

HANSER

# Objektorientierte Softwareentwicklung mit UML

Peter Forbrig

ISBN 3-446-40572-0

Leseprobe

Weitere Informationen oder Bestellungen unter  
<http://www.hanser.de/3-446-40572-0> sowie im Buchhandel

# 1 Grundbegriffe der objektorientierten Softwareentwicklung

## 1.1 Einführung

Mit der Objektorientierung ist in den letzten Jahren ein Paradigmawechsel 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 /1.5/ propagierte Anfang der 70er Jahre die Nutzung von Datenkapseln, bei denen der Zugriff auf die Daten nur über eine Menge bereitgestellter Funktionen, die so genannte 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.

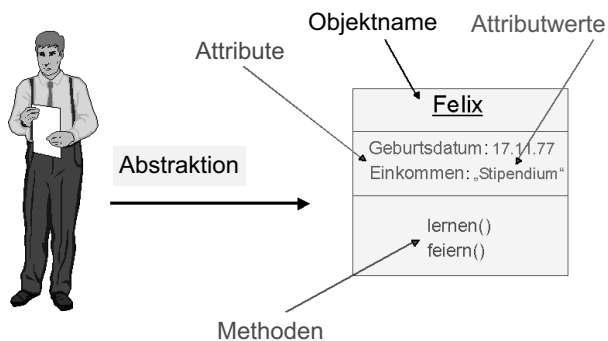
Ein Objekt wird durch Eigenschaften und Fähigkeiten charakterisiert. Die Eigenschaften beschreiben den aktuellen Zustand des Objektes und die Fähigkeiten stellen Tätigkeiten dar, die auf das Objekt angewendet werden können, um seine Eigenschaften zu verändern. So kann ein Füllfederhalter durch die Farbe der Tinte, mit der er gefüllt wurde, beschrieben werden. Diese Eigenschaft ist durch *Entleeren* und nachfolgendes *Füllen* änderbar. Damit sind auch schon zwei Fähigkeiten genannt, die mit Hilfe des Füllfederhalters ausgeführt werden können. Die wichtigste Fähigkeit ist natürlich *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 modelliert.

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. 1977 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 **Abbildung 1.1** dargestellt grafisch repräsentiert werden.



**Abbildung 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. Der schon in der Schule bekannte Begriff einer *Klasse*, der eine Reihe von Schülern beschreibt, die in etwa den gleichen Ausbildungsstand besitzen, hat sich dafür 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, welches das Einkommen charakterisiert, entscheidend. Studenten verfügen über ein Stipendium. Auch die beobachteten Fähigkeiten wie *Lernen* und *Feiern* stehen zu der Einschätzung nicht im Widerspruch.

Eine recht anschauliche Abstraktion des Prozesses der Modellierung wurde von Heeg /1.3/ gegeben, die hier aufgegriffen und etwas modifiziert in **Abbildung 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. 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.

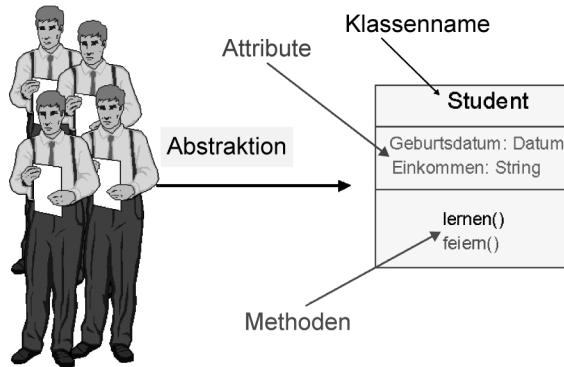


Abbildung 1.2 Modellierung einer Klasse

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.

