

Kod szkolenia: **J/ADV**

Tytuł szkolenia: **Wydajność w języku Java**

Dni: 5

## Opis:

### Adresaci Szkolenia:

Szkolenie jest adresowane do programistów języka Java, którzy chcą poznać sekrety tworzenia wydajnego oprogramowania w tym języku.

### Cel szkolenia:

Celem szkolenia jest obycie się z praktykami tworzenia wydajnego oprogramowania w Java, poprzez tworzenie benchmark'ów i badanie wydajności, profilowanie aplikacji, używanie struktur danych odpowiednich do trybu pracy, wydajny dostęp do plików i przede wszystkim obalenie błędnych mitów dotyczących pracy Garbage Collector'a. Wielu programistów Java posiadających nawet wieloletnie doświadczenie nie zdaje sobie sprawy z tego jakie grzechy popełnia próbując pomagać wirtualnej maszynie Java, czy Garbage Collector'owi, w rezultacie otrzymując efekt przeciwny do zamierzonego. To szkolenie obala te mity i uczy prawdziwych praktyk tworzenia wydajnego oprogramowania w Javie. A także jak zwiększyć wydajność aplikacji bez dotykania samego kodu, czyli profilowanie JVM i GC.

Uczestnik szkolenia uczy się również używania zaawansowanych elementów języka takich jak refleksja, introspekcja czy dynamiczne proxy.

### Wymagania:

Od uczestników wymagana jest umiejętność programowania w języku Java.

### Parametry szkolenia:

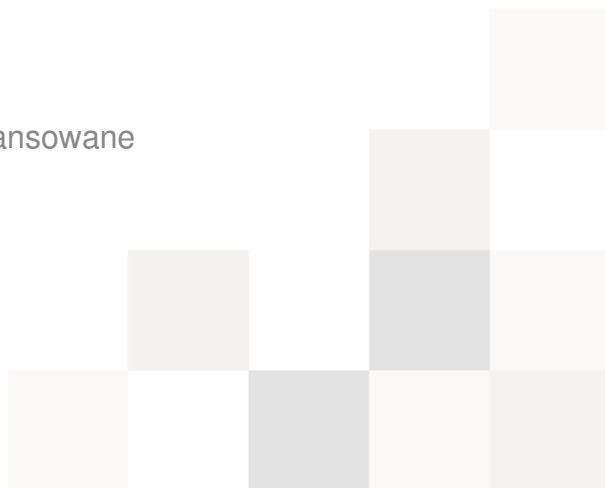
5\*8 godzin (5\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów). W trakcie warsztatów, odnajdywane są wąskie gardła aplikacji, badane aspekty wydajnościowe różnych struktur danych i sposobów dostępu do dysku, a także ćwiczone zaawansowane aspekty języka Java.

Wielkość grupy: maks. 8-10 osób.

### Program szkolenia:



1. Tworzenie wydajnego oprogramowania (1,5 h)
  - Różne wymiary wydajności
  - Proces tworzenia wydajnego oprogramowania
  - Co wpływa na wydajność w Javie
2. Pomiary wydajności (8,5 h)
  - Problemy ze zwiększaniem wydajności
  - Micro i makrobenchmarking
  - Profilowanie aplikacji
  - Przegląd narzędzi do profilowania i monitorowania
  - Użycie wybranych narzędzi (jconsole, VisualVM, JMC, MAT)
3. Wydajna praca z plikami (4,5 h)
  - Strumienie - jak zepsuć i jak naprawić
  - Swobodny dostęp i inne sztuczki
  - Kanały i bufory (NIO)
  - Odzworowywanie plików w pamięci (NIO)
  - Optymalizacja Serializacji
  - Przegląd funkcjonalny NIO2
  - Przegląd rozwiązań dla XML
4. Algorytm wydajności (0,5 h)
  - Złożoność obliczeniowa
  - Sztuka doboru rozwiązania
  - Rekurencja a iteracja
  - Nie tylko algorytm się liczy
5. Kolekcje i tablice (7 h)
  - API kolekcji z wysoka
  - Porównanie, dobór, konfiguracja i poprawne użycie: zbiorów, list, kolejek, map
  - Stare kontenery, czyli czego i czemu unikać
  - Widoki kolekcji
  - Klasa pomocnicza Collections
  - Przegląd kolekcji wielowątkowych (java.util.concurrent)
  - Tablice - wady i zalety
  - Klasa pomocnicza Arrays
  - Kolekcje na typach prostych - przegląd bibliotek
6. JMH - microbenchmark Framework (7 h)
  - Wstęp do JMH
  - Tworzenie benchmarków
  - Uruchomienie i wyniki
  - Konfiguracja benchmarków
  - Wsparcie wielowątkowości
  - Benchmarki parametryzowalne
  - Blackhole i poprawne pętle
  - Pułapki, profilersy i inne tematy zaawansowane
7. Kontrola ładowania klas (2 h)
  - Proces ładowania klas
  - Obiekty Class i jak je pozyskać
  - Jak i kiedy tworzyć ClassLoader



- Po co kontrolować ładowanie klas
  - Jak unikać zachłannego ładowania klas
  - Redukowanie ilości klas: łączenie listenerów, refleksja, invokedynamic, dynamiczne Proxy
  - Kiedy warto kontrolować ładowanie klas?
  - Wstęp do modułów (projekt jigsaw)
8. Zarządzanie pamięcią (7,5 h)
- Jak działa odśmiecanie pamięci
  - Co gwarantuje Garbage Collector?
  - Cykl życia obiektu
  - Referencje soft, weak i phantom
  - Wycieki pamięci w Javie - powody i unikanie
  - Złe praktyki przy Garbage Collector
  - Ograniczanie zajętości pamięci
  - Słaba teoria generacji, czyli skąd minor i major collections
  - Obszary pamięci: young, eden, survival spaces, old/tenured, metaspace
  - Wymiarowanie pamięci - parametry tuningowania
  - Opcje podglądu GC
  - Wybór, działanie i tuning GC: Serial, Parallel, Concurrent, G1
  - Przegląd eksperymentalnych GC: Z, Epsilon, Shenandoah
  - Inne opcje strojenia pamięci
  - Zmiany w najnowszych maszynach Java
  - Zing jako płatna alternatywa bez pauz
9. Maszyna wirtualna Javy (1 h)
- Podstawowe tryby pracy a wydajność
  - Parametry tuningowania JVM
  - Porównywanie wydajności różnych JVM
  - Przegląd wybranych macrobenchmarków: JBB2015, VolanoMark
  - Usprawnienia w Java 9-12
10. Optymalizacja a HTTP (opcjonalne)
- Kłopoty z optymalizacją serwerów
  - Przegląd przydatnych narzędzi i bibliotek:
  - HttpUnit, JWebUnit, Grinder, Apache JMeter
  - Pokaz możliwości Apache JMeter
11. Problemy z optymalizacją (0,5 h)
- Optymalizacje kompilatora a microbenchmark
  - Microbenchmark a GC
  - Uruchamianie wielu aplikacji
  - Przyzwyczajenia programistów
  - Optymalizacja za kompilator
  - Przedwczesna optymalizacja
  - Antywzorce związane z wydajnością

