

1

Grundbegriffe der objektorientierten Softwareentwicklung

■ 1.1 Einführung

Mit der Objektorientierung ist in den letzten Jahren ein Paradigmenwechsel in der Softwareentwicklung eingetreten, der sich von der Implementation über den Entwurf bis zur Analyse in die sehr frühen Phasen durchgesetzt hat. Die zunächst gültige Trennung von Daten und Funktionen wurde überwunden. David Parnas [1.6] propagierte Anfang der 70er Jahre die Nutzung von Datenkapseln, bei denen der Zugriff auf die Daten nur über eine Menge bereitgestellter Funktionen, die sogenannte Schnittstelle, ermöglicht wurde. Es hat eine ganze Weile gedauert, bis sich diese Idee in der Praxis der Softwareentwicklung durchgesetzt hat. Um eine Vielzahl von derartigen Datenkapseln schnell erzeugen zu können, folgte später die Idee der Programmierung von abstrakten Datentypen. Auch hier trat der Erfolg nicht sofort ein. Erst die Einordnung dieser Datentypen in eine Hierarchie, die über Vererbungsmechanismen verfügt, führte zu einem durchgreifenden Erfolg. Dieser Ansatz wurde als *objektorientiert* charakterisiert. Er ist eng mit den Begriffen von *Klasse* und *Objekt* verbunden.

Ein Objekt wird durch Eigenschaften und Fähigkeiten charakterisiert. Die Eigenschaften beschreiben den aktuellen Zustand des Objektes, und die Fähigkeiten stellen Tätigkeiten dar, die auf das Objekt angewendet werden können, um seine Eigenschaften zu verändern. So kann ein Füllfederhalter durch die Farbe der Tinte, mit der er gefüllt wurde, beschrieben werden. Diese Eigenschaft ist durch das *Entleeren* und nachfolgendes *Füllen* änderbar. Damit sind auch schon zwei Fähigkeiten genannt, die mithilfe des Füllfederhalters ausgeführt werden können. Die wichtigste Fähigkeit ist natürlich das *Schreiben*. Sie ist der Grund, warum man sich den Federhalter zulegt. Eine weitere Eigenschaft des Schreibgerätes beschreibt den Bereitschaftszustand. Es kann durch eine Schutzkappe *verschlossen* oder *unverschlossen* sein. Eine Veränderung des Bereitschaftszustandes erfolgt durch die Tätigkeiten *Öffnen* und *Schließen*, die weitere Fähigkeiten darstellen.

Bei der objektorientierten Analyse wird von den Objekten ausgegangen, die in der realen Welt existieren. Durch geeignete Abstraktion wird aus einem realen Objekt ein Objekt eines Modells. Dabei wird besonderes Augenmerk auf Charakteristika gelegt, die im Zusammenhang mit einer bestimmten Aufgabe von Interesse sind. Eine Modellierung ohne ein bestimmtes Ziel ist nicht möglich, weil die Anzahl der Charakteristika fast ins Unendliche steigt. Eigenschaften und Fähigkeiten werden durch *Attribute* und *Methoden* beschrieben.

Die Problematik der Modellierung der Charakteristika wird sicher am Beispiel deutlich. Gegenstand des Interesses sei eine Person. Da keine konkrete Zielvorgabe existiert, ist deren Modellierung praktisch nicht möglich. Ist die Haarfarbe von Interesse? Sind die Kinderkrankheiten wichtig? Welche Bedeutung haben Hobbys? Niemand kann das ohne präzisere Zusatzinformationen genau wissen. Für ein Programm zur Beratung von Farbvorschlägen für einen Friseur ist die momentane Haarfarbe der zu betreuenden Kundin von großer Bedeutung. Für die Diagnose von Erkrankungen sind die bereits durchlaufenen Kinderkrankheiten sicher wichtig. Ein System, das Literatur für einen Kunden empfehlen soll, möchte sicher auf die Hobbys der betreffenden Person zurückgreifen.