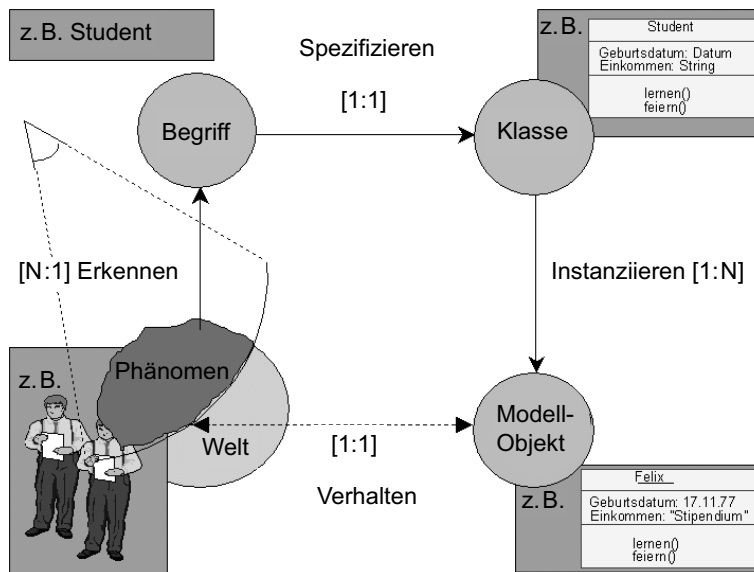


Abbildung 1.3 Objektorientierte Sichtweise der Softwareentwicklung

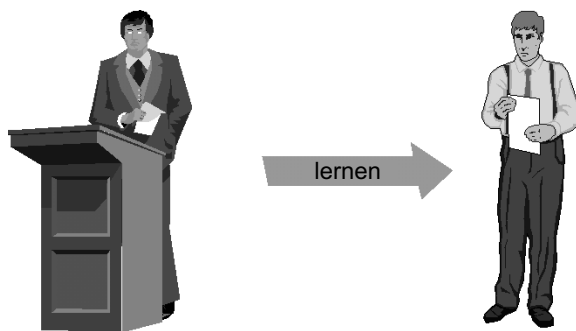
Im Allgemeinen entsprechen mehrere Objekte oder Phänomene der realen Welt einem Begriff, zu dem dann eineindeutig eine Klasse gleichen Namens existiert. Während der Laufzeit eines Programms kann aus einer Klasse eine Vielzahl von Objekten erzeugt werden. Ist die Modellierung korrekt gelungen, dann sollte 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.

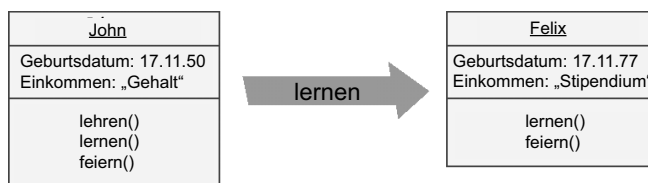
An der Hochschule wird eventuell eine Lehrkraft einen Studenten die Botschaft *lernen* schicken, wenn sie der Meinung ist, dass diese Aufforderung notwendig ist. **Abbildung 1.4** stellt diesen Sachverhalt grafisch dar. Ein Student wird entsprechend auf die Botschaft reagieren und seine Methode *lernen*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

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

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.



**Abbildung 1.4** Übermittlung einer Botschaft in der Realität



**Abbildung 1.5** Kommunikation von Objekten in der Modellwelt

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.

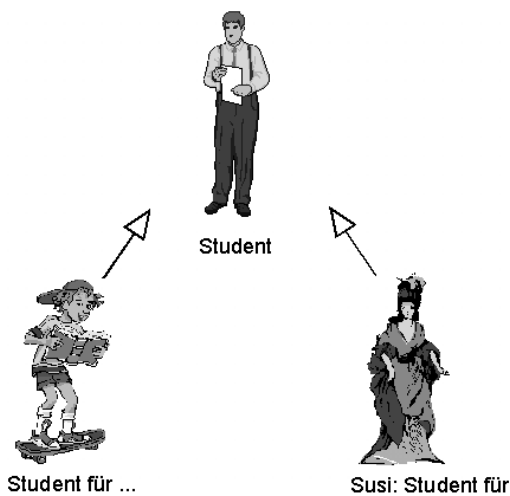
Von den Programmiersprachen wird meistens gefordert, dass man nur Nachrichten an Objekte verschickt, auf die der Adressat auch reagieren kann. Es wird bei der Analyse von Programmen bereits überprüft, ob die Versendung von Nachrichten sinnvoll ist. Ein Compiler signalisiert meist schon einen Fehler. 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.

