

[News](#) [Rubriken](#) [Blogs](#) [Podcast](#) [Szene](#) [Software](#) [FAQ-Liste](#)
Sie sind G

[Sprachen](#) [Architektur/Methoden](#) [Werkzeuge](#) [Know-how](#) [Standards](#) [Literatur](#)


[heise Developer](#) > [Rubriken](#) > [Sprachen](#) > [Dart: Googles neue Programmiersprache](#)

Sprachen 10.10.2011 - 20:24
Schlagwörter: Dart, JavaScript

Eberhard Wolff

Dart: Googles neue Programmiersprache

Neuanfang



Google hat mit Dart eine neue Programmiersprache vorgestellt – und positioniert sie auf den ersten Blick auf Augenhöhe mit JavaScript. Dabei weiß Dart mit einem neuen Typsystem zu Punkten, das die Vorteile statischer und dynamischer Typisierung miteinander verknüpft. Interessant ist die Sprache besonders für den mobilen

Bereich.

Die [Goto Conference](#) in Århus begann mit einem echten Highlight: Einer Keynote über die [neue Programmiersprache Dart](#). Vorgestellt wurde sie von Gila Bracha, der zuvor an der Java Language Specification gearbeitet hat, und Lars Bak, der an der Java Hotspot VM beteiligt war und nun an der Google V8 JavaScript-Engine entwickelt.

Vorab: Dart ist bei weitem noch nicht fertig. Der momentane Stand ist eine Technologie-Preview, die die Entwickler vor allem für Feedback nutzen möchten. Aber die Richtung ist schon erkennbar.

Dart ist von der Syntax her in die Tradition der C-ähnlichen Sprachen einzureihen. Besonders interessant ist das Typsystem. Klassisch teilt sich die Programmiersprachen-Welt in statisch typisierte und dynamisch typisierte Sprachen. Typsysteme werden genutzt, um Fehler im Quelltext frühzeitig zu identifizieren. Diesen Ansatz treibt Dart auf die Spitze: Das statische Typsystem meldet nur Warnungen, falls Code fehlerhaft sein könnte. Der Code ist aber wie in einem dynamischen Typsystem dennoch ausführbar. So kombiniert man die Sicherheit eines statischen Typsystems mit der Flexibilität eines dynamischen Typsystems.

Nebenläufigkeit ist ein weiterer Bereich, in dem sich eine Programmiersprache positionieren muss. Hier wählt Dart, ähnlich wie Scala, ein durch Erlang inspiriertes System mit Aktoren. In Dart nennen sie sich *Isolates*. Isolates haben intern nur einen Thread und kommunizieren durch Nachrichten miteinander. Dieses Konzept erlaubt eine parallele Ausführung, da mehrere Isolates gleichzeitig aktiv sein können. Die Entwicklung ist hierbei sehr einfach, da in jedem Isolate jeweils nur ein Thread aktiv sein kann.

Als Werkzeug steht schon eine auf Eclipse basierte Umgebung namens Dart Editor bereit. Entwickler können Dart-Anwendungen aber auch direkt im Browser ausführen, da Dart sich auch direkt in JavaScript kompilieren lässt. Alternativ kann die Ausführung auf einer eigenen VM stattfinden. Es gibt bereits eine GUI-Library und als Demo-Anwendung einen Newsreader, der unter Chrome, aber auch auf einem iPad läuft.

Dart ist aber bei weitem noch nicht fertig: So ist noch nicht klar, wie Reflection unterstützt wird und auch das Pattern Matching könnte Google noch in die Sprache einbauen. Es gibt zudem noch keine Pläne, die Dart-VM in Chrome zu integrieren – aber dies ist sicher eine Option in der Zukunft. Zur Zeit lässt Dart-Code sich in allen Browser durch die Wahl der JavaScript-Kompilierung ausführen. Die Performance bietet derzeit noch Optimierungspotential, sei aber laut Bak schon vergleichbar mit den ersten JavaScript-Implementierungen in Chrome.

Dart ist kein Ersatz für JavaScript, sondern soll vor allem den fragmentierten Mobile-Bereich als Zielplattform haben. Im Vortrag führten Bracha und Bak das Thema jedoch nicht weiter aus. Die Ansätze sehen von der Syntax und den Sprachmöglichkeiten her interessant aus. Ein derartiges Typ-System nutzt noch keine andere Sprache dieser Größenordnung.

Wer Dart schon ausprobieren will, kann sich erste Beispiele direkt im Browser unter dartlang.org ansehen, das Dart-Open-Source-Projekt liegt unter dart.googlecode.com. (r)

Eberhard Wolff

ist Gründungsmitglied der Java-Champions, Autor zahlreicher Fachartikel und Fachbücher und regelmäßiger Sprecher auf internationalen Konferenzen.

Permalink: <http://heise.de/-1358408>