Classes, Jim, But Not as We Know Them — Type Classes in Haskell: What, Why, and Whither

Simon Peyton Jones

Microsoft Research

Haskell is now quite widely used, but its most important contributions are the *ideas* that it embodies. In this talk I will focus on one of these ideas, namely type classes, with a few anecdotes and reflections along the way about the process of developing the language.

Type classes are probably Haskell's most distinctive feature. The original idea is very neat and, better still, it led to a long series of subsequent generalisations and innovations. Indeed, although the language is now nineteen years old, Haskell's type system is still in a state of furious development. For example, I am involved in adding type-level functions to Haskell, as I will briefly describe.

I will explain what type classes are, how they differ from the classes of mainstream object oriented languages, why I think they are so cool, and what the hot topics are. I'll give plenty of examples, so you don't need to already know Haskell.

S. Drossopoulou (Ed.): ECOOP 2009, LNCS 5653, p. 1, 2009.