Zurückkommend zu dem Beispiel eines Studenten fällt auf, dass es in der Realität aber nicht nur eine Art von Student gibt, sondern man kann Studenten in verschiedene Gruppen unterteilen. Eine Möglichkeit ist die Klassifizierung entsprechend ihrer Studiendisziplinen.



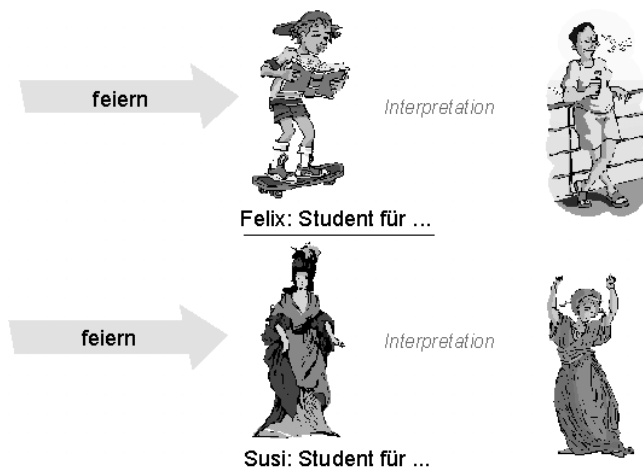
**Abbildung 1.6** Weiterleitung (Vererbung) von Eigenschaften und Fähigkeiten

Die allgemeinen Eigenschaften und Fähigkeiten eines Studenten haben sie alle, sie besitzen aber mitunter noch weitere Fähigkeiten, wie Singen oder Tanzen. Was diese Fähigkeiten im Einzelnen sind, soll hier zunächst erst einmal nicht interessieren. **Abbildung 1.6**

stellt grafisch dar, dass es verschiedene spezielle Klassen gibt. Sie haben die Eigenschaften einer allgemeinen Klasse *Student*. Man spricht auch davon, dass sie diese Charakteristika erben. (Genauer wird dazu später noch unter Definition 1.9 erläutert.)

Die beiden etwas exotischer anmutenden Klassen von *Student für ...* haben die gleichen Attribute und Methoden wie die Klasse *Student*. Alles, was ein Student kann, das können sie auch. Sie haben sogar noch zusätzliche Eigenschaften und/oder Fertigkeiten. Alle sind also in der Lage, auf die Botschaft *feiern* zu reagieren. Das bedeutet nun aber nicht unbedingt die Ausführung des gleichen Algorithmus. Jede Gruppe (Klasse) von Studenten hat ihre eigene Methodenimplementierung, ihren eigenen Algorithmus zu feiern.

Die Auslösung verschiedener Algorithmen durch die gleiche Botschaft wird auch als Polymorphismus (Definition 1.12) bezeichnet. **Abbildung 1.7** versucht eine Visualisierung dieses Sachverhaltes.



**Abbildung 1.7** Unterschiedliche Reaktion auf Botschaften – Polymorphismus

## 1.2 Konzepte und Notationen

Dieser Abschnitt beschäftigt sich mit einigen grundlegenden Konzepten und deren Notation. Späteren Kapiteln ist die Erläuterung weiterer Details vorbehalten. Hier soll zunächst ein Grundverständnis von Basismodell, statischem Modell, dynamischem Modell und dem Modell der Systemnutzung gegeben werden.

