

Universität Innsbruck

Seminararbeit

über das Thema

Paradigmen der objektorientierten Programmierung

Karl Greuter
karl.greuter@student.uibk.ac.at

Einleitung

Diese Arbeit beschreibt den Begriff Paradigma und versucht diese im Zusammenhang mit der objektorientierten Programmierung zu setzen. Sie ist unterteilt in drei Kapitel, wobei sich die ersten zwei mit den Begriffen Paradigma und Objektorientierung beschäftigen. Im dritten Teil wird näher auf die einzelnen Konzepte eingegangen.

1 Was ist ein Paradigma?

„Ein Paradigma ist ein symbolisches Modell oder Diagramm, das es uns erleichtert, die wesentlichen Merkmale eines Prozesses zu verstehen.“ (Wolf,1995,S. 266)

Ein Paradigma hat unterschiedliche Bedeutungen für einzelne Bereiche der Wissenschaft. Beispiele dafür sind Paradigmen der Psychologie oder der Wirtschaft. Im Bereich der Programmierung sind Paradigmen verschiedene Prinzipien die den einzelnen Sprachen zugrunde liegen.

2 Was ist Objektorientierung im Bereich Programmierung?

Die Suche nach dem Ausgangspunkt der Programmiersprachen führt uns zum sequenziellen Automat und zur Turingmaschine. Diese Ursprünge bilden sich in der tatsächlichen Funktionsweise, dem anfänglichen Programmierkonzept und der Maschinensprache ab. Es ist zu verfolgen, dass eine zunehmende Abstraktion vom wirklichen Funktionsprinzip der Maschine auf immer höherem Niveau stattfindet. Von Maschinensprache im Binärcode über die Assemblersprache wurde zunehmend abstrahiert zu Hochsprachen, mit leistungsfähigeren Befehlen und komplexeren Grammatiken. Diese Schritte beinhalten die Entfernung vom Konzept der programmierbaren Maschine (Eingabe, Ausgabe). Die im Rechner abgebildeten Programme sollen nicht mehr den funktionsprinzipien eines Digitalrechners folgen, sondern in Struktur und Verhalten einem abgebildeten Realitätsausschnitt entsprechen. Das beschreibt der Begriff Objektorientierung.

3 Paradigmen der objektorientierten Programmierung

Als Basiskonzepte der Objektorientierung gelten im Wesentlichen:

Abstraktion, Objekt, Klasse, Generalisierung, Kapselung, Polymorphie, Assoziation, Aggregation, Komposition.

Es wird versucht den Ausgangspunkt in der Betrachtung der Realität sowie der menschlichen Art und Weise zu denken und zu sprechen nachzuempfinden. Objektorientierte Programmiersprachen zeichnen sich aus, für diese abstrakten Konzepte Ausdrucksmittel anzubieten.

Die klassische, imperative Programmieretechnik betrachtet Daten und Funktionen als voneinander getrennt. Die objektorientierte Sichtweise überwindet diese Einstellung und führt dadurch zur Verschmelzung von Daten und ihre Verarbeitung miteinander.

3.1 Abstraktion

Computerprogramme sollen Sachverhalte der realen Welt abbilden, um Menschen die Arbeit zu erleichtern oder auf irgendeine Art dienstbar zu sein. Die Beschreibung der realen Welt bezieht sich auf die menschliche Sprache. Um einen Computer einen bestimmten Sachverhalt als Programmlogik zu vermitteln, soll sich dieser Prinzipien bedient werden. Ein wahrgenommener Sachverhalt muss in Worte gefasst werden, um diesen zu verstehen und zu beschreiben. Die Begriffsbildung ist der Schlüssel zum Verständnis der Abstraktion. Zum Beispiel der Begriff „Auto“ ist abstrakt. Jeder Mensch erlernt den Begriff „Auto“ indem er es sieht. Er kann ein Auto sofort erkennen obwohl es verschiedene Farben hat und es in den einzelnen Fällen anders aussieht. Wir können über Autos reden, obwohl kein Auto da ist. Voraussetzung ist, dass wir den Begriff erlernt und verstanden haben. Im menschlichen Denken liegt eine Verknüpfung von allgemeingültigen Merkmalen, man kann sagen Bauplänen und Begriffen, vor. In der Sprache und Begriffsbildung entsteht die menschliche Fähigkeit zur Abstraktion.

3.2 Objekt

Beim erfassen der Realität sehen wir Sachverhalte mit verschiedenen Merkmalen. Zum Beispiel ein Auto hat verschiedene Farben, wobei eine Farbe ein Merkmal dieses speziellen Sachverhaltes beziehungsweise Objektes ist. In der objektorientierten Programmierung sind diese Objekte Datenstrukturen. Wobei jedes Objekt gewisse Merkmale hat und eigene Methoden zur Verarbeitung der Daten beinhaltet.

3.3 Klassen

Der Mensch ist in der Lage, den wahrgenommenen Merkmalen ein Wort zuzuordnen und dieses in Form von Sprache weiter zu geben. Das Wahrgenommene sind in der Summe die Struktur- und Verhaltensmerkmale der Dinge, also deren Baupläne. Klassen sind Baupläne für Objekte. Jedes Objekt gehört einer Klasse an. Klassen definieren die Verhaltensweisen von Objekten sowie deren Attributen.¹ Man nennt ein Objekt eine Instanz einer Klasse und eine Klasse entsprechend eine Abstraktion eines Objektes.

