

Entwurf und Implementierung einer ADT-orientierten Programmiersprache

Abstract

MICHAEL BURZAN

MICHAEL KUßMANN

„Programmiersprachen sind Sprachen und als solche Medien des Denkens“¹. Daher sollte der Entwurf einer Programmiersprache immer zum Ziel haben, den Programmierer beim Lösen von Problemen zu unterstützen, indem die Sprache ihm dabei hilft, das Problem zu strukturieren.

In oben genannter Arbeit wird eine statisch und stark typisierte, objektorientierte Programmiersprache namens TeaJay entworfen und implementiert, die den Begriff des Typs (im Sinne eines ADTs) in den Vordergrund stellt und dadurch das Nachdenken über den Schnittstellenentwurf fördern soll. Dies soll einem Entwickler dabei helfen, Probleme zu strukturieren, indem die Sprache es natürlich erscheinen lässt, Programme in schwach gekoppelte Module zu unterteilen. Die Sprache soll vorwiegend eine Lehrsprache sein, sich jedoch nicht so weit von der Pragmatik entfernen, dass sie nur zu einer Inelsprache der Lehre werden kann. Es soll mit ihr auch möglich sein, verschiedene Programmierparadigmen zu lehren (z.B. funktionale Programmierung).

Ein Typ stellt in TeaJay nur die Beschreibung einer Schnittstelle mit Sorten und Operationen dar. Die Kommunikation mit dem Typ darf ausschließlich über diese Schnittstelle geschehen. Eine Implementierung muss erst zur Laufzeit vorliegen. Dazu sorgt eine Laufzeitumgebung dafür, dass Implementierungen Just-In-Time geladen werden. Die entstandene Sprache lehnt sich syntaktisch an Java an und wird direkt zur JVM übersetzt. Weitere Besonderheiten der Sprache sind: Funktionsabschlüsse (Closures), typeswitch und Operatorüberladung.

¹Clasing, Achim: Programmiersprachen - Konzepte, Strukturen und Implementierung in Java, Spektrum, 2011, S. 2.