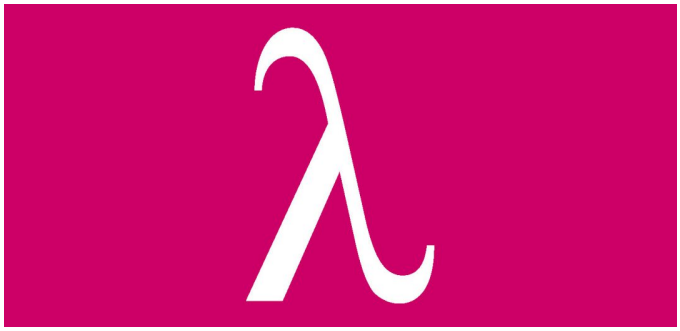

Funktionale Programmierung in Java

0

BY HANNES TSCHERRIG ON 7. MÄRZ 2016 INFORMATIK

SHARE.

– Mit der achten Version der Programmiersprache Java wird vieles anders für Software-Entwicklerinnen. Prof. Dr. Ruedi Arnold von der Hochschule Luzern – Informatik erklärt im Interview, wie sich mit neuen Features wie Streams und Lambdas Java-Programmierern eine ganz neue – funktionale – Programmierweise auftut.

Warum ist funktionale Programmierung gerade jetzt (wieder) aktuell?

Programmiersprachen kommen und gehen. Bestehende entwickeln sich ständig weiter. So auch die Sprache, welche wir an der Hochschule Luzern hauptsächlich verwenden, Java. 2014 kam Version 8 von Java heraus und damit eine ganze Menge neuer Features. Vor allem ermöglichen neue APIs (Application Programming Interfaces) wie Streams und neue Sprachmittel wie Lambdas einen anderen Programmierstil. Konkret bietet die primär als objektorientiert bekannte Sprache Java damit bessere Unterstützung für funktionale Programmieransätze. Grund genug also um sich die Sache etwas genauer anzusehen. Denn funktionales Programmieren bringt im Vergleich zu klassisch imperativ-objektorientierter Programmierung einige Vorteile.

Was bringen die neuen Java-8-Features?

Ein einfaches und bekanntes Beispiel für die Vorteile der neuesten Java-Version sind Filter-, Map- und Reduce-Operationen. Diese erlauben einfaches Filtern und Umwandeln einer Datenstruktur. Statt wie bisher eine Liste zu definieren, mit Hilfe von Schleifen und einer Zählvariablen für jedes Element der Liste Operationen durchzuführen, dazwischen den Zähler zu inkrementieren, sagt man neu einfach: «Hier ist die Liste. Filtere alle leeren Elemente heraus. Füge zu jedem Element ‹java› hinzu. Und fasse alles zu einem grossen Wort zusammen, bitte.» Was doch viel verständlicher klingt, oder? Ausserdem können wir neue Funktionalität als Lambda herumreichen…

Als Lambda?

Ja. Lambda-Ausdrücke sind eine Art anonyme Methoden. Seit Java 8 sind sie sogenannte «First Class Citizen». Das heisst, sie sind neu integraler Bestandteil der Programmiersprache und können überall unmittelbar angewandt werden. Das Tolle an Lambdas: Sie lassen sich inline, d.h. direkt im Code erstellen und direkt an andere Funktionen weiterreichen. Dies erlaubt ganz neue, funktionale Möglichkeiten. Zudem sind Lambdas prägnant, konzise und gut lesbar.

Was ist anders beim funktionalen Programmieren?

Imperative Programme, zu denen auch objektorientierte Java-Programme gehören, beschreiben das «Wie». Sie beschreiben aus Sicht der Maschine, welche Schritte ausgeführt werden müssen, um ein Problem zu lösen. Der Programmablauf ist explizit vorgegeben durch die charakteristischen

Kontrollstrukturen Sequenz, Verzweigung und Schleife. Dabei denken Programmierer wie die Maschine, die sie programmieren.

