Working with Groovy

Course Summary

Description

Groovy is a dynamic scripting and programming language for the Java platform. It combines the dynamic features of modern scripting languages such as Ruby and Python with familiar Java syntax. To quote one of the Groovy developers: “Groovy is what Java would have been if it had been created in the 21st century.”

This course introduces the Java developer to the Groovy language. The course focuses on understanding the internals of how Groovy works in addition to understanding the Groovy language syntax. After taking this course developers will understand the Groovy syntax and be able to leverage existing Java classes within Groovy.

Objectives

At the end of this course, students will be able to:

- Write applications using Groovy.
- Understand how Groovy operates within the Java Virtual Machine.
- Incorporate existing Java classes and libraries within Groovy applications.
- Learn to add new methods and member variable to existing Java or Groovy classes dynamically.
- Understand the role of Closures within Groovy.
- Take advantage of Groovy’s simplified object configuration syntax.
- Learn how to override operators for Groovy or Java classes
- Understand the concept of Metaprogramming and how to leverage it to simplify application development.
- Explore Groovy’s Regular Expression syntax for easily managing String processing

Topics

- Language Overview
- Basic syntax and Scalar variables
- Advanced Classes and Closures
- AST Transformations

Audience

This an intermediate-level Groovy training course, designed for developers who need to understand how and when to use Groovy in Java and JEE applications.

Prerequisites

Attendees should have practical basic Java development experience.

Duration

Three days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Working with Groovy

Course Outline

I. *Language Overview*
   A. What is Groovy?
   B. What Groovy can do
      1. Java programmers
      2. Script programmers
      3. Agile programming
   C. Installing
   D. Running Groovy scripts

II. *Basic syntax and Scalar variables*
   A. Syntax rules
   B. Numbers
      1. Integers
      2. Floats
      3. BigDecimal
   C. Strings
      1. Double quoted
      2. Single quoted
      3. Here documents
      4. Slash quoted
      5. GStrings
      6. Operators
      7. Collections
      8. Lists
      9. Coding a closure
         a) The it parameter
         b) Passing multiple parameters
         c) Naming parameters (the -> operator)
      10. Maps
      11. Ranges
      12. Flow Control
      13. If statements
         a) The truth in Groovy
      14. Switch statements
      15. While loops
      16. For loops
      17. Exceptions
      18. Classes
      19. Defining classes
         a) File-to-class relationships
      20. Member variables
         a) Automatic getter/setter generation
         b) Default visibility
         c) Safe dereferencing with ? operator
      21. Methods
         a) Optional parameters and default parameters
      22. Operator overloading
      23. Automatic constructor generation
         a) Initializing property values in the constructor
      24. The Closure Groovy class
      25. Coding a method that expects a closure
         a) Calling into the closure
         b) Passing parameters

III. *Advanced Classes and Closures*
   A. Closures
      1. Using methods as closures
      2. Polymorphic closures
   B. Operator overloading
   C. Metaprogramming
      1. Discovering a class
      2. Discovering fields
      3. Discovering methods
      4. Method resolution
         a) MetaClass
         b) MetaProperty
         c) MetaMethod
      5. Pointers
         a) Method
         b) Field
      6. Calling methods that do not exist
         a) ExpandoMetaClass
         b) Categories

IV. *AST Transformations*
   A. What are Transformations?
   B. Understanding the Abstract Syntax Tree
   C. Common built-in Transformations
      1. @Singleton
      2. @Delegate
      3. @Category
      4. @Mixin
      5. @TimedInterrupt
      6. @ConditionalInterrupt
      7. @EqualsAndHashCode
      8. Regular Expressions
      9. Regular Expression syntax
      10. The =~ operator
      11. The ==~ operator
      12. Common methods that use Regular Expressions
      13. Builders and Slurpers
      14. What are Builders and Slurpers?
      15. NodeBuilder
      16. MarkupBuilder
      17. AntBuilder
      18. Creating custom builders
      19. Using the ConfigSlurper
      20. Writing a Slurper

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically