

Übung 10: Modellierungsstrategien für Vererbungshierarchien

In dieser Übung befassen wir uns mit den beiden Modellierungsstrategien *bottom-up* und *top-down*, wie in der Vorlesung besprochen.

Aufgabe 1: Bottom-Up, Refactoring

Gegeben sind die bereits modellierten Klassen `Fahrrad`, `Motorrad`, `Auto` und `Lastwagen`, sowie die Teile `Cantileverbremse`, `Scheibenbremse` und `Trommelbremse`.

- Analysieren Sie die Klassen und suchen Sie nach Gemeinsamkeiten bzw. Verschiedenheiten.
- Entwerfen Sie eine geeignete Klassenhierarchie in UML; was sind geeignete Oberklassen bzw. Schnittstellen?
- Implementieren Sie die modellierten Komponenten und passen Sie die bestehenden Klassen entsprechend an; achten Sie dabei insbesondere auf die korrekte Verwendung von Konstruktoren. Dabei wird die Funktionalität der gegebenen Klassen kleiner und verlagert sich in der Klassenhierarchie weiter nach oben.

Hinweis: Es bietet sich eine mehrschichtige Klassenhierarchie für die Fahrzeuge sowie eine Schnittstelle für die Bremsen an. Sie haben gut gearbeitet, wenn es keinen redundanten Code mehr gibt.

Aufgabe 2: Top-Down, Modellierung einer Klassenhierarchie

Für das Programm gegeben in der Klasse `Zeichenprogramm` sollen Kreise, Dreiecke und Rechtecke modelliert werden. Diese sollen später in einem X/Y-Koordinatensystem gezeichnet werden.

- Welche Attribute sind für alle Formen nötig, welche sind speziell? Welche Attribute kann man berechnen, welche müssen vorgegeben werden?
- Zeichnen Sie ein UML Diagramm Ihrer Klassenhierarchie.
- Implementieren Sie Ihre Klassenhierarchie. Implementieren Sie für jede Klasse die `toString` Methode; achten Sie insbesondere auf die korrekte Verwendung von Konstruktoren.
- Vervollständigen Sie die `main` Methode der Klasse `Zeichenprogramm` gemäß der Kommentare.

Beispielausgabe des fertigen Programms:

Kreis um (5, 10) mit Radius 5

Rechteck von (0, 0) bis (3, 2) mit Breite 3 und Höhe 2

Dreieck an (0, 0), (1, 0) und (0, 1) mit Seitenlängen 1, 1 und 1.4142135623730951.