

# Stolpersteine bei der Integration funktionaler Konzepte in die imperative Programmierung

Frank Huch  
Christian-Albrechts-Universität zu Kiel  
fhu@informatik.uni-kiel.de

Als sehr weit verbreitetes Programmierparadigma hat sich die imperative, objektorientierte Programmierung durchgesetzt. Sprachen, wie Java oder C++ sind heutzutage sehr weit verbreitet. Ein Trend geht jedoch dahin, solche Sprachen mit Konzepten aus der funktionalen Programmierung anzureichern, wie z.B. Scala oder aktuelle Erweiterungen von Java in der neuesten Version 8. Auch erfreuen sich Skriptsprachen, wie Javascript, Ruby oder Python immer größerer Beliebtheit. Auch hier werden vermehrt funktionale Konzepte, wie z.B. funktionale Parameter und anonyme Funktionen verwendet.

Die Kombination von imperativen, seiteneffektbehafteten Konzepten der OO-Welt mit funktionalen Konzepten birgt aber auch einige, auf den ersten Blick vielleicht nicht erkennbare Gefahren, welche in diesem Vortrag näher beleuchtet werden sollen. In der funktionalen Programmierung werden Datenstrukturen in der Regel nicht verändert, sondern bei Veränderung neu erzeugt. Dies ist auch in objektorientierten Sprachen möglich und wird z.T. auch verwendet. Bei der Kombination von destruktiven und nicht-destruktiven Methoden bei der Definition von Klassen entstehen allerdings Fallstricke, welche man manchmal nicht überschauen kann.

Im Vortrag werden wir diese Stolpersteine genauer beleuchten. Eine Sensibilisierung für diese Themen in der Ausbildung angehender Informatiker scheint zunehmend wichtiger zu sein, weshalb wir auch darauf eingehen werden, in wie weit, diese Stolpersteine in die Programmierausbildung eingehen sollten.