

SZKOLENIE ZAAWANSOWANE

Wielowątkowość w języku Java

J/MT

Czas trwania: 5 dni

Kompleksowe szkolenie z zakresu wielowątkowości w Javie z użyciem API `java.util.concurrent` oraz narzędzi monitorowania i wykrywania deadlock

Cele szkolenia

.....

- Dogłębne poznanie API Java dla wielowątkowości, umiejętny dobór rozwiązań w celu optymalizacji wydajności oraz użycie narzędzi do monitoringu i wykrywania problemów wielowątkowości
- Nabycie umiejętności tworzenia aplikacji wielowątkowych począwszy od podstaw współpracy między wątkami, przez zaawansowane algorytmy busy-waiting z użyciem operacji typu CAS, na dokładnym przećwiczeniu API `java.util.concurrent` skończywszy
- Poznanie wzorców wielowątkowości z perspektywy ich pokrycia oraz braków w `java.util.concurrent` z omówieniem alternatywnych strategii. Tak, aby uczestnicy mieli świadomość, kiedy sięgać po gotowe rozwiązania, a kiedy należy tworzyć własne
- Omówienie wpływu systemów operacyjnych, optymalizacji Java (JIT, Hotspot) czy samych procesorów
- Omówienie i przećwiczenie związanych z tym problemów wielowątkowości oraz sposobów ich rozwiązywania

Zalety

.....

- Szkolenie uwzględni zarówno sposoby radzenia sobie z wielowątkowością przy użyciu gotowych klas z API `java.util.concurrent`, jak i dogłębnie omawia mechanizmy, problemy i niskopoziomowe rozwiązania dla wielowątkowości, których znajomość niezbędna jest do tworzenia własnych zaawansowanych rozwiązań
- Kameralne grupy - szkolenia technologiczne prowadzimy w grupach liczących do 8 osób. Pozwala to na indywidualne podejście oraz aktywizację każdego uczestnika
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretność umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how



Dla kogo?

- Programiści Java chcący wytwarzać aplikacje wielowątkowe

Wymagania

- Umiejętność programowania w języku Java



Program

1. Podstawy wielowątkowości
 - a. Czym jest wielowątkowość
 - b. Przetwarzanie liniowe
 - c. Przetwarzanie wielowątkowe (parallel, concurrent)
 - d. Wyłączanie wątków
 - e. Szeregowanie wątków
 - f. Współpraca między wątkami
2. Procesy
 - a. Proces a wątek
 - b. Komunikacja między wątkami
 - c. Komunikacja między procesami
 - d. Przegląd rozwiązań JEE
 - e. Niskopoziomowe API procesów w Javie
3. Wątki
 - a. Podstawy API wątków (Thread, Runnable)
 - b. Startowanie i kończenie wątków
 - c. Parametry wątków
 - d. Wyjątki i handlers
 - e. Podstawy współpracy między wątkami
 - f. Zagrożenia w aplikacjach wielowątkowych
 - g. Narzędzia monitorowania pracy wątków (JConsole, VisualVM, JMC)
4. Wykonywanie zadań
 - a. Wzmianka o starym API (Timer, TimerTask)
 - b. Co poza Runnable (Delayed, Callable, różne wersje Future)
 - c. Przegląd kolejek blokujących zadania
 - d. Wykonywanie zadań
 - e. Pule wątków
 - f. Przegląd kolejek blokujących zadania
 - g. Scheduling
 - h. Konsumpcja wyników
 - i. Fork-Join API
 - j. Flow - programowanie reaktywne
 - k. Promises” w Javie czyli CompletableFuture
5. GUI a wielowątkowość w Javie
 - a. Problemy wielowątkowości w GUI
 - b. Rozwiązania AWT
 - c. Rozwiązania Swing
 - d. Kończenie pracy
6. Operacje atomowe
 - a. Czym jest atomowość?
 - b. Problemy braku atomowości



- c. Unikanie problemów (stałe, dane lokalne)
 - d. Gdzie Java gwarantuje atomowość
 - e. Bariery pamięci
 - f. Klasy atomowe i leniwy zapis
 - g. Operacje CAS w API (`java.util.concurrent.atomic` i `java.lang.invoke`)
 - h. Pętla CAS jako alternatywa synchronizacji (spinlock)
7. Zaawansowana współpraca wątków
- a. Monitor a Lock (`java.util.concurrent.locks`)
 - b. Gotowe realizacje Lock
 - c. Lock a współpraca wątków (Condition)
 - d. Tworzenie klas blokujących
 - e. Zaawansowane spinlock'i
 - f. Inne klasy blokujące
8. Kolekcje a wielowątkowość
- a. Stare kolekcje (Hashtable, Vector, Stack)
 - b. Collections API a wielowątkowość
 - c. Cechy kolekcji wielowątkowych `java.util.concurrent`
 - d. Kolekcje blokujące
 - e. Kolekcje kopiujące przy zapisie
 - f. Kolekcje współbieżne
 - g. Porównanie wydajności `java.util.concurrent` z Collections API
9. Garbage Collector a wielowątkowość
- a. Pauza czyli Stop The World
 - b. Słaba teoria generacji i model pamięci
 - c. Parametry wydajnościowe GC
 - d. Rodzaje GC a wielowątkowość
 - e. Dobór GC
10. Asynchroniczne IO
- a. Asynchroniczna praca z plikami
 - b. Asynchroniczna praca z TCP/IP
 - c. `CompletableFuture` w `java.net.http` (wprowadzenie)
11. Wzorce wielowątkowości a gotowe rozwiązania
- a. Wzorce blokowania
 - b. Wzorce współpracy
 - c. Wzorce zarządzania wątkami
 - d. Wzorce obsługi zdarzeń
 - e. Wzorce wywołań asynchronicznych
 - f. Wzorce inicjalizacji
12. Wydajność aplikacji wielowątkowych
- a. Współpraca a odpytywanie
 - b. Sposoby ograniczania blokowania
 - c. Granica opłacalności dla Fork-Join
 - d. Ograniczenia równoleglenia - prawo Amdahla
 - e. Kontrola ilości wątków



f. Tuning Javy dla wątków

