Idiomatic Java

Angelika Langer

Consultant, Germany

This is a tutorial for Java programmers who want to deepen their knowledge of the language. We will look into various core Java language features in order to explore how these features can be sensibly used in practice. The focus is not so much on the features themselves, but on advanced usage of them and Java specific programming idioms.

Some examples: What is it that we express when we use an abstract class as opposed to an interface? What is the difference between the various types of inner classes and when would we prefer one over the other? How does object initialization work in detail and why would we want to avoid invocation of virtual functions in constructors? How can we implement object release and why and when can't we rely on finalization for this purpose?

The language features will be put into the context of higher-level concepts such as generic programming and preservation of object integrity. Since Java lacks many language features that other programming languages have, what do we do in Java instead? How can we express the notion of a function object or function pointer in Java? What is the Java counterpart of generic programming? How do we express immutability of objects in Java?

Angelika Langer works as a freelance trainer and mentor. Her main area of interest and expertise is object oriented software development in C++ and Java. She writes a column for the US magazine C++ Report, is a recognized speaker at OO conferences all over the world, contributed to the standardization of C++ on the ISO/ANSI committee, and is author of the book "The Standard IOStreams and Locales". For further information see http://www.camelot.de/~langer/.