Java Programming Idioms

Angelika Langer
Freelance trainer/consultant
Munich, Germany

The programming language Java was introduced to the IT community as an “easy-to-learn” and “easy-to-use” language (typically referring to C++ for comparison). While this is certainly true, it turns out that even this supposedly easy language offers a lot of choices and comes with many programming idioms that must be understood thoroughly in order to produce high-quality software implemented in Java. In this tutorial we look into the less obvious limitations of some concepts and discuss some fundamental, yet typical misunderstandings.

Due to the time frame the discussion cannot be exhausting and we will focus on two language features that exist in Java since its advent, namely the implications of reference semantics on object copying and comparison and the use of exceptions for error indication and error handling.

Copying and Comparison. Regarding object copying and comparison in Java we examine the semantics of clone() and equals(). Questionable implementations of these functions can be found galore (in the JDK, in popular textbooks, even in commercial applications). It turns out that correct and symmetric implementation of these two functions is a non-trivial task. We will look into the details.

Exception Handling. Use of exceptions is mandatory in Java and cannot be avoided, yet in practice they are often misunderstood. We explain why and when one would want to handle exceptions at all (a common misunderstanding is that every potential exception must be caught.) We discuss common problems such as exception hiding, improper use of throws specification, and resource leaks and inconsistent object state.

Tutorial objective
Attendants will gain a deeper understanding of core Java language features such as copying and comparison of objects and correct use of exception handling. Every Java programmer should be familiar with the topics covered in this tutorial.

Attendee background
Participants must be familiar with the basic concepts of the Java programming language. Those attendants with a serious interest in high-quality software will get most out of the tutorial. It’s not for the casual Java programmer.

Presentation format
This tutorial will be presentation based, but the presentation format is open to questions and discussions. Participants are expressly encouraged to submit concrete problems for discussion and we will sketch out the principals of solution wherever possible and useful for further illumination of the tutorial objective.
Angelika Langer is a freelance trainer/consultant working and teaching in the area of object-oriented software development in C++ and Java. She is co-author of the book “Standard C++ IOStreams and Locales” released in January 2000, and co-author of a column in C++ Report since 1997. Her teaching is backed by 12+ years of software development in the IT industry, including 5 years of compiler construction and involvement in C++ library development as well as the standardization of C++. She has been interested in Java for 5 years meanwhile, and many of the observations regarding Java presented in this tutorial are based on practical experience in large-scale Java projects. She is currently at work on a book about advanced Java programming idioms.