

## Thema einer Diplomarbeit

# Moderne Programmiermethodik — Auf der Suche nach Prinzipien

Wie schreibt man qualitativ hochwertige Programme? Und wie schafft man es mit wenig Aufwand? Die meisten Bücher über Programmierung behandeln nur Programmiersprachen, nicht die Methodik der Programmierung selbst. Ebenso behandeln Bücher über Softwareentwurf nur gewisse Notationen (seit 1998 quasi ausschließlich UML), aber nicht Kräfte und ihren Ausgleich (*tradeoff*). Eine lobenswerte Ausnahme unter diesen Büchern ist Bertrand Meyers *Object-Oriented Software Construction* (OOSC) von 1988. Dieses Buch basiert auf einer Theorie der Programmierung (nämlich axiomatischer Semantik) und auf einer Theorie der Modularisierung (von Meyer selbst in den ersten Teilen des Buches dargelegt).

Während andere Bücher nur das offensichtliche beschreiben (oder ganz formale Grundlagen), finden sich in OOSC Software-Prinzipien: Aussagen, die weder ganz formal sind, aber trotzdem verbindlich und universell anwendbar. Solche Prinzipien sind rar, aber sehr wertvoll, weil man sie nicht einfach in einem Software-Werkzeug implementieren kann. Die Diplomarbeit führt Meyers Arbeit, die nun schon 15 Jahre alt ist, fort und bringt sie quasi „auf den neuesten Stand“. Die wichtigsten Neuerungen sind dabei:

- Anstelle axiomatischer Semantik gibt es jetzt eine Theorie der Verfeinerung in der Programme und Spezifikationen gleichgestellt sind. (Siehe Literatur von Morgan und Hehner.)
- Seit 1988 haben sich objektorientierte Methoden überall durchgesetzt, Leser haben also ganz andere Voraussetzungen.
- Sowohl für die Vererbung als auch die Fehlerbehandlung kann ich genauere Richtlinien angeben als Meyer.

Programmiermethodik lebt von der Arbeit mit „richtigem Code“. Und um Entwurf zu erklären, braucht man sehr große Beispiele. Es gehört zum Wesen der Arbeit, Beispiele nicht nur als Demonstrationen zu betrachten, sondern als wesentliches Mittel, um konkrete Methoden überhaupt zu vermitteln. Fallstudien spielen daher in der Arbeit eine große Rolle.

### **Inhaltliche Schwerpunkte**

- Innere und äußere Softwarequalität; Abgrenzung der Programmierung (Konstruktion) als Teil der Softwareentwicklung
- Elemente einer Programmiermethode
- Spezifikationen und Programme; Module und Schnittstellen
- Formale Redundanz und Programmkonsistenz
- Entwurf mit Alternativen
- Vererbung: geleitet von Schnittstellen und ihren Implementierungen
- Fallstudien (mindestens zwei)
- Fehlerbehandlung

### **Literatur**

Bertrand Meyer, *Object-Oriented Software Construction*

Carroll Morgan, *Programming from Specifications*

E.C.R. Hehner, *A practical theory of programming*

Harlan and Mills, *Structured Programming: Theory and Practice*

C.B. Jones, *Software Development: A Rigorous Approach*