### 1.2.1 Basismodell

Zum Verständnis des Basismodells der objektorientierten Softwareentwicklung gehört die richtige Vorstellung von den Begriffen Objekt, Attribut, Methode und Klasse. Im vorigen Abschnitt wurden diese Ausdrücke bereits intuitiv benutzt. Hier sollen sie teils unter Ver-

wendung von Zitaten etwas ausführlicher diskutiert werden. In der Literatur gibt es zahlreiche Nuancen der Definition des Begriffs Objekt. Wir wollen auf die Definition von Balzert /1.1/ zurückgreifen.

### Definition 1.1 Objekt

Ein Objekt ist allgemein ein Gegenstand des Interesses, insbesondere einer Beobachtung, Untersuchung oder Messung. Objekte können Dinge und Begriffe sein.

In der objektorientierten Softwareentwicklung besitzt ein Objekt bestimmte Eigenschaften und reagiert mit einem definierten Verhalten auf seine Umgebung. Außerdem besitzt jedes Objekt eine Identität, die es von allen anderen Objekten unterscheidet.

Die Eigenschaften eines Objektes werden durch dessen Attributwerte ausgedrückt, sein Verhalten durch eine Menge von Methoden.

Das Verhalten eines Objektes kann mit anderen Worten auch als die Menge seiner Fähigkeiten bezeichnet werden. Es ist in der Lage, bestimmte Aktivitäten auszuführen. Für das Eingangsbeispiel (Abbildung 1.1) ist der Student *Felix* ein Objekt, dessen Verhalten durch die Tätigkeiten des *Lernens* und des *Feierns* bestimmt ist. Außerdem besitzt es die Attribute *Geburtsdatum* und *Einkommen*, die seine Eigenschaften charakterisieren und deren Attributwerte „17. 11. 1977“ und „Stipendium“ sind.

Ein Objekt gehört zu einer Klasse gleichartiger Objekte, die gleiche Attribute und Methoden besitzen.

### Definition 1.2 Klasse

Eine Klasse beschreibt eine Sammlung von Objekten mit gleichen Eigenschaften (Attributen), gemeinsamer Funktionalität (Methoden), gemeinsamen Beziehungen zu anderen Objekten und gemeinsamer Semantik.

Der Klassenname ist nach Coad /1.2/ stets ein Substantiv im Singular, das durch ein Adjektiv ergänzt werden kann.

Die Eigenschaften eines Objektes, die durch Werte charakterisierbar sind, werden in Klassen in Form von Attributen modelliert.

### Definition 1.3 Attribut

Die Attribute beschreiben die Daten bzw. Eigenschaften einer Klasse. Alle Objekte einer Klasse besitzen dieselben Attribute, jedoch unterschiedliche Attributwerte. Das bedeutet für die Implementation, dass jedes Objekt Speicherplatz für alle seine Attribute erhalten muss.

Der Attributname muss im Kontext eines Objektes eindeutig sein. Im Allgemeinen wird ein Substantiv dafür verwendet.

Eine Methode ist eine Tätigkeit, die ein Objekt ausführen kann. Sie wird durch ein Verb oder durch ein Substantiv mit Verb bezeichnet und muss im Kontext eines Objektes eindeutig sein. Das Verhalten eines Objektes wird durch eine Menge von Methoden charakterisiert.

Beispielsweise hat der Student Felix die Fähigkeiten (Methoden) zu *feiern* und zu *lernen*. Die Definition der Algorithmen, die mit den Methoden verbunden sind, erfolgt durch die jeweilige Klassen. Im Falle von *Felix* ist das die Klasse *Student*.

#### Definition 1.4 Methode

Eine Methode ist ein Algorithmus, der einem Objekt zugeordnet ist und von diesem abgearbeitet werden kann.

Objekte kommunizieren untereinander über Nachrichten, die Botschaften genannt werden. Wird an ein Objekt eine Botschaft geschickt, so führt dies zum Aufruf der entsprechenden Methode. Im Eingangsbeispiel hat die Lehrkraft *John* dem Studenten *Felix* die Botschaft *lernen* gesandt.

