

JavaGl: bessere Interfaces für Java

Stefan Wehr
Universität Freiburg
`wehr@informatik.uni-freiburg.de`

JavaGl erweitert Java 1.5 um ein verbessertes und allgemeineres Interface-Konzept. Das von Haskell's Typklassen-Mechanismus beeinflusste Konzept erlaubt eine Reihe interessanter Features:

1. Implementierungen von Interfaces können retroaktiv erfolgen, d.h. abgekoppelt von den Definitionen der beteiligten Klassen und Interfaces sein.
2. Interfaces können Referenzen auf die implementierende Klassen enthalten. Damit lassen sich binäre Methoden spezifizieren.
3. Interfaces können sich über mehrere Typen erstrecken und damit verschränkt-rekursive Abhängigkeiten widerspiegeln.
4. Interfaces können nicht nur als Typen sondern auch in Typconstraints verwendet werden.
5. Bestimmte Methoden von Interfaces erlauben symmetrischen Multi-Dispatch.
6. Retroaktive Implementierungen von Interfaces können Bedingungen unterworfen sein, die es erlauben, Implementierungen auf bestimmte Typen einzuschränken.
7. Interfaces können statische Methode enthalten.

Diese Features subsumieren bestimmte Designpatterns, erhöhen die Ausdruckstärke des Typsystems und erlauben einheitliche Lösungen von Problemen im Bereich der Erweiterung und Integration bestehender Softwaresysteme, für deren Lösung bisher mehrere verschiedene Spracherweiterungen nötig waren.

Die Sprache JavaGl besitzt eine Formalisierung, für welche Soundness und Entscheidbarkeit des Typsystems, sowie Determiniertheit der Ausführungsregeln bewiesen sind. Desweiteren gibt es eine Implementierung der Sprache, welche auf dem in der Entwicklungsumgebung "Eclipse" verwendeten Java Compiler beruht. Mit Hilfe dieser Implementierung wurde eine Reihe von Fallstudien durchgeführt, welche die praktische Anwendbarkeit der Sprache demonstrieren. Außerdem zeigen Benchmarks, dass der Compiler Code mit guter Performanz erzeugt.

Weitere Informationen zu JavaGl, inklusive des Compilers, sind im Internet unter der URL <http://www.informatik.uni-freiburg.de/~wehr/javagi/> zu finden.