BO·TT·EGA
IT minds

Program szkolenia:

Scala - programowanie obiektowo-funkcyjne

Informacje:

Nazwa:	**Scala - programowanie obiektowo-funkcyjne**
Kod:	**Scala-Scala**
Kategoria:	Scala
Grupa docelowa:	developerzy
Czas trwania:	3-4 dni
Forma:	50% wykłady / 50% warsztaty

Szkolenie zostało opracowane z myślą o uczestnikach posiadających doświadczenie programistyczne w językach wywodzących się z C++ (Java, C#, PHP).
Podczas szkolenia nie poruszamy nieistotnych zagadnień, które rozpraszają początkujących
Skupiamy się na technikach przygotowujących do tworzenia aplikacji enterprise/webowych.

Podczas szkolenia w naturalny sposób zostały plecione narzędzia codziennej pracy oraz najlepsze praktyki tworzenia czytelnego, rozszerzalnego i testowalnego kodu.
Program został rozszerzony o najbardziej popularne frameworki: Testowanie, DSL, REST, Web.

Zalety szkolenia:

- Myślenie w stylu funkcyjnym i wzorce
- Najpopularniejsze narzędzia i frameworki
- Łatwe przejście dla programistów Java

BOTTEGA IT minds

Szczegółowy program:

1. Wstęp

- 1.1. Historia języka
- 1.2. Programowanie funkcyjne

2. Podstawy

- 2.1. Konfiguracja środowiska
- 2.2. Składnia
- 2.3. Podstawowe typy danych
- 2.4. Definiowanie funkcji
- 2.5. Definiowanie klas
  - 2.5.1. Ciało klasy jako konstruktor
  - 2.5.2. Definiowanie drugorzędnych konstruktorów
  - 2.5.3. Ograniczenia narzucane przez Scalę na konstruktory
- 2.6. Definiowanie obiektów
  - 2.6.1. Semantyka i działanie słowa kluczowego object
  - 2.6.2. Companion Object
  - 2.6.3. Object vs. Singleton
- 2.7. Operatory
- 2.8. Klasy Abstrakcyjne
- 2.9. Case class i case object
- 2.10. Zagnieżdżanie
- 2.11. Modyfikatory dostępu
- 2.12. Ekstrapolacja String
- 2.13. Type Parameters

BOTTEGA IT minds

2.13.1. Covariance oraz Contravariance

2.13.2. Różnice w traktowaniu type parameters w stosunku do Javy

2.14. Trait

2.14.1. Dlaczego nie dziedziczenie wielobazowe?

2.14.2. Linearyzacja hierarchii typów

2.15. Self Type Annotation

2.15.1. Alternatywa dla “wstrzykiwania zależności”

2.16. Leniwa ewaluacja

2.16.1. Poprzez słowo kluczowe lazy

2.16.2. Poprzez mechanizm call-by-name

2.16.3. Klasyczne pułapki oraz jak ich unikać

2.17. Implicits

2.17.1. Implicit methods

2.17.2. Implicit conversions

2.17.3. Implicit parameters

2.17.4. “Pimp my Library” pattern

2.17.5. Znaczenie IDE podczas pracy z implicitami

3. Programowanie Obiektowe

3.1. Integracja z Javą

3.2. Ducktyping

3.3. Traits

3.4. Niejawne konwersje

3.5. Parametryzacja typów i typy generyczne

4. Programowanie funkcyjne

4.1. Funkcje jako obiekty

BOTTEGA IT minds

- 4.2. Obiekty jako funkcje
- 4.3. Funkcje wyższych rzędów
- 4.4. Funkcje częściowo zaaplikowane
- 4.5. Pattern matching
- 4.6. Myślenie w stylu funkcyjnym w Scali?
  - 4.6.1. Wzorce wbudowane w język
  - 4.6.2. Realizacja założeń S.O.L.I.D. przy pomocy Scali

5. Kolekcje

- 5.1. Praca z kolekcjami w Scali
  - 5.1.1. Design biblioteki, na przykładzie List
  - 5.1.2. Jak radzić sobie z niezmiennymi kolekcjami?
- 5.2. Java Interop
- 5.3. CanBuildFrom

6. Kompilator

- 6.1. Makra
- 6.2. Wtyczki

7. Biblioteki i Frameworki

- 7.1. Lift
- 7.2. Play
- 7.3. Scalaz
- 7.4. Specs
- 7.5. Akka - biblioteka służąca tworzeniu rozproszonych / współbieżnych systemów
  - 7.5.1. Implementacja prostego systemu współbieżnego przy z zastosowaniem modelu Aktorów
  - 7.5.2. Composable Futures - ponieważ aktory bywają zbyt skomplikowane
- 7.6. AntiXml