#### Definition 1.5 Botschaft

Eine Botschaft ist eine Nachricht, die den Aufruf einer Methode gleichen Namens zur Folge hat.

Auf die Botschaft (bitte) *lernen* führt ein freundlicher Felix in der Realität die entsprechenden Aktivitäten durch. In der Modellwelt würde ein Objekt *Felix* mit der Ausführung der gleichnamigen Methode reagieren.

Nach absolviertem Lernen sendet Felix eine Vollzugsmeldung an seinen Lehrer.

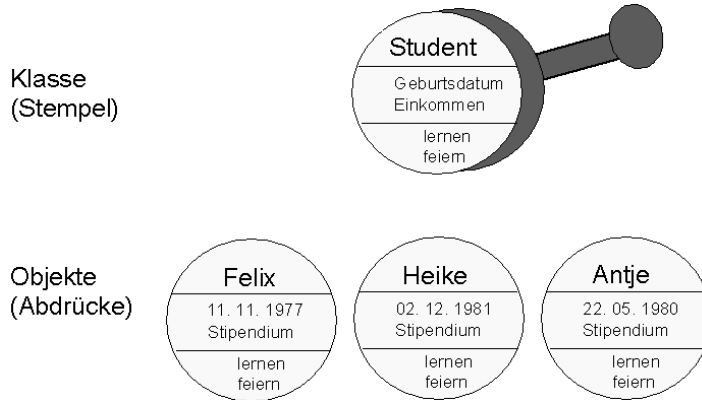
In /1.1/ wird zur Verdeutlichung der Beziehung zwischen Klasse und Objekten die Metapher des Stempels herangezogen. Die Metapher stimmt nicht vollständig, gibt aber einen ganz anschaulichen Eindruck. Die Besonderheit dieses Stempels ist, dass nicht nur eine vollständige Kopie der Attribute und Methoden erzeugt wird, sondern dass die Attribute mit Attributwerten versehen werden können.

Der Stempel entspricht einer Klasse, die unterschiedliche Exemplare von Objekten erzeugen kann. Die Stempelabdrücke symbolisieren die Objekte. Das Problem bei der Metapher besteht darin, dass die Abdrücke teilweise andere Informationen besitzen als der Stempel. Ein Attributwert für das Geburtsdatum ist in der Klasse *Student* noch nicht enthalten. Dort steht nur der Typ dieses Attributes. Die Objekte *Felix*, *Antje* und *Heike* haben jeweils aber einen solchen Attributwert.

Trotz dieses kleinen Fehlers ist in **Abbildung 1.8** die Beziehung zwischen Klasse und Objekt gut sichtbar und verdeutlicht, dass Objekte ihre Eigenschaften nicht direkt durch Vererbung erhalten. Sie profitieren nur indirekt davon. Um sich diesen Sachverhalt ständig vor Augen zu führen, ist die Stempelmetapher sehr gut geeignet.

Eine Klasse dient zur Erzeugung von Objekten. Sie prägt sie im gewissen Sinne. In der Umgangssprache wird der Klassenbegriff mehr zur Gruppierung von Objekten genutzt. In der objektorientierten Welt dient eine Klasse hauptsächlich zur Erzeugung von Objekten.

Die erzeugten Objekte nennt man auch Instanzen. Der Prozess ihrer Generierung wird dementsprechend als Instanziierung bezeichnet.



**Abbildung 1.8** Stempelmetapher für Klassen

Damit sind zunächst die wichtigsten Begriffe des Basismodells des objektorientierten Paradigmas dargestellt worden.

Die folgenden Abschnitte befassen sich mit weiteren grundsätzlichen Konzepten, die in Form spezieller Modelle ausgedrückt werden. Dabei handelt es sich um das statische Modell, das dynamische Modell und das Modell der Systemnutzung.

