Migrating to Java 6

Course Summary

Description

This course shows Java-1.4 developers the new language features in Java 5 and 6, and, more importantly, illustrates best programming practices as they've been affected by these new features. This is an accelerated course that assumes long experience with earlier versions of Java. Students will leave familiar with various new bits of syntax, and also with an understanding of the exciting new options they have for API design and general-purpose Java 6 coding.

Objectives

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

- Take advantage of ease-of-use features including the simplified for loop, auto-boxing, and varargs.
- Use and develop enumerated types, including those with their own state and inherent behavior.
- Use generics effectively when coding to the Collections API.
- Develop one's own generic types.
- Use Java annotations as defined by other APIs, and develop one's own.
- Understand issues of compatibility and migration in mixed 1.4/5 environments.
- Develop annotation processors according to the new standard API, and plug them into Java compilers and other tools.
- Use built-in support for XML binding via JAXB.
- Use built-in support for web-service clients via JAX-WS.

Topics

- New Features in Java 5
- New Features in Java 6

Audience

This course is designed for Java Developers.

Prerequisites

Experience with Java programming using the 1.4 SDK is essential.

Duration

Two 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

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Migrating to Java 6

Course Outline

I. New Features in Java 5

A. Ease-of-Use Features
1. Goals of Java 5
2. Simplified for Loop
3. Variable Parameter Lists
4. Formatted Output
5. Static Imports

B. Enumerated Types
1. Native Enumerations
2. Ordinal and String Representations
3. Looping Over Enums
4. Stateful and Behavioral Enumerations

C. Generics
1. Using Generics
2. Generics in the Collections API
3. Developing Generic Classes
4. Auto-Boxing
5. Convertability of Generics
6. Wildcards
7. Type Erasure
8. Generic Methods
9. What You Can't Do
10. Strong and Weak Suits

D. Annotations
1. Annotations
2. Aspect-Oriented Programming
3. Native Annotations
4. The Java Annotations Model
5. What Can Be Annotated
6. Annotations vs. Descriptors
7. Java EE Annotation Examples

E. Compatibility and Migration
1. Compatibility: Compiler and Runtime
2. Mixing 1.4 and 5 Classes
3. Compatibility with Generics: Type Erasure
4. Compatibility with Enumerations and Varargs
5. Migrating 1.4 Code to Java 5
6. Runtime Type Safety with "Checked" Collections

II. New Features in Java 6

A. Annotation Processors
1. Pluggable Annotation Processors
2. Processing in Rounds
3. Reading the Type Model
4. Validation
5. Generating Code
6. Reading Resources
7. Creating Source Files

B. XML and Web Services APIs
1. The Java API for XML Binding
2. Binding vs. Parsing
3. Generating an Object Model
4. Working with JAXB Object Models
5. JAXB Contexts, Marshallers, and Unmarshallers
6. Interoperable Web Services
7. SOAP and WSDL
8. The Java API for XML Web Services
9. Generating Proxies from WSDL
10. Using Generated Proxies
11. The Role of JAXB
12. Building Client Applications

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