

1 Grundideen der objektorientierten Softwareentwicklung

Die objektorientierte Softwareentwicklung ist ein Vorgehensmodell oder Methode, bei der die Ergebnisse der Phasen Analyse, Definition, Entwurf und Implementierung objektorientiert erstellt werden.

Im Gegensatz zu anderen Vorgehensmodellen ist dadurch eine Durchgängigkeit in der Benennung der Konzepte gewährleistet, in Analyse und Entwurf wird sogar dieselbe Notation (UML) eingesetzt. Es entsteht kein Strukturbruch.

- Die reale Welt wird der Software mit Hilfe von **Objekten** abgebildet



Jedes Objekt hat **Eigenschaften** und **Verhaltensweisen**: Eigenschaften werden als Attribute, Verhaltensweisen als Operationen oder Methoden bezeichnet. Im Unterschied zur strukturierten Programmierung werden also Daten (Variablen) und Methoden in *einem* Objekt zusammengefasst und nicht getrennt voneinander verwaltet.

Alle Objekte der realen Welt haben

- **Eigenschaften**, die ihren momentanen Zustand beschreiben
- bestimmte **Verhaltensweisen**, was sich durch ihr mögliches Tun ausdrückt.



Farbe: Rot
Km-Stand: 14.321
Preis: 49.900
AktGeschwindigkeit: 0

Bremsen
Starten
BeschleunigenUm (..)
...

- Durch diese **Zusammenfassung von Daten und Methoden** in Objekten werden **Robustheit**, **Korrektheit**, **Erweiterbarkeit** und **Wiederverwendbarkeit** von Programmen entscheidend verbessert.
- **Wiederverwendbarkeit** von Modulen: ist leichter als mit prozeduralen Programmiersprachen, da Objekte bzgl. ihrer Daten und Methoden klar definiert sind, die Interna dem Anwender aber nicht bekannt sein müssen (=Datenkapselung, „Black-Box-Prinzip“)
- **Datenkapselung**: Dadurch werden inkompetente Zugriffe und damit viele Fehlermöglichkeiten, z.B. durch unbeabsichtigtes Ändern von Variablen in Folge von Nebenwirkungen, ausgeschlossen.
- **Vererbung**: die Eigenschaften und Methoden bestehender Objekte können an davon abgeleitete Objekte weitergegeben werden

2 Grundbegriffe

Objekt (Instanz)	<p>= konkretes individuelles Exemplar¹ (engl. Instance) von Dingen (z.B. ein bestimmtes Auto oder Haus), Personen (z.B. ein bestimmter Kunde oder Lieferant) oder Begriffen (z.B. eine bestimmte Bestellung, ein bestimmter String) der realen oder der Vorstellungswelt.</p> <p>Ein Objekt besitzt einen → Zustand und reagiert mit einem bestimmten → Verhalten auf seine Umwelt.</p> <p>Jedes Objekt besitzt eine Objekt-Identität, die es von allen anderen Objekten unterscheidet.</p>
Klasse	<p>= Gruppe von Dingen, Lebewesen oder Begriffen mit gemeinsamen Merkmalen. D.h. eine Menge von Objekten mit gleichen Attributen und gleichen Operationen (Methoden) und gleichen Beziehungen (Assoziationen, Vererbungsbeziehungen)</p> <p>Eine Klassenbeschreibung dient als Schablone, die angibt, wie ein Objekt einer Klasse aussehen soll.</p> <p>Eine Klasse besitzt einen Mechanismus um Objekte zu erzeugen.</p> <p>Der Klassenname ist ein Substantiv im Singular und beginnt mit einem Großbuchstaben.</p>
Attribut	<p>Die Attribute beschreiben die Daten, die von den Objekten einer Klasse angenommen werden können.</p> <p>Jedes Attribut ist von einem bestimmten Typ.</p> <p>Alle Objekte einer Klasse besitzen dieselben Attribute, jedoch individuelle Attributwerte.</p>
Zustand	<p>Der Zustand eines Objekts wird durch seine → Attribute und deren Attributwerte bzw. Daten bestimmt.</p>
Verhalten	<p>Das Verhalten eines Objekts wird durch eine Menge von → Operationen (Methoden) bestimmt. Die Operationen werden durch Botschaften aktiviert. Jede Botschaft aktiviert eine Operation gleichen Namens.</p>
Operationen (Methoden)	<p>Eine Operation ist eine ausführbare Tätigkeit im Sinne eines Algorithmus und wird durch eine Methode implementiert.</p> <p>Alle Objekte einer Klasse verwenden dieselben Operationen</p> <p>Jede Operation kann auf alle Attribute einer Klasse direkt zugreifen.</p>
Geheimnisprinzip	<p>= Datenkapselung = „Black-Box-Prinzip“.</p> <p>Eine Änderung oder Abfrage des Zustands eines Objekts ist nur über seine Operationen möglich. D.h. die Attributwerte sind außerhalb des Objekts nicht sichtbar.</p>

¹ Leider haben dem Englischen nicht ganz mächtige Informatiker den englischen Begriff „Instance“ mit Instanz statt mit Beispiel oder Exemplar übersetzt. Deshalb wird der Begriff Instanz in der deutschen Fachsprache auch als Synonym für Objekt verwendet.

3 UML – Unified Modeling Language

Mit der Unified Modeling Language hat sich eine Sprache etabliert, mit der objektorientierte Konzepte grafisch dargestellt werden können. Sie liegt seit Oktober 2015 in der Version 2.5 vor und bietet insgesamt 13 verschiedene Diagrammart, mit denen objektorientierte Softwaresysteme grafisch modelliert werden können:

- 6 Strukturdiagramme, wie z.B. Klassendiagramm, Objektdiagramm oder Komponentendiagramm, dienen zur Beschreibung der statischen Gegebenheiten eines Systems.
- 7 Verhaltensdiagramme, wie z.B. das Sequenzdiagramm, Anwendungsfalldiagramm oder Aktivitätsdiagramm dienen zur Beschreibung dynamischer Abläufe in einem System.

Im Rahmen des Programmierunterrichts werden wir uns fast ausschließlich mit Klassendiagrammen beschäftigen.

Beispiel: Klasse Auto

Auto
- farbe: TColor - kmStand: int - preis: double - aktGeschwindigkeit: int
+ bremsen(): int + starten(): void + beschleunigenUm(delta: int): int

Beispiel: Klasse Bruch

Bruch
- iZaehler: int - iNenner: int
+ Bruch() + Bruch(iZ: int, iN: int) + addiereDazu(b2: Bruch) : Bruch + subtrahierVon(b2: Bruch) : Bruch + multipliziereMit(b2: Bruch) : Bruch + dividiereDurch(b2: Bruch) : Bruch + toString() : String - <u>berechneGGT(int a, int b) : int</u>