

## Program szkolenia:

# Java - zagadnienia zaawansowane

### Informacje:

|                        |  |
|------------------------|--|
| <b>Nazwa:</b>          | <b>Java - zagadnienia zaawansowane</b> |
| <b>Kod:</b>            | <b>Java-SE-Java Advanced</b>           |
| <b>Kategoria:</b>      | Java Standard Edition                  |
| <b>Grupa docelowa:</b> | developerzy                            |
| <b>Czas trwania:</b>   | 3 dni                                  |
| <b>Forma:</b>          | 50% wykłady / 50% warsztaty            |

---

Program szkolenia został opracowany na podstawie doświadczeń praktyków zajmujących się na co dzień zaawansowanym programowaniem w Javie.

Szkolenie jest przeznaczone dla programistów i projektantów, którzy w swej pracy wykorzystują zaawansowane mechanizmy języka, wirtualnej maszyny oraz zmagają się z problemami ich wydajności i złożoności

### Zalety szkolenia:

- Pragmatyczne podejście
- Realne problemy

## Szczegółowy program:

### 1. Współbieżność

#### 1.1. Wstęp

##### 1.1.1. Źródła problemów

##### 1.1.2. Podejścia do ich rozwiązań

##### 1.1.3. Modele współbieżności

##### 1.1.4. Wzorce

#### 1.2. Typy atomowe

#### 1.3. Executor framework

#### 1.4. FutureTask

#### 1.5. Framework Fork/Join

#### 1.6. Locki

#### 1.7. ThreadLocal

### 2. Refleksja

#### 2.1. Obiekty klas

#### 2.2. Typy generyczne

#### 2.3. Manipulowanie obiektami

#### 2.4. Anotacje

#### 2.5. Wprowadzenie do bytcodeu

#### 2.6. Zagadnienia proxy

#### 2.7. Przykłady zastosowania

##### 2.7.1. Budowa własnych frameworków

##### 2.7.2. Narzędzia analizujące design

### 3. Struktury danych

### 3.1. Enum

#### 3.1.1. Enum jako multiton

#### 3.1.2. Visitor enumów zamiast instrukcji switch

### 3.2. Rodzaje referencji

#### 3.2.1. Weak, Phantom, Soft

#### 3.2.2. Wykorzystanie do budowy cache

#### 3.2.3. Pułapki

### 3.3. Kolejki

### 3.4. ConcurrentHashMap, WeakHashMap

## 4. IO

### 4.1. Wstęp

#### 4.1.1. Jak działa IO na poziomie systemu operacyjnego

#### 4.1.2. Problemy z wydajnością

### 4.2. RandomAccessFile

### 4.3. Channels

### 4.4. Buffers

## 5. Logowanie

### 5.1. Dobre praktyki

#### 5.1.1. Co logować

#### 5.1.2. Kiedy logować

### 5.2. Wpływ na wydajność

### 5.3. Biblioteki

### 5.4. Efektywne czytanie logów

## 6. Programowanie niskopoziomowe

### 6.1. Flagi JVM

6.2. Klasa Unsafe

6.3. Operacje Compare and Swap

6.4. Struktury off-heap

6.5. Serializacja obiektów

## 7. Problemy i ich rozwiązania

7.1. Logiczne wycieki pamięci

7.1.1. Mapy

7.1.2. Kolekcje

7.1.3. Singletony

7.2. Dokładność obliczeniowa

7.3. Czas (źródła czasu, dokładność, rozdzielczość)

7.4. Wydajność

7.4.1. Praca z obiektami String

7.4.2. Problemy z Value Objects

7.4.3. IO