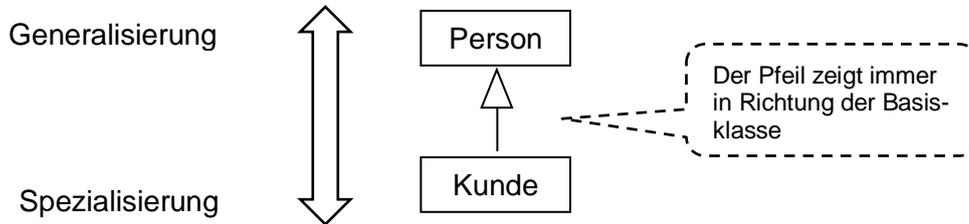


Das Konzept der **Generalisierung** (bzw. Vererbung) und Spezialisierung ist ein zentrales Thema der objektorientierten Softwareentwicklung. Situationen aus der realen Welt können besser abgebildet werden und es kann existierender Programmcode sehr gut weiterverwendet werden. Die Softwareentwicklung ist dadurch effizienter und sicherer. Die Vererbung beschreibt eine Beziehung zwischen einer allgemeinen Klasse (=Basisklasse) und einer spezialisierten Klasse.

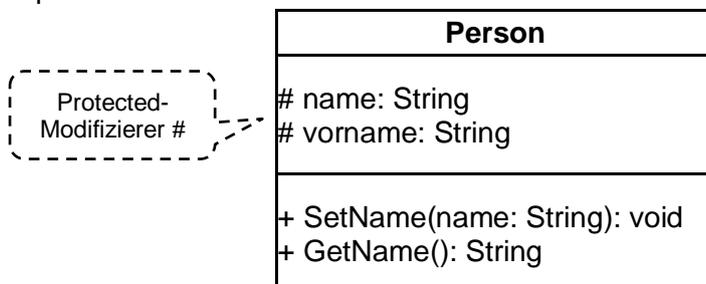
Bsp.:



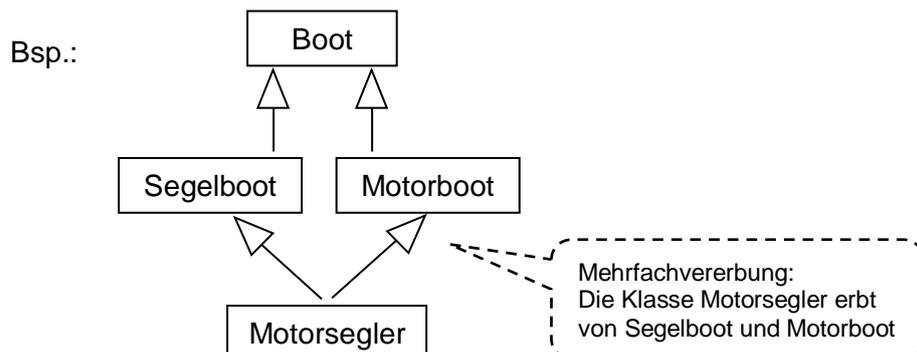
Hier: Die Basisklasse Person vererbt an die Klasse Kunde. Der Kunde ist eine Person (Ist-Beziehung). Der Kunde-Klasse stehen (mit Einschränkungen) alle Elemente der Basis-Klasse zur Verfügung. Die Klasse die vererbt (Person), wird in der Regel Basisklasse, Oberklasse oder Superklasse genannt. Die Klasse die erbt (Kunde), wird abgeleitete Klasse, Unterklasse oder Subklasse genannt.

In einer vererbten Klasse können alle öffentlichen Elemente der Basisklasse so genutzt werden als wären sie in der Klasse selbst angelegt. Der direkte Zugriff auf private Elemente ist gesperrt. Dies ist jedoch meistens umständlich. Es existiert daher ein weiterer **Zugriffsmodifizierer**. Mit *protected* versehene Attribute verhalten sich nach außen wie private, aber innerhalb der Vererbungshierarchie wie öffentliche Attribute.

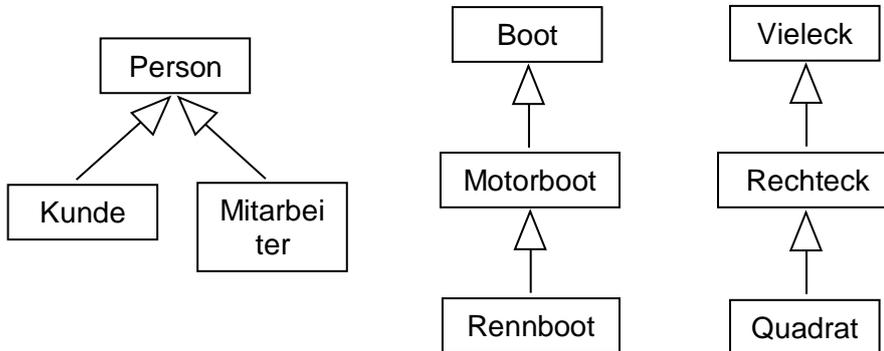
Bsp.:



Die **Mehrfachvererbung** unterscheidet sich von der einfachen dadurch, dass eine Klasse mehrere Klassen generalisiert. Ein Softwaresystem kann dadurch noch realitätsnäher umgesetzt werden, ist aber weitaus komplexer, fehleranfälliger und bedarf besonderer Berücksichtigung bei der Implementierung. In C# und Java wurde aus diesem Grund die Technik der Mehrfachvererbung nicht implementiert.



Weitere Beispiele für einfache Vererbungen:



Umsetzung der Vererbung:

```
class Unterklasse : Oberklasse { }
```

Bsp.:

```
class Person
{
    protected string name;

    public Person()
    {
    }

    public Person(string n)
    {
        this.name = n;
    }
}
```

```
class Kunde : Person
{
    private int kundenNr;

    public Kunde()
    {
        this.kundenNr = 0;
    }

    public Kunde(string name, int kNr)
    {
        this.kundenNr = kNr;
        this.name = name;
    }

    public string Ausgeben()
    {
        string zwerg;
        zwerg = this.name + this.kundenNr;
        return zwerg;
    }
}
```

Möglich aufgrund der Sichtbarkeit *protected* in der Superklasse

Aufgabe:

Erstellen sie ein UML Diagramm für folgende Begriffe und vermeiden sie Mehrfachvererbung:

Cabrio, Pritsche, BMW, LKW, Limousine, Fahrzeug, PKW, Mercedes, Kasten