

Program szkolenia:

Wielowątkowość w Javie

Informacje:

| | |
|----------------------|-------------------------------|
| Nazwa: | Wielowątkowość w Javie |
| Kod: | Java-wat |
| Kategoria: | Java i JVM |
| Czas trwania: | 5 dni |
| Forma: | 50% wykłady / 50% warsztaty |

Szkolenie adresowane jest do programistów Java, aplikacji wielowątkowych.

Celem szkolenia jest dokładne poznanie API Java w zakresie wielowątkowości i nabycie umiejętności posługiwania się nim. Od podstaw współpracy między wątkami, poprzez zaawansowane operacje typu CAS, na dokładnym przećwiczeniu biblioteki `java.util.concurrent` skończywszy. Także z uwzględnieniem wpływu systemów operacyjnych, czy optymalizacji Java, a nawet samych procesorów - związanych z tym problemów programowania wielowątkowego i sposobów ich rozwiązywania.

Szczegółowy program:

1. Wprowadzenie do wielowątkowości

- 1.1. Czym jest wielowątkowość
- 1.2. Szeregowanie wątków
- 1.3. Współpraca między wątkami

2. Procesy w Javie

- 2.1. Procesy a wątki
- 2.2. Niskopoziomowe API procesów w Javie
- 2.3. Przegląd rozwiązań wyższego poziomu w Java (logika zdalna)

3. Wątki w Javie

- 3.1. Tworzenie wątków w Javie
- 3.2. Parametry wątków
- 3.3. Wyjątki w wątkach
- 3.4. Wątki wirtualne
- 3.5. Podstawowa współpraca między wątkami
- 3.6. Zagrożenia w aplikacjach wielowątkowych
- 3.7. Narzędzia monitorowania pracy wątków

4. Zadania schedulowane i pule wątków

- 4.1. Stare API schedulingu
- 4.2. Różne typy zadań - czyli nie tylko Runnable
- 4.3. Wykonawca zadań
- 4.4. Pule wątków
- 4.5. Kolejki zadań - przegląd kolejek blokujących
- 4.6. Scheduling w pulach wątków

4.7. Pule wątków a konsumpcja wyników

4.8. Fork-Join API

4.9. Flow - programowanie reaktywne

4.10. "Promises" w Javie

5. GUI a wielowątkowość w Javie

5.1. Problemy wielowątkowości w GUI

5.2. Rozwiązania AWT - EventQueue

5.3. Rozwiązania SWING

5.4. Kończenie pracy

6. Operacje atomowe w Javie

6.1. Czym jest atomowość?

6.2. Problemy braku atomowości

6.3. Unikanie problemu

6.4. Gwarantowana atomowość w Javie

6.5. Bariery pamięci - opis i rodzaje

6.6. Wymuszanie barier pamięci

6.7. CAS w API Javy (`java.util.concurrent.atomic` i `java.lang.invoke`)

6.8. Spinlock - pętle CAS jako alternatywa synchronizacji

7. Zaawansowana współpraca między wątkami

7.1. Monitor a Lock (`java.util.concurrent.locks`)

7.2. Gotowe realizacje Lock

7.3. Współpraca między wątkami na Lock

7.4. Condition jako alternatywa wait-notify

7.5. Tworzenie własnych klas blokad

7.6. Inne implementacje blokad (`java.util.concurrent`)

8. Kolekcje a wielowątkowość w Javie

- 8.1. "Stare" kolekcje Java a wielowątkowość
- 8.2. Collections API a wielowątkowość
- 8.3. Kolekcje wielowątkowe java.util.concurrent
- 8.4. Kolekcje blokujące
- 8.5. Kolekcje kopiujące przy zapisie
- 8.6. Kolekcje współbieżne
- 8.7. Podsumowanie ze względu na interfejsy kolekcji
- 8.8. Porównanie wydajności java.util.concurrent z Collections API

9. Garbage Collector a wielowątkowość

- 9.1. Stop The World
- 9.2. Model Pamięci
- 9.3. Różne rodzaje Garbage Collector a wielowątkowość
- 9.4. Różne rodzaje Garbage Collector a obszary pamięci
- 9.5. Wybór Garbage Collector
- 9.6. Wskazówki doboru Garbage Collector (GC)

10. Asynchroniczne IO

- 10.1. Asynchroniczna praca na plikach
- 10.2. Asynchroniczna praca na gniazdkach
- 10.3. CompletableFuture w java.net.http (wprowadzenie)

11. Wzorce wielowątkowości a rozwiązania w Javie

- 11.1. Wzorce blokowania
- 11.2. Wzorce współpracy
- 11.3. Wzorce zarządzania wątkami
- 11.4. Wzorce obsługi zdarzeń

11.5. Wzorce wywołań asynchronicznych

11.6. Wzorce inicjalizacji

12. Wydajność aplikacji wielowątkowych w Javie

12.1. Współpraca zamiast odpytywania

12.2. Ograniczanie blokowania

12.3. Fork-Join granica opłacalności

12.4. Prawo Amdahla - czyli ograniczenia równoleglenia

12.5. Kontrola ilości wątków - pule wątków

12.6. Tuning Javy dla wątków