BO·TT·EGA
IT minds

Program szkolenia:

Produktywne tworzenie aplikacji webowych z wykorzystaniem Groovy i Grails

Informacje:

Nazwa:	**Produktywne tworzenie aplikacji webowych z wykorzystaniem Groovy i Grails**
Kod:	**Groovy-Grails**
Kategoria:	Groovy
Grupa docelowa:	developerzy
Czas trwania:	4 dni
Forma:	35% wykłady / 65% warsztaty

Szkolenie zostało przygotowane z myślą o uczestnikach pragnących poznać zalety programowania dynamicznego na platformie JVM na przykładzie języka programowania Groovy oraz szybkiego tworzenia wydajnych aplikacji internetowych przy użyciu platformy Grails.

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

Zawiera praktyczną wiedzę z zakresu tworzenia utrzymywalnych projektów informatycznych.

Zalety szkolenia:

- Realne przygotowanie do tworzenia aplikacji w języku Groovy
- Podstawy tworzenia specyficznych języków domenowych
- Testowanie jednostkowe
- Wykorzystanie zaawansowanych technik programowania dynamicznego w testowaniu jednostkowym
- Szybkie tworzenie w pełni testowalnych aplikacji internetowych
- Podstawy poprawnego tworzenia kodu przy użyciu metodologii SOLID oraz Simple Design

BOTTEGA IT minds

Szczegółowy program:

1. Groovy jako język programowania

- 1.1. Struktura skryptu
- 1.2. Struktura programu
- 1.3. Właściwości
- 1.4. Literały list i map
- 1.5. Domknięcia
- 1.6. Literały łańcuchów znakowych
- 1.7. Interpolacja łańcuchów znakowych
- 1.8. Literały wyrażeń regularnych

2. Meta klasy i sposoby ich użycia

- 2.1. Podstawowe metody rozszerzania istniejących klas
- 2.2. Przykłady zastosowań

3. Dynamiczność w działaniu

- 3.1. Mechanizm obsługi brakujących metod
- 3.2. Mechanizm obsługi brakujących właściwości

4. GDK: rozszerzenia standardowej biblioteki uruchomieniowej

- 4.1. Rozszerzenia klasy String
- 4.2. Operacje na plikach
- 4.3. Operacje na bazach danych

5. DSL – Specyficzne języki domenowe

- 5.1. Tworzenie czytelnych plików konfiguracyjnych
- 5.2. Wczytywanie i tworzenie plików XML
- 5.3. Wzorzec „budowniczy” w kontekście języków dynamicznych

BO·TT·EGA
IT minds

5.4. Wzorzec „delegat” w kontekście domknięć

5.5. Przykłady tworzenia specyficznego języka domenowego na potrzeby opisu reguł biznesowych

6. Groovy i Java – projektowanie obiektowe

6.1. Programowanie proceduralne, obiektowe i dynamiczne

6.2. Wybrane wzorce projektowe i ich zakres stosowania w Javie i Groovy w przykładach

6.3. SOLID – czym jest zestaw regół poprawnego programowania obiektowego i jak stosować je w Groovym?

6.4. Simple Design – dlaczego i dla kogo piszemy kod aplikacji

7. Grails – wprowadzenie

7.1. Hello, word! w Grails

7.2. Omówienie podstawowych składowych platformy

7.3. Rusztowania dynamiczne i statyczne

7.4. Konsola aplikacji

8. Struktura projektu Grails

8.1. Konwencja ponad konfiguracje

8.2. Standardowe pliki konfiguracyjne

9. Konfiguracja aplikacji

9.1. Konfiguracja zewnętrznych zależności

9.2. Konfiguracja połączenia do bazy danych

9.3. Konfiguracja logowania

10. Groovy Server Pages a biblioteka

10.1. Tworzenie własnych bibliotek tagów

11. GORM – omówienie podstawowych funkcjonalności

11.1. Tworzenie obiektów domenowych

11.2. Powiązania jeden-do-wielu i wiele-do-wielu

BOTTEGA IT minds

11.3. Wyszukiwanie za pomocą dynamicznych finderów

11.4. Optymalizacja mapowań domenowych

11.5. Pułapki wydajności

12. Wzorzec „Dependency injection" w Grails

12.1. Czym jest wstrzykiwanie zależności i inwersja kontroli

12.2. Wstrzykiwanie zależności przez nazwy i po typie

12.3. Definiowanie własnych obiektów do wstrzykiwania

12.4. Zastosowanie

12.4.1. Pluginy

12.4.2. Polityki w Domain Driven Design

12.4.3. Zwiększenie testability

13. Testowanie jednostkowe poszczególnych części systemu

13.1. Testowanie kontrolerów

13.2. Testowanie serwisów

13.3. Zastępowanie (mockowanie) fragmentów systemu

14. Rozszerzanie platformy Grails za pomocą pluginów

14.1. Wprowadzenie do Spring Security

14.2. Integracja Apache Camel

14.3. Wykorzystanie bazy danych MongoDB

15. Tworzymy projekt aplikacji zarządzania magazynem w Grails

15.1. Utworzenie podstawowego modelu domenowego (magazyn, towar)

15.2. Prezentacja i edycja danych za pomocą dynamicznego rusztowania

15.3. Własny interface użytkownika

15.4. Ajax przy użyciu jQuery

15.5. Single Page Application z użyciem jQuery UI – tworzymy aplikację w stylu Web 2.0

BO·TT·EGA
IT minds

15.6. Dodawanie i usuwanie materiałów z magazynu

15.7. Powiadamianie przez email o niskim stanie danego towaru

15.8. Mobilny widok na aplikację

15.9. Przykładowa instalacjia aplikacji w chmurze CloudFoundry i Heroku