuerung:

Attribute oder Methoden der Basis-Klasse sind:	Funktionen der Objekte der Basis-Klasse	Funktionen der Objekte der abgeleiteten Klasse	alle Funktionen, die Objekte der Klasse kennen
public:	Zugriff möglich	Zugriff möglich	Zugriff möglich
protected:	Zugriff möglich	Zugriff möglich	Zugriff nicht möglich
private:	Zugriff möglich	Zugriff nicht möglich	Zugriff nicht möglich

- Konstruktoren / Destruktoren
  - werden an die abgeleiteten Klassen vererbt
  - jede abgeleitete Klasse kann eigene Konstruktoren und Destruktoren haben
  - Reihenfolge der Aufrufe:
    - Konstruktor der Basis-Klasse
    - Konstruktor der abgeleiteten Klasse
    - Destruktor der abgeleiteten Klasse
    - Destruktor der Basis-Klasse
  - der Standardkonstruktor der Basisklasse wird automatisch vom Compiler aufgerufen
  - reicht der Standardkonstruktor der Basisklasse nicht, um die geerbten Elemente zu initialisieren, so ruft man den entsprechenden Konstruktor der Basisklasse explizit auf: KLASSENNAME::KLASSENANME(*Parameter*):VATER-KLASSENNAME(*Parameter*) {..}

mit Datentypen

ohne Datentypen

- **Syntax in C++:**
  - **class** KLASSENNAME : **public** BASIS-KLASSENNAME
    - Bsp.:

```
class GIROKONTO : public KONTO
{
        private:
                double dGrenzbetrag;
}
```

Methoden überladen

- eine Methode der Basis-Klasse kann in der Klasse selber oder einer abgeleiteten Klasse überladen werden
- die überladene Methode muss dazu folgende Bedingungen erfüllen:
  - selber Name wie die zu überladene Methode
  - veränderte Übergabeparametertypliste

(Beispiel einer Überladung: Standardkonstruktor und weiterer Konstruktor in einer Klasse.)

Methoden überschreiben

- eine Methode der Basis-Klasse kann in einer abgeleiteten Klasse überschrieben werden
- die überschriebene Methode der abgeleiteten Klasse muss dazu folgende Bedingungen erfüllen:
  - selber Name wie in der Basis-Klasse
  - selber Rückgabetyp wie in der Basis-Klasse
  - selbe Übergabeparametertypliste wie in der Basis-Klasse

  die Methode der Basisklasse kann danach nur noch über die vollständige Qualifizierung mit ihrem Klassennamen aufgerufen werden.

(Beispiel setKontostand() wird in GIROKONTO überschrieben, da der Grenzbetrag als zusätzliches Attribut der abgeleiteten Klasse bei jeder Änderung überprüft werden muss.)

Aufgabe:
Erstellen Sie die Klassenhierarchie und Testen Sie die Zugriffsmöglichkeiten auf die einzelnen Elemente.