

Programmiermethoden in der Mathematik Design einer Klasse

Das Design einer Klasse in C++ wird hier an einem Beispiel erklärt:
Es soll eine Klasse `Bruch` definiert werden, die Brüche (also rationale Zahlen) repräsentiert.

Vor der Implementierung stellt sich die Frage:

Was ist ein Bruch?

In nicht-OO Denkweise bedeutet diese Frage einfach

Woraus (aus welchen Daten) besteht ein Bruch?

und die Antwort ist:

(1) Aus zwei ganzen Zahlen (Zähler und Nenner), wobei der Nenner ungleich Null ist.

In der OO-Denkweise kommt die Frage hinzu:

Was wollen wir mit der zu definierenden Klasse `Bruch` machen?

Hier wird der Vorteil der OO-Denkweise klar: Die Ziele, die mit der Klasse verfolgt werden, bestimmen evtl. entscheidend ihre Definition und die benötigten Daten.

In unserem Beispiel soll die Antwort lauten:

(2) Ein beliebiger Bruch soll erzeugt, initialisiert und ausgegeben werden können.

Damit ist klar, welche Attribute/Daten und welche Methoden implementiert werden müssen:

- Attribute: `zaehler`, `nenner`.
- Zusicherungen: `nenner` muss ungleich Null sein.
- Methoden: Neben Konstruktor und Destruktor soll es eine Ausgabemethode geben. Bei den Konstruktoren sind mehrere vorstellbar:
 - ohne Parameter (Default-Konstruktor)
 - mit einem Parameter (interpretiert diesen Parameter als Zähler und setzt den Nenner standardmäßig auf 1).
 - mit zwei Parametern für Zähler und Nenner.
- Zugriffsrechte: Zugriff auf die beiden Attribute soll nicht direkt, sondern nur über die Konstruktoren möglich sein. Damit wird vermieden, dass z.B. der Nenner auf Null gesetzt werden kann.