BO·TT·EGA
IT minds

Program szkolenia:

Tworzenie wydajnych aplikacji w Java

Informacje:

Nazwa:	**Tworzenie wydajnych aplikacji w Java**
Kod:	**Java-wyd**
Kategoria:	Java i JVM
Czas trwania:	5 dni
Forma:	50% wykłady / 50% warsztaty

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

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.

BOTTEGA IT minds

Szczegółowy program:

1. Tworzenie wydajnego oprogramowania

- 1.1. Różne wymiary wydajności
- 1.2. Wydajność a użytkownik – subiektywne odczucie wydajności
- 1.3. Proces tworzenia wydajnego oprogramowania
- 1.4. Co wpływa na wydajność w Javie

2. Pomiary wydajności

- 2.1. Problemy ze zwiększaniem wydajności
- 2.2. Co to jest Benchmarking
- 2.3. Czym jest profilowanie aplikacji

3. Wydajna praca z plikami

- 3.1. Strumienie
- 3.2. Swobodny dostęp do plików
- 3.3. Lepsza wydajność – biblioteka NIO
- 3.4. NIO2
- 3.5. Serializacja

4. Algorytm wydajności

- 4.1. Sztuka doboru algorytmu
- 4.2. Problemy z rekurencją
- 4.3. Nie tylko algorytm się liczy

5. Kolekcje i tablice

- 5.1. API kolekcji
- 5.2. Struktury danych – podstawy wydajnych operacji
- 5.3. Zbiory

BO·TT·EGA
IT minds

5.4. Listy

5.5. Kolejki

5.6. Mapy

5.7. Stare kontenery (Java 1.0 i 1.1)

5.8. Widok kolekcji

5.9. Collections - klasa pomocnicza

5.10. Wydajność a kolekcje odporne na wielowątkowość

5.11. Tablice

6. JMH – microbenchmak framework

6.1. Czym jest JMH

6.2. @Benchmark

6.3. Przygotowanie projektu JMH

6.4. Uruchomienie JMH

6.5. Wyniki benchmarków

6.6. Dodatkowa konfiguracja benchmarków

6.7. Współbieżność

6.8. Benchmarki parametryzowalne

6.9. Blackhole i bezpieczne pętle

6.10. Pułapki

6.11. Inne frameworki

6.12. Informacje dodatkowe (opcjonalne)

7. Zarządzanie pamięcią

7.1. Java a zarządzanie pamięcią

7.2. Typy referencji a Garbage Collector i proces odśmiecania

7.3. Wycieki pamięci w Javie

BOTTEGA IT minds

7.4. Garbage Collector - złe praktyki

7.5. Ograniczanie zajętości pamięci

7.6. Opcje strojenia Garbage Collector'a

8. Maszyna wirtualna Javy

8.1. Podstawowe tryby pracy a wydajność

8.2. Parametry wydajnościowe JVM

8.3. Przegląd wybranych macrobenchmarków

8.4. Usprawnienia w Java

9. Problemy z optymalizacją

9.1. Optymalizacje kompilatora a micro-benchmark

9.2. Micro-benchmark a GC

9.3. Uruchamianie wielu aplikacji

9.4. Przyzwyczajenia programistów

9.5. Optymalizacja za kompilator

9.6. Przedwczesna optymalizacja

9.7. Antywzorze związane z wydajnością