Introduction

With the release of Java 25, developers gain access to one of the most intriguing and forward-looking language enhancements—Stable Values, introduced through JEP 502. This new feature redefines how immutability and performance can coexist by allowing developers to defer initialization while still enabling the JVM to treat values as constants. This seminar introduces participants to the motivation, design, and practical use of Stable Values in modern Java applications. Through clear explanations and real-world examples, we will explore how Stable Values bridge the gap between lazy initialization and immutable design, enabling cleaner, safer, and more performant code. By the end of the session, attendees will understand how Stable Values fit into Java's evolving approach to performance optimization, startup efficiency, and memory safety—and how to start integrating this concept into their own projects.

Prerequisites

This seminar was developed for experienced software developers with practical experience in the Java programming language.

Duration

This seminar includes two academic hours. It is possible to add more topics and more coding exercises by extending its duration.

Premium Training

This seminar can be delivered either in Hebrew or in English. It can be delivered either online or in a hybrid format, allowing participants to choose whether to attend the training in the meeting room or join online. During the training itself, we use a Moodle-based Learning Management System, which you can access using a web browser from anywhere.

The Lecturer

This seminar is delivered by Haim Michael, an experienced, well-known software development trainer with more than 25 years of experience in software development training, more than 30 years of experience in Java, more than 30 years of experience in JavaScript, more than 15 years of experience in TypeScript, and more than 15 years of experience in Python.

The Topics

The following is the list of topics we are going to cover. Each topic has a dedicated presentation:

Introduction

Immutability in Java: Then and Now

The Motivation Behind JEP 502

Understanding Stable Values

Deferred Initialization and Lazy Loading

JVM Optimizations and Constant Folding

Performance and Thread Safety Considerations

Practical Use Cases for Stable Values

Best Practices