

Auch wenn wir bislang noch nicht über das sprachliche Wissen verfügen, um diese Zeilen vollständig zu verstehen (der Vererbung ist mit Kapitel 9 ein eigener Teil dieses Buches gewidmet), so lässt sich doch relativ einfach nachvollziehen, was sie zu bedeuten haben:

- Die Klasse `FaxAdresse` stellt eine Subklasse von `Adresse` dar. Sie spezialisiert oder *erweitert* also die eigentliche Adresse – im Englischen erklärt dies das Schlüsselwort **`extends`**.
- Als Subklasse von `Adresse` ist eine `FaxAdresse` automatisch auch eine `Adresse`. Sie *erbt* sämtliche Eigenschaften ihrer Superklasse, sodass wir die Instanzvariablen `name` oder `hausnummer` nicht erneut definieren müssen. Sie sind dank der verwandtschaftlichen Beziehung automatisch vorhanden!
- Da der größte Teil unserer Arbeit bereits mit der Klasse `Adresse` erledigt wurde, können wir uns auf jene neuen Aspekte beschränken, die wir unserer Subklasse hinzufügen wollen. In diesem Fall bedeutet dies die Definition zweier neuer Variablen in den Zeilen 3 und 4.

Wie wir sehen, kann uns die Kombination von Generalisierung und Vererbung in der Programmierung eine Menge Schreibarbeit ersparen. Dies ist jedoch nicht der einzige Vorteil, den uns diese beiden Grundpfeiler der Objektorientierung bieten:

- Weil wir durch Vererbung gemeinsame Eigenschaften nur einmal modellieren müssen, brauchen wir diese Eigenschaften auch nur an einer Stelle zu testen. Wir haben nur eine Möglichkeit, Programmierfehler einzubauen, und somit auch nur eine Stelle, an der wir diese korrigieren müssen. Würden wir etwa den Vorgang des Milchgebens bei jedem Tier einzeln realisieren, so müssten wir jede dieser neu geschriebenen Methoden auf Fehler überprüfen.
- Algorithmen, die gewisse Spezialeigenschaften einer Subklasse nicht benötigen, können für die allgemeinere Superklasse definiert werden. Auf diese Weise können sie auch automatisch auf die verschiedensten Subklassen angewendet werden. So kann Java beispielsweise alle Objekte sortieren, die sich auf eine bestimmte Art und Weise miteinander vergleichen lassen („größer als“, „kleiner als“). Wollten wir also unsere Adressen sortieren, müssten wir lediglich dafür sorgen, dass sich unsere Objekte auf die richtige Art und Weise miteinander vergleichen lassen – den Rest erledigt eine vordefinierte Methode.

Vererbung hilft Programmierern also nicht nur, Fehler zu vermeiden. Sie erlaubt es den Entwicklern auch, sich im Laufe der Zeit ganze Bibliotheken von vorgefertigten Objekten für jeden Zweck zusammenzustellen, die sie bei Bedarf einfach um zusätzliche Funktionalität erweitern. Auf diese Weise lassen sich Softwareprodukte schneller und günstiger auf den Markt bringen als mit konventionellen Programmiersprachen.