Hier sei eine Person im Kontext einer Universität exemplarisch modelliert. Zunächst wird sie als Felix identifiziert, der am 17.11.2007 geboren wurde und als Einkommen über ein Stipendium verfügt. Als besondere Fähigkeit fällt nur auf, dass er lernen und feiern kann. Diese Informationen können wie in Bild 1.1 dargestellt grafisch repräsentiert werden.

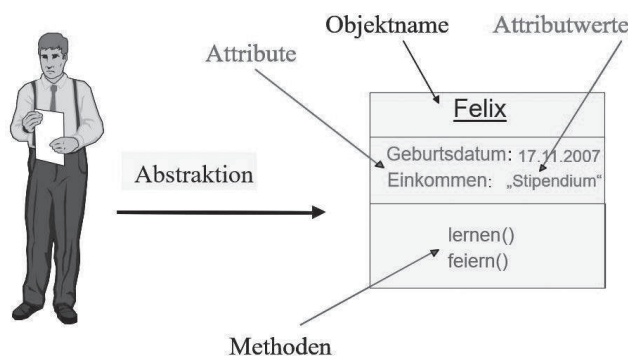


Bild 1.1 Modellierung eines Objektes

In der grafischen Repräsentation eines Objektes werden die Eigenschaften (Geburtsdatum, Einkommen) und die Fähigkeiten (lernen, feiern) getrennt dargestellt.

Fallen weitere Personen auf, die ähnliche Eigenschaften und Fähigkeiten besitzen, so sollten diese auch als Gruppe modellierbar sein. Dafür hat sich der schon in der Schule bekannte Begriff einer *Klasse*, der eine Reihe von Schülern beschreibt, die in etwa den gleichen Ausbildungsstand besitzen, eingebürgert. In diesem Zusammenhang betitelt der Begriff *Klasse* eine Sammlung von Objekten mit gleichen Charakteristika. Dabei handelt es sich um Objekte, die die gleichen Eigenschaften (Attribute) und Fähigkeiten (Methoden) besitzen.

Für obiges Beispiel weisen die Charakteristika von Felix darauf hin, dass die Klasse als Student bezeichnet werden kann. Dabei ist das Attribut entscheidend, welches das Einkommen charakterisiert. Studenten verfügen über ein Stipendium. Auch die beobachteten Fähigkeiten wie lernen und feiern stehen zu der Einschätzung nicht im Widerspruch.

Von dem Objekt Felix kann daher auf die Klasse Student abstrahiert werden. Das grafische Symbol für Objekte und Klassen ist gleich. Auch die Methoden werden identisch dargestellt. Nur an der Stelle des unterstrichenen Namens eines Objektes steht der Name einer Klasse (ohne Unterstrich). Die Darstellung der Namen von Attributen ist bei Objekten und

Klassen auch gleich. Nur haben Attribute von Objekten einen Wert, während das bei Klassen noch nicht vorliegt. Bei ihnen ist nur der Typ der Attribute bekannt (bei Programmiersprachen findet man Erweiterungen, die nicht dieser klassischen Sicht entsprechen).

Bild 1.2 zeigt die Darstellung einer Klasse, die mehrere Objekte repräsentiert.

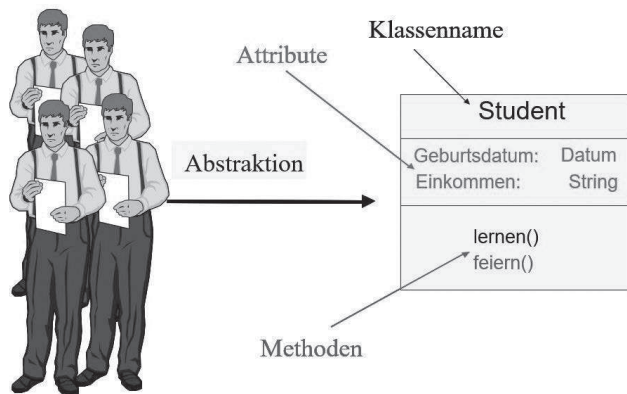


Bild 1.2 Modellierung einer Klasse

Eine recht anschauliche Abstraktion des Prozesses der Modellierung wurde von Heeg [1.4] gegeben, die hier aufgegriffen und etwas modifiziert in Bild 1.3 dargestellt wird.

