Course code:	**J/MT**
Course title:	**Developing multithreaded applications in Java**
Days:	5

Description:

Course intended for:

The training is intended for Java programmers wanting to get familiar with the issues of multithread application development.

Course objective:

The training objective is to get the users familiar with theoretic and practical aspects of multithread programming in Java language. During the training, basic concepts associated with concurrent computing will be presented, as well as selected theoretical issues, such as prioritization models, starving, deadlock and its solving, exclusion in access or lack of it, design patterns associated with concurrent computing or methods for improvement of performance of multithread programs. The participants will also discuss and practice the functionality and mechanisms available in J2SE, used for development of concurrent programs, divided into Java platform versions, in which they have been published: Java 7, Java 5 and the preceding versions: data collections, concurrency pattern realization, methods of excluding and synchronizing, concurrency in GUI and classes for implementation of concurrent "divide and rule" algorithms.

The training participants will encounter various problems and traps that emerge during development of multithread applications, and the material and course of exercises allow them to identify these and avoid them in the future.

Course strengths:

During the training, the participants will implement concurrent programming pattern realizations, compare them with solutions delivered by Java platform and then use the latter for development of concurrent algorithms. Thanks to the format of the training, it allows the participants to understand the underlying concepts of the java.util.concurrent library, and knowledge on multithread programming will be built "from the start". The knowledge acquired will be useful in creation of multithread applications, to a great extent, independently of the language applied. The training curriculum is largely focused on reproducing various errors in concurrent applications in order to understand them.

Requirements:

The training participants are required to have the Java programming skills (to be learned at the course J/JP).

Course parameters:

5*8 hours (5*7 net hours) of lectures and workshops (with a visible emphasis on workshops).

Group size: no more than 8-10 participants.

Course curriculum:

1. Introduction to multithreading
   - I. Concurrent, parallel and serial processing
   - II. Preemption and lack of preemption
   - III. Thread and process
   - IV. Multithreading realization models
   - V. The thread priority in Java and in the Operating System
   - VI. Communication between processes and communication between threads
   - VII. Semafor, Mutex
2. Multithreading in Java, low level mechanisms and other issues
   - I. Thread life cycle control
   - II. Thread parameters
   - III. Thread grouping
   - IV. Handling of exceptions in threads
   - V. Monitor, „synchronized" and wait/notify
   - VI. Thread Local storage
   - VII. Timer

VIII. Starting of new processes

IX. Threads in Swing and AWT

X. Concurrent Mark-Sweep GC

XI. Memory barriers and Java memory model

XII. volatile, final, lazySet and CAS

3. Problems with multithreading programming

I. Exemplary structures of a multithreaded program

II. Race, atomic operations

III. Deadlock and Livelock

IV. Starving and Priority Inversion

V. Dekker's algorithm

4. Design patterns for multithreading programs

I. Synchronization patterns

II. Concurrency patterns

III. Initialization patterns

IV. Event management patterns

5. Concurrency library in Java 5

I. Atomic variable classes and atomic operations in Java 5

II. Lock interfaces – Lock, ReadWriteLock

III. Condition

IV. BlockingQueue

V. Thread safe collections

VI. Synchronizers

    VII. Tasks and Future

    VIII. Concurrent performance of tasks

6. Discussing of new issues in Java 7 associated with concurrence

7. Optimization of concurrent solutions

    I. Sources of decrease of performance in concurrence

    II. Reduction of lock contention

    III. Amdahl's law