MULTIPARADIGM PROGRAMMING IN SCALA

TUTORIAL PRESENTATION

H. Conrad Cunningham and James C. Church
Department of Computer and Information Science
University of Mississippi
University, MS 38677 USA
(662) 915-7396
cunningham@cs.olemiss.edu

OVERVIEW

This tutorial introduces the new programming language Scala [2,3,4] being developed by Martin Odersky and his team at EPFL in Switzerland. Although Scala arose from the academic research community, it has generated considerable interest in the practitioner community as a possible replacement for Java in enterprise software development. (Odersky was one of the designers of the generic type features added to Java 5.)

Scala programs execute on the Java Virtual Machine (JVM) and can interoperate with Java programs and application programmer interfaces (APIs). It is a *multiparadigm programming language* that natively supports the imperative, object-oriented, and functional styles of programming. In addition, using the flexible features of the language's syntax, powerful library-based extensions provide actor-based concurrency-oriented programming and language-oriented programming facilities. These multiparadigm programming features appear in some form in other languages of contemporary interest such as Ruby, F#, and even C# and likely future versions of Java.

Why multiparadigm programming? Author and researcher Timothy Budd reports, "Research results from the psychology of programming indicate that expertise in programming is far more strongly related to the number of different programming styles understood by an individual than it is the number of years' experience in programming" [1]. He also states that the "goal of multiparadigm computing is to provide ... a number of different problem-solving styles" so that a programmer can "select a solution technique that best matches the characteristics of the problem to be solved" [1].

This tutorial presents Scala's functional and object-oriented features in the context of multiparadigm programming. It is based on the presenters' recent experiences in learning and teaching Scala programming.

PRESENTERS

H. Conrad Cunningham is Professor and Chair of Computer and Information Science at the University of Mississippi (Ole Miss). His professional interests include software architecture, programming methodology, programming and domain-specific languages, and concurrent and distributed computing. He has a BS degree in mathematics from Arkansas State University and MS and DSc degrees in computer science from Washington University in St. Louis. Cunningham has taught courses on software engineering, software architecture, software components, object-oriented programming, program semantics, concurrent and distributed programming, functional programming, and programming languages. In the Fall 2008 semester, he taught a graduate course on multiparadigm programming using the Scala language and in Spring 2009 is teaching a graduate course on domain-specific languages that is also using Scala for projects.

James C. Church is a PhD student in Computer and Information Science at the University of Mississippi (Ole Miss). His professional interests are in image processing, computer vision, and programming languages. He has a BS degree in computer science from the University of Tennessee at Martin and an MS in computer science from the University of Mississippi. Church has taught CS 1 and CS 2 courses using Java, an advanced undergraduate course on Web programming using Python, and a "CS 0" course using Alice. In Spring 2009, he is teaching a service course on C and C++ programming. He learned Scala during the 2008 summer and fall by first writing a Sudoku-solving program in Scala.

REFERENCES

[1] Budd, T. A., *Multiparadigm Programming in Leda*, Reading, Massachusetts: Addison-Wesley, 1995.

[2] Odersky, M., *Scala by Example*, Lausanne, Switzerland: Programming Methods Laboratory, EPFL, 2009, http://www.scala-lang.org (documentation), retrieved January 20, 2009.

[3] Odersky, M., Spoon, L., Venners, B., *Programming in Scala: A Comprehensive Step-By-Step Guide*, Artima, Inc., 2009.

[4] Scala Project, Scala website, Lausanne, Switzerland: Programming Methods Laboratory, EPFL, 2009, http://www.scala-lang.org, retrieved January 20, 2009.