Das statische Modell beschreibt Beziehungen zwischen Modellelementen, die strukturelle Zusammenhänge aufweisen. Das Verhalten der einzelnen Modellelemente wird durch ein dynamisches Modell beschrieben. Der Aspekt der Systemnutzung ist im Fokus des letzten Modells, das auch den entsprechenden Namen trägt.

**Abbildung 1.9** gibt eine grafische Veranschaulichung, wie diese Modelle zusammenhängen. Sie zeigt auch, wie sich geschichtlich die Sicht auf objektorientierte Softwarespezifikationen erweitert hat.

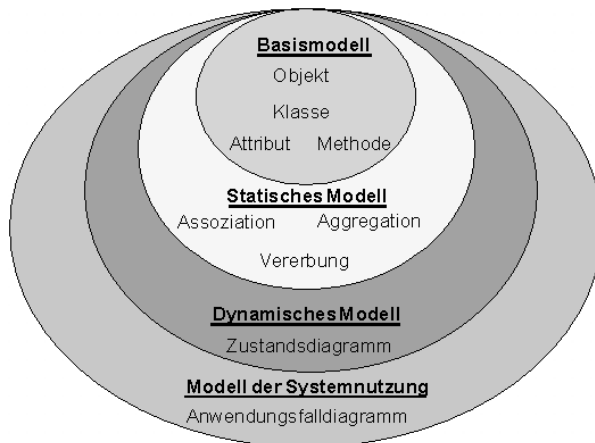
Das Basismodell entspricht der Idee der Datenkapselung und der Idee der Programmierung mit Hilfe von abstrakten Datentypen. Dieser Gedanke wurde von der Implementation bis in die frühen Phasen der Softwareentwicklung verbreitet. Dazu kam die Idee der Nutzung von Assoziation und Vererbung, die aus dem Bereich der Datenmodellierung in die allgemeine Softwareentwicklung Einzug hielten. Danach wurde der Aspekt der Spezifikation der Dynamik von Objekten genauer untersucht.

Dabei erhielt das Konzept der endlichen Automaten ein neues Anwendungsfeld. Ursprünglich zur Spezifikation mathematischer Systeme eingeführt, wurde es bereits bei der Modellierung von Dialogabläufen und bei der Spezifikation von Reihenfolgen bei der „Strukturierten Analyse“ genutzt. Das Konzept hielt speziell durch die Arbeiten von Harel [1.7/

verstärkt Einzug in die Modellierung allgemeiner dynamischer Systeme. Später wurde es zur Spezifikation der Dynamik von Klassen herangezogen.

Jacobson hat es durch seine Arbeiten verstanden, den Aspekt der Systemnutzung mehr in den Vordergrund zu stellen und als Ausgangsbasis für die Systemmodellierung zu nutzen. Seine Vorgehensweise integriert einige Ideen aus der aufgabenorientierten Softwareentwicklung, die auch als strukturierter Ansatz bezeichnet werden, mit der objektorientierten Modellierung.

Nicht nur die Objekte sind von Interesse, sondern die Aufgaben, die mit Hilfe eines Systems zu erfüllen sind, stehen im Mittelpunkt. Die von Jacobson entwickelte, auf Anwendungsfälle aufbauende Methodik erfreut sich großer Beliebtheit.



**Abbildung 1.9** Entwicklung des objektorientierten Modellverständnisses

Angelehnt an die Struktur der Modelle in **Abbildung 1.9** werden auch die folgenden einführenden Abschnitte zu Methoden und Konzepten gegliedert. Nach dem Basismodell folgen nun noch das statische Modell, das dynamische Modell und das Modell der Systemnutzung.

### Aufgaben



- 1.1 Erläutern Sie, warum die Stempelmetapher nicht vollständig das Zusammenspiel zwischen Klasse und Objekt widerspiegelt.
- 1.2 Gibt es Objekte, die zu keiner Klasse gehören?
- 1.3 Welcher Zusammenhang besteht zwischen einer Botschaft und einer Methode?
- 1.4 Was versteht man unter der Instanz einer Klasse?