Zunächst wird versucht, ein Problem der realen Welt aus einem subjektiven Blickwinkel zu modellieren. Der subjektive Blick des Modellierers wird durch das Auge symbolisiert. Er ist durch das Ziel der Modellierung beeinflusst. Dabei werden subjektiv Personen, Phänomene oder reale Dinge erfasst, die für das Modellierungsziel relevant sind. Deren Bezeichnungen werden in einem Glossar verwaltet. Dabei ist bereits auf Synonyme (unterschiedliche Bezeichnungen für gleiche Sachverhalte – z.B. Behälter, Ablage und Container) und Homonyme (gleiche Bezeichnungen für unterschiedliche Sachverhalte – z.B. Schloss) zu achten. In unbekanntenen Anwendungsbereichen ist das nicht immer ganz einfach.

Ist das Glossar ausreichend mit Informationen gefüllt, kann mit der Modellierung von Klassen begonnen werden. Zu den zunächst nur durch Begriffe charakterisierten Objekten werden Eigenschaften und Fähigkeiten ermittelt, die mit dem Modellierungsziel in Zusammenhang stehen.

Auf der Basis der modellierten Klassen findet die Entwicklung von Programmen statt, in denen Instanzen der Klassen erzeugt werden. Während der Laufzeit der Programme erfolgt die Erzeugung der entsprechenden Objekte einer Klasse, die durch Variablen repräsentiert werden. Diese Variablen ermöglichen die Beschreibung des Zugriffs auf die Objekte bereits während der Programmierung.

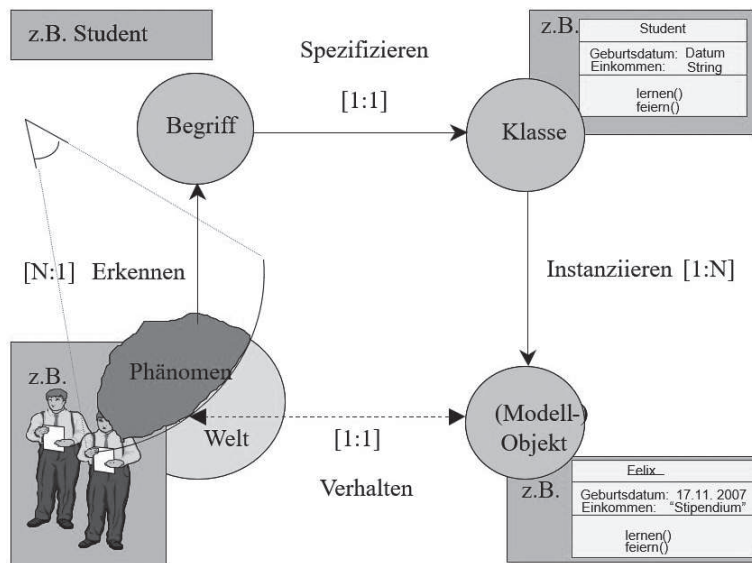


Bild 1.3 Objektorientierte Sichtweise der Softwareentwicklung

Im Allgemeinen entsprechen mehrere Objekte oder Phänomene der realen Welt einem Begriff, zu dem dann eineindeutig eine Klasse gleichen Namens existiert. Während der Laufzeit eines Programms kann aus einer Klasse eine Vielzahl von Objekten erzeugt werden. Ist die Modellierung korrekt gelungen, dann sollte das beobachtbare Verhalten der Objekte im Modellbereich dem beobachtbaren Verhalten in der realen Welt unter Einschränkung auf das Modellierungsziel exakt entsprechen. Es sollte jedem Modellobjekt eineindeutig ein Objekt aus der realen Welt zugeordnet werden können.

Genauso wie in der realen Welt kommunizieren Objekte in der Modellwelt über Botschaften.

