



Generalisierung durch Vererbung

Projekt „DoME“

Wir betrachten wieder das DoME-Projekt. DoME steht übrigens für „Database of Multimedia Entertainment“ (ein cooler Name gehört zum Marketing ☺!).

Aufgabe 1 (DoME1)

Öffnen Sie das Projekt DoME1 und erzeugen Sie einige CD- und Video-Objekte. Erzeugen Sie ein Datenbank-Objekt und tragen Sie Ihre CDs und Videos ein. Lassen Sie sich den Datenbankinhalt auflisten. Versehen Sie Ihre CDs und Videos mit Kommentaren.

Aufgabe 2

a) Verschaffen Sie sich einen Überblick über den Quellcode der beteiligten Klassen.

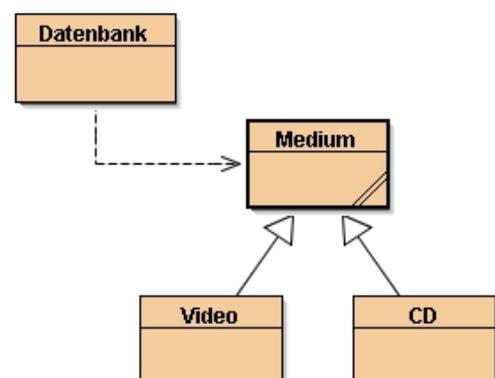
b) Welche fundamentalen Schwächen hat die vorliegende Lösung (Stichwörter: Code-Duplizierung, wartungsfreundlicher Code, erweiterungsfreundlicher Code)? Nennen Sie die Stellen im Code.

Aufgabe 3 (DoME2)

Das Konzept der Vererbung erlaubt es, eine Klasse als **Erweiterung** einer anderen zu definieren. Statt die Klassen `CD` und `Video` völlig unabhängig voneinander zu definieren, definieren wir eine Oberklasse `Medium`, die die Gemeinsamkeiten beider Klassen zusammenfasst.

a) Öffnen Sie das Projekt DoME2. Implementieren Sie die Klasse `Medium`. Welche Attribute/Methoden aus den Klassen `CD` und `Video` lassen sich in dieser Oberklasse **generalisieren**?

b) Analysieren Sie, wie in der vorgegebenen Klasse `Video` die Vererbungsbeziehung realisiert wurde. Erzeugen Sie ein `CD`-Objekt und rufen Sie einige Methoden auf. Können Sie auch geerbte Methoden aufrufen?



c) Passen Sie nun den Quelltext der Klasse `CD` an. Verwenden Sie anschließend in der Klasse `Datenbank` wieder eine `ArrayList` zur Verwaltung von Medien. Durch das Konzept der **Polymorphie** (griechisch: Vielgestalt) kann man eine Variable `medium` vom Typ `Medium` deklarieren, der man zur Laufzeit Variablen vom Typ `CD` oder `Video` zuweist.

```
private ArrayList <Medium> medien;  
  
medien = new ArrayList();  
  
public void erfasseMedium(Medium neuesMedium) {  
    medien.add(neuesMedium);  
  
}
```

Aufgabe 4

Vererbung erlaubt die Wiederverwendung bereits erstellter Klassen in neuen Zusammenhängen. Durch Vererbung wird es einfacher, eine bestehende Anwendung zu erweitern. Fügen Sie in Ihr Projekt ein weiteres Medium (z. B. Buch oder Videospiel) ein.

Aufgabe 5

Sicherlich ist Ihnen aufgefallen, dass bei der Ausgabe in der aktuellen Version des DoME-Projekts Informationen fehlen, z. B. über den Künstler der CD und die Anzahl der Titel. Der Grund liegt darin, dass die Methode `ausgabe` nur in der Klasse `Medium` implementiert ist. In Methoden der Klasse `Medium` sind aber nur die gemeinsamen Attribute verfügbar, die in `Medium` definiert sind. Die Klasse `Medium` darf nicht auf Attribute ihrer Unterklassen zugreifen.

Die Lösung des Problems erfolgt durch „**Überschreiben der Methode**“. Eine Unterklasse kann die Implementierung einer Methode überschreiben. Dazu deklariert die Unterklasse eine Methode mit der gleichen Signatur wie in der Oberklasse, implementiert jedoch einen anderen Rumpf. Die überschreibende Methode wird dann bei Aufrufen an Objekte der Unterklasse vorgezogen.

Bsp: Klasse `Video`

```
public void ausgabe(){  
    System.out.println("Video:");  
  
    super.ausgabe();  
  
    System.out.println("    Regisseur: " + regie);  
  
}
```

Im Gegensatz zur Regel für `super`-Aufrufe in Konstruktoren kann der `super`-Aufruf in einer Methode an jeder beliebigen Stelle in der Methode erfolgen. Er muss nicht die erste Anweisung sein.

Modifizieren Sie die Methode `ausgabe` der Klasse `CD` entsprechend.