

Program szkolenia:

Efektywne programowanie w Javie oraz zaawansowane techniki OO oraz wzorce

Informacje ogólne

Nazwa:	Efektywne programowanie w Javie oraz zaawansowane techniki OO oraz wzorce
Kod:	Java Pro
Kategoria:	Java Standard Edition
Grupa docelowa:	projektanci, programiści
Czas trwania:	3 dni
Forma:	50% wykłady / 50% warsztaty

Szkolenie zostało opracowane z myślą o uczestnikach pragnących poszerzyć swoje kompetencje programistyczne w pragmatycznym kierunku.

Materiał został dobrany na podstawie wieloletnich doświadczeń programistów biorących udział w wielu projektach - nie jest to rodzaj bezproduktywnych łamigłówek znanych z niektórych testów certyfikacyjnych.

Zakres szkolenia został rozszerzony o zaawansowane zagadnienia Object Oriented, testowania i aspekty architektury aplikacji.

Zalety szkolenia:

- » Pragmatyczne podejście
- » Realne zastosowania
- » Zaawansowane wzorce i techniki obiektowe
- » Testowanie jednostkowe

Program szkolenia:

1. Elementy języka Java (składnia 1.5)

1.1. Adnotacje

1.1.1. Tworzenie i wykrywanie istnienia własnych adnotacji

1.1.2. Techniki budowania frameworków opartych na własnych adnotacjach

1.2. Typy wyliczeniowe

1.2.1. Typy wyliczeniowe jako pełnoprawne obiekty

1.2.2. Redukcja złożoności kodu

1.2.3. Zamiast instrukcji switch

1.2.4. Rozbudowa do wzorca Visitor

2. Typowe potrzaski podczas programowania w Javie

2.1. Wycieki pamięci - sposoby unikania

2.2. Dokładność obliczeniowa - dobór typów

2.3. Klonowanie

2.4. Poprawna implementacja equals i hashCode (również w kontekście JPA)

3. Efektywne wykorzystanie klas Biblioteki Standardowej

4. Współbieżność

4.1. Planowe wykonywanie wątków przez Executor

4.2. Kolekcje bezpieczne ze względu na wątki

4.3. Pułapki współbieżności

4.4. API Java 1.5 dla współbieżności

5. Obsługa wyjątków

5.1. Style i konwencje

5.2. Poprawne wykorzystanie wyjątków weryfikowalnych i nieweryfikowalnych

5.3. Typowe błędy podczas obsługi wyjątków

5.4. Przydatne wyjątki standardowe

6. Efektywne wykorzystanie Object Oriented

6.1. Pułapki dziedziczenia

6.1.1. Zamknięcie kodu na rozbudowę

6.1.2. Zastępowania dziedziczenia kompozycją – praktyczne zalety zmiany podejścia

6.2. Najlepsze praktyki

6.3. Code smell

7. Zestaw kilkunastu wzorców implemetacyjnych

8. Przegląd efektywnych bibliotek Apache

9. Techniki programistyczne

9.1. Praktyczne elementy GRASP i SOLID

9.2. Projektowanie otwarte na testy (testability)

10. TDD

10.1. Rodzaje testów

10.2. Narzędzia

10.3. Techniki testów jednostkowych (fake, stub, mock)

10.4. Biblioteka Mockito