An der Hochschule wird eventuell eine Lehrkraft einem Studenten die Botschaft lernen schicken, wenn sie der Meinung ist, dass diese Aufforderung notwendig ist. Bild 1.4 stellt diesen Sachverhalt grafisch dar. Ein Student wird entsprechend auf die Botschaft reagieren und seine Methode lernen aktivieren.

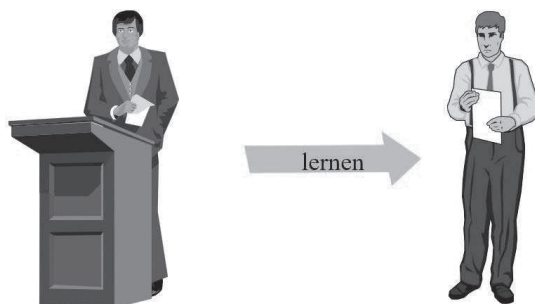


Bild 1.4 Übermittlung einer Botschaft in der Realität

Genauso sieht es auch in der Modellwelt aus (Bild 1.5). Trifft eine Botschaft bei einem Objekt ein, so wird überprüft, ob sie für das jeweilige Objekt von Bedeutung ist, ob eine Interpretation möglich ist.

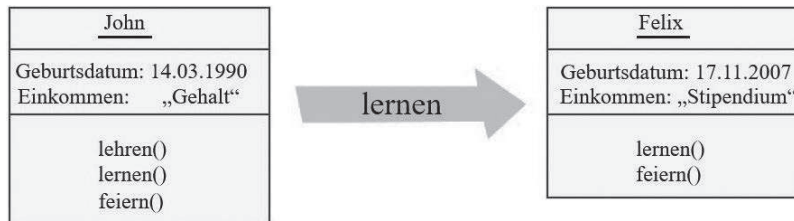


Bild 1.5 Kommunikation von Objekten in der Modellwelt

Mitunter treffen Botschaften ein, bei denen dem jeweiligen Adressaten nicht klar ist, wie er darauf reagieren soll. In der Modellwelt ist dies der Fall, wenn zu einer eintreffenden Nachricht keine gleichnamige Methode existiert.

In der Realität ist mitunter nicht so genau festzustellen, ob eine Botschaft verstanden wurde oder nicht. In der Modellwelt wird die Botschaft nicht verstanden, wenn zu einer eintreffenden Nachricht keine gleichnamige Methode existiert. In dem Falle reagiert ein Objekt nicht.



Ein Objekt reagiert nur auf eine Botschaft, wenn eine Methode gleichen Namens existiert.

Von den Programmiersprachen wird meistens gefordert, dass man nur Nachrichten an Objekte verschickt, auf die der Adressat auch reagieren kann. Es wird bei der Analyse von Programmen bereits überprüft, ob die Versendung von Nachrichten sinnvoll ist. Ein Compiler signalisiert meist schon einen Fehler. Das gilt für Sprachen wie Pascal, Java, C++ oder C#. Bei der Nutzung eines Interpreters kommt es erst zur Laufzeit der Programme zu Fehlermeldungen. Das kann man beispielsweise bei der Nutzung der Entwicklungsumgebung der Sprache Python beobachten. Es wird also nicht einfach ignoriert, dass die Nachricht nicht verstanden wird.



Programmiersprachen fordern, dass Objekte zu jeder eintreffenden Botschaft eine entsprechende Methode besitzen.

Es gibt damit einen Unterschied zwischen der Realität und den in Programmiersprachen spezifizierten Modellwelten.

Nicht alle Studenten haben im Detail das gleiche Verhalten. Man kann sie beispielsweise nach ihren Studiendisziplinen klassifizieren, die unterschiedliche Fähigkeiten fordern.

Die allgemeinen Eigenschaften und Fähigkeiten eines Studenten haben alle Studierenden, sie besitzen aber mitunter noch weitere Fähigkeiten wie Singen oder Tanzen. Was diese Fähigkeiten im Einzelnen sind, soll hier zunächst erst einmal nicht interessieren. Bild 1.6 stellt grafisch dar, dass es verschiedene spezielle Klassen gibt. Sie haben die Eigenschaften