3.4 Vererbung

In der realen Welt denkt man in Hierarchien wie zum Beispiel Familie, Eltern, Kinder, Hund. Dasselbe Prinzip wird bei der Vererbung angewandt. Es werden ausgewählte Klassen mit gemeinsamen Merkmalen, in der übergeordneten Klasse zusammengefasst, genau wie im Beispiel Familie. Das Zusammenfassen nennt man auch Generalisierung. Es entstehen Klassenhierarchien. Dabei werden meist übergeordnete, abstrakte Klassen verwendet wie in Abbildung 1 zu sehen

¹Attribute sind die Merkmale der Objekte

ist. Somit ist die Wiederverwendung von allgemeinen Merkmalen gleich wie im menschlichen Denken sehr effizient gelöst.

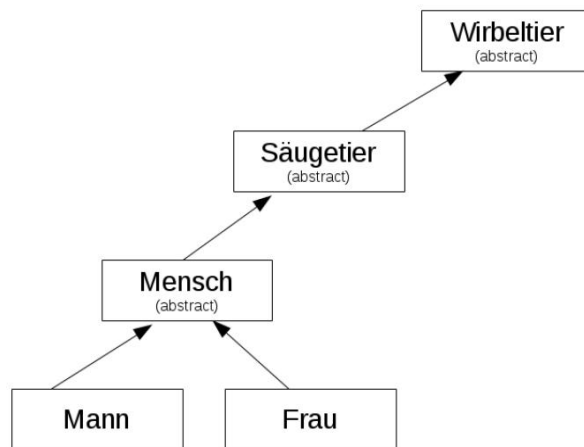


Abbildung 1: Vererbung

3.4.1 Einfachvererbung

Die Einfachvererbung beschreibt den Vorgang der Manipulation einer einzelnen Klasse durch Hinzufügen von neuen oder Sperren von bestehenden Methoden.² Durch die einfache Vererbung entsteht aus einer Klasse eine Unterklasse.

3.4.2 Mehrfachvererbung

Die Mehrfachvererbung beschreibt den Vorgang der Manipulation mehrerer Klassen durch Kombination, Ergänzen oder Sperren von Methoden. In diesem Fall hat eine Unterklasse mehrere Oberklassen.

3.5 Kapselung

Die Kapselung bezieht sich auf das Geheimprinzip. Jeder hat seine eigenen Geheimnisse und will sie in einem bestimmten Kreis nicht Preis geben. Ein Beispiel ist die Familie weiß mehr persönliche Dinge über mich, als mein Arbeitskollege. Mit der Kapselung ist es möglich genau dies zu bewerkstelligen. Die exakte Realisierung der Methode sowie für die Abarbeitung nötige und im Objekt abgelegte Daten dürfen nicht nach außen hin nicht in Erscheinung treten. In der Objektorientierung unterscheidet man derzeit folgende Sichtbarkeiten“:

- public, ein für alle Objekte im System sichtbares Merkmal
- package, ein für alle Objekte im selben Raum sichtbares Merkmal
- protected, sichtbar für alle Objekte dieser Klasse und deren Ableitungen
- private, nur sichtbar für alle Objekte dieser Klasse

²Eine Methode verarbeitet Daten einer Klasse

3.6 Polymorphie

Polymorphie steht für Vielgestaltigkeit. Das heißt jedes Objekt bringt seine Logik selbst mit und entscheidet für sich selbst. Die Programmlogik ist nicht der Befehlsgeber, sondern der Moderator. Wenn an eine Menge von Objekten einer Klasse ein und dieselbe Nachricht gesendet wird, reagieren diese mit unterschiedlichem Verhalten. Es werden dabei gleichnamige Methodenbezeichner bei unterschiedlicher oder gleicher Funktionalität verwendet.

3.7 Assoziation

Systeme bestehen aus Objekten, die zusammenarbeiten. Objekte arbeiten über den Austausch von Nachrichten zusammen. Dafür müssen sie sich kennen. Diesen Umstand nennt man Assoziation. Die Objekte eines Systems stehen in Beziehung zueinander. In der Praxis heißt das man muss den Instanznamen für das Objekt kennen um ihm Nachrichten senden zu können. Die Objekte kennen sich zwar, sind aber nicht für einander verantwortlich.

3.8 Aggregation

Systeme bestehen aus Objekten. Das System selbst ist auch ein Objekt und es ist sinnvoll, komplexe Systemteile als Komponenten zusammenzufassen, die wiederum Objekte sind, welche aus Objekten bestehen. Diese Ganz-Teil-Beziehungen bezeichnet man als Aggregation. Dabei ist das Ganze jeweils für seine Teile verantwortlich. Die Aggregation ist eine Erweiterung der Assoziation.

3.9 Komposition

Die Komposition beschreibt eine verschärfte Form der Aggregation. Bei einer Komposition stehen das Ganze und das Teil in einer existenziellen Abhängigkeit. Dieses Konzept wird in aktuellen Sprachen nur gering unterstützt.

4 Zusammenfassung

Was sind nun Paradigmen der objektorientierten Programmierung?

Paradigmen sind Konzepte oder Prinzipien für ein bestimmten Bereich, in unseren Fall für die Programmierung. Die objektorientierte Programmierung besteht aus abstrakten Konzepten, welche der menschlichen Denkweise nachempfunden ist, um das Leben der Programmierer für bestimmte Aufgaben leichter zu machen. Hier noch ein kleiner Vergleich aus unserer Sprache und der Programmierung:

Wenn der Taster gedrückt ist schalte die LED an.

If the button is pressed the LED will turn on.

```
if button.isPressed then led.on  
if(button.isPressed()) led.on();
```

Lange Rede kurzer Sinn. Unsere natürliche Sprache ist objektorientiert.