

Sonntag, 25. Oktober 2015:

Einsteiger-Workshop zu Haskell, der rein funktionalen Programmiersprache



Was ist schneller als C++, prägnanter als Perl, regelmäßiger als Python, flexibler als Ruby, typisierter als C#, robuster als Java und hat absolut nichts mit PHP gemeinsam? Es ist Haskell!

Der Curry Club Augsburg, vertreten durch Mathe-Studierende wie Evelyn A., Ingo B., Matthias H., Maximilian H., Tim B. und Stefan H., lädt zum großen Haskell-Workshop ein. Zielgruppe sind in erster Linie völlige Haskell-Neulinge. Teilnahmevoraussetzung ist Erfahrung mit einer beliebigen anderen Programmiersprache.

Wer vorhat, am Workshop teilzunehmen, kann sich auf <http://curry-club-augsburg.de/> unverbindlich anmelden. Kosten gibt es keine.

Haskell ist eine moderne und innovative Programmiersprache, die sich von bekannten imperativen Sprachen in vielerlei Hinsicht deutlich unterscheidet: Ein Haskell-Programm besteht nicht etwa aus einer Abfolge von auszuführenden Anweisungen, sondern aus einer Ansammlung von Deklarationen, deren Reihenfolge keine Rolle spielt. Auch gibt es keine veränderlichen Variablen, und ausgewertet wird nur, was wirklich benötigt wird; unendliche Datenstrukturen sind möglich und sinnvoll.

Dieses Denkparadigma mag anfangs sehr ungewohnt sein, zieht jedoch eine Reihe von Vorteilen mit sich: Da es keine Nebenwirkungen wie beispielsweise globale Variablen gibt, kann man Code rein lokal verstehen. Damit wird es einfacher, modular Komponenten zusammenzubauen, sich Datenflüsse klarzumachen und Code auf seine Korrektheit hin zu überprüfen. Insbesondere vereinfacht sich die Programmierung mit Threads enorm.

Ferner ist Haskell's starkes statisches Typsystem eine große Hilfe beim Programmieren und verhindert viel mehr Fehler schon während des Kompilierens, als man vielleicht aus anderen Sprachen gewohnt ist. Es gibt das Motto, dass, wenn Haskell-Code erst einmal erfolgreich durchkompieliert, er dann auch schon korrekt sei. Das ist sicherlich übertrieben, hat aber einen erstaunlich wahren Kern.

Bei der Beschäftigung mit Haskell lernt man viele neue Herangehensweisen kennen, die auch in anderen Sprachen nützlich sind; das ist einer der Hauptvorteile an Haskell, der auch dann noch relevant ist, wenn man aus verschiedenen Gründen im täglichen Leben nicht in Haskell programmieren möchte.