Deklarative Ansätze – und damit funktionale Programme – beschreiben eher das «Was». Hierbei wird nicht definiert, wie ein Programm vorzugehen hat, sondern das Problem wird eher beschrieben. Die Details des «Wie» bleiben der angewandten Programmiersprache überlassen, der Programmablauf ist nicht explizit vorgegeben. Dabei können Programmierer auf einer höheren Abstraktionsstufe denken und programmieren.

Wozu dient diese Programmierweise?

Funktionale Programmierung bietet eine höhere Stufe der Abstraktion. Dadurch können Probleme anders kategorisiert werden und Gemeinsamkeiten ähnlicher Komponenten lassen sich leichter erkennen.

So ist zum Beispiel das Parallelisieren einer Stream-Operation in funktionalem Code ein Leichtes. Werden dieselben Aufgaben imperativ und ohne Streams implementiert, verlangt eine Parallelisierung wesentlich mehr Gehirnschmalz und Code. Dies deshalb, weil der Code «von Hand» auf mehrere parallele Programm-Teile aufgeteilt werden muss und diese Teile untereinander explizit koordiniert werden müssen.

Und Parallelisierung und nebenläufiger Programme werden heute mit Multi-Core und Multi-Prozessor-Maschinen immer wichtiger und auch um hochverfügbare (Stichwort «Reactive») Online-Dienste anbieten zu können. Java 8 bietet für die Ausführung von asynchronem Code zum Beispiel mit der Klasse CompletableFuture sehr elegante und mächtige funktionale Möglichkeiten.

Was bedeutet «Zustandslosigkeit» beim Programmieren?

In der rein funktionalen Programmierung herrscht Zustandslosigkeit. Reine Funktionen („pure functions“) produzieren keine (beobachtbaren) Nebeneffekte, sind zustandslos und liefern für dieselbe Eingabe immer dasselbe Resultat. Reine Funktionen sind in sich abgeschlossene Einheiten und müssen nicht versteckt oder untereinander koordiniert werden. Dadurch lassen sie sich wunderbar wiederverwenden und parallelisieren (d.h. gleichzeitig mit anderen Funktionen ausführen). In rein funktionalen Programmen gibt es überhaupt keine sich verändernden Zustände

Ganz anders bei der imperativ-objektorientieren Programmierung: Hier spielt der Zustand von einem laufenden Programm typischerweise eine zentrale Rolle. Dies lässt sich schön an der oben erwähnten typischen for-Schleife illustrieren, da wird normalerweise zuerst eine Zählvariable initialisiert und dann modifiziert, um den aktuellen Zustand von der Abarbeitung dieser Schleife zu repräsentieren. Und eben: Es ist viel schwieriger Programme mit veränderlichem Zustand zu parallelisieren.

Wird ab jetzt also nur noch funktional programmiert?

Das ist nicht das Ziel! Imperative Programme haben durchaus ihre Berechtigung. Ich würde persönlich nicht alles funktional programmieren. Gut ausgebildete Programmiererinnen kennen die aktuellen Sprachmittel und können dann eben passend auswählen, darum geht es! Solide, handwerklich gute Programmierer («Software Craftsmen») kennen unterschiedliche Programmiersprachen und –paradigmen. Software Craftsmen unterscheiden eben zwischen Nagel und Schraube. Sie wissen, wann sie mit dem Hammer und wann mit einem Schraubenzieher zu Werke gehen müssen.

Mehr zu funktionalem Programmieren in Java

Oracle: What's New im JDK 8
Ruedi Arnold präsentiert «Java 8: Lambdas, Streams & Co».
Ruedi Arnold präsentiert funktionales Denken und Programmieren mit Java 8

RELATED POSTS

29. FEBRUAR 2016 0
Blockchain einfach erklärt

9. FEBRUAR 2016 1
«In der Informatik-Lehre musst du immer up-to-date sein»

15. JANUAR 2016 0
«Als Dozent bin ich Coach»

LEAVE A REPLY

Your Comment

Your Name

Your Email

Your Website

CAPTCHA Code *

POST COMMENT