

Program szkolenia:

Współczesna Java - nowości, usprawnienia i migracja istniejących rozwiązań

Informacje:

Nazwa:	Współczesna Java - nowości, usprawnienia i migracja istniejących rozwiązań
Kod:	Java-SE-modern
Kategoria:	Java Standard Edition
Odbiorcy:	developerzy, architekci
Czas trwania:	3 dni
Forma:	40% wykłady / 60% ćwiczenia

Szkolenie jest przeznaczone dla programistów i projektantów, którzy na co dzień programują w Javie i chcą być gotowi do migracji rozwiązań do nowej wersji.

Program szkolenia oprócz nowej koncepcji modułowości (projekt Jigsaw/JPMS) zawiera również zmiany i nowości w języku, kompilatorze oraz maszynie wirtualnej. Duża liczba ćwiczeń pozwoli na praktyczne zapoznanie się omawianymi zagadnieniami i gotowość do ich wykorzystania w rozwijanych projektach.

Uwaga. W przypadku migracji z Java 7 (lub starszej) zalecane jest rozszerzenie programu o nowości w Java 8 oraz programowanie funkcyjne (w zależności od wybranego zakresu dodatkowe 1 lub 2 dni) - program szkolenia "[Java 8 - myślenie funkcyjne oraz zmiany w API](#)"

Zalety szkolenia:

- Dogłębne zrozumienie nowej koncepcji modułowości
- Dostosowanie/migrację istniejących rozwiązań do najnowszej wersji Java

Szczegółowy program:

1. Przegląd kluczowych obszarów zmian i nowości

2. Zmiany w języku

2.1. prywatne metody w interfejsach

2.2. collection factory methods

2.3. process API (JEP 102)

2.4. stack walker

2.5. logging redirection

2.6. reactive streams

2.7. usprawnienia w stosie sieciowym

2.8. główne rozszerzenia i zmiany w istniejącym API

2.8.1. Optional, Stream, Date Time, Atomic, ...

2.9. zmiany w niskopoziomowym API (m.in. Unsafe)

2.10. przestarzałe (deprecated) i usunięte API

2.11. simpler AutoCloseable

2.12. deprecated warnings

2.13. safeVarargs

3. Stream Additions

3.1. iterate

3.2. takeWhile

3.3. dropWhile

3.4. optional stream

3.5. stream nullables

4. Wprowadzenie do modułowości

4.1. motywacja i ogólne założenia (JSR 376 - Java Platform Module System)

4.2. silna enkapsulacja

4.3. zależności między modułami

4.4. podział na moduły w JDK

4.5. tryby działania

4.6. narzędzia wspomagające (jdeps, ...)

5. Praca z modułami

5.1. definiowanie modułów

5.2. używanie modułów

5.3. ograniczenia w dostępie

5.4. refleksja

5.5. nienazwane moduły

6. Migracja kodu

6.1. tryb pobłażliwości

6.2. prywatne API (``sun.*``)

6.2.1. wybrane opcje migracji

6.3. praca z niezmodularyzowanym kodem

6.4. modularyzacja aplikacji

6.4.1. dobre praktyki

7. Zmiany w JVM

7.1. multi-release JAR

7.2. zmiany związane z wydajnością

7.2.1. G1 jako podstawowy algorytm Garbage Collection

7.3. zmiany w kompilatorze

7.4. zmiany w wersjonowaniu

8. Powiązane aspekty

8.1. wsparcie IDE

8.2. wsparcie systemów budowania

8.2.1. Gradle

8.2.2. Maven

8.3. budowanie mniejszych obrazów uruchomieniowych (run-time images)