

Язык Scala для Java-разработчиков

Код: JVA-031

Длительность: 30 ч.

Описание:

Язык Scala в последние годы набирает популярность не только в программировании, но в enterprise-разработке. К его преимуществам относятся использование самых современных нововведений в языки программирования и обширная поддержка функционального подхода к разработке. Это дает возможность повысить эффективность разработки, создавая очень краткий и эффективный код. Код на Scala компилируется в Java байт-код и выполняется внутри JVM, что обеспечивает отличную совместимость с существующими библиотеками и инструментами Java.

Слушатели получают представление о специфике и особенностях языка Scala, его преимуществах и сложных местах.

Курс включает рассмотрение операторов и типов данных Scala, изучение основных структур данных, таких как массивы, map, tuples, использование объектно-ориентированного и функционального программирования, работу с коллекциями Scala, применение pattern matching, case-классов, generic-типов и неявных преобразований.

Предлагается большое количество практических упражнений.

Цели:

Познакомить разработчиков с языком программирования Scala.

Разбираемые темы:

1. Введение в Scala.
2. Инструкции, операторы и типы:
 - Условные операторы;
 - Циклы;
 - Функции;
 - Переменные;
 - Операторы;
 - Тип Unit;
 - Типы функций;
 - Типы Null, Nothing, Any, AnyRef и AnyVal;

- Исключения;

3. Arrays, maps, tuples:

- Создание, добавление и удаление элементов массива;
- Доступ к элементам массива;
- Массивы фиксированной и вариативной длины;
- Многомерные массивы;
- Операции с массивами;
- Сортировка массива;
- Конструирование и доступ в ассоциативным таблицам;
- Итерирование, изменение, сортировка ассоциативных таблиц;
- Кортежи;
- Зиппирование;

4. Классы и наследование:

- Классы;
- Свойства с геттерами и сеттерами;
- Использование скобок при вызове методов;
- Свойства только с геттерами;
- Вспомогательные конструкторы;
- Основной конструктор;
- Вложенные классы;
- Расширение класса;
- Переопределение методов;
- Проверка и приведение типа;
- Вызов конструктора суперкласса;
- Переопределение полей;
- Анонимные подклассы;
- Абстрактные поля;
- Порядок конструирования и ранние определения;
- Иерархия наследования;
- Сравнение объектов на равенство;

5. Объекты:

- Синглтоны;
- Объекты-компаньоны;
- Объекты, расширяющие класс или трейт;
- Перечисления;
- Пакеты-объекты;
- Методы apply и unapply;
- Ленивые объекты;
- Объекты, использующие несколько трейтов;

6. Трейты:

- Трейты как интерфейсы;
- Трейты с конкретной имплементацией;
- Переопределение абстрактных методов в трейтах;
- Конкретные и абстрактные поля в трейтах;
- Порядок конструирования трейта;
- Инициализация полей трейта;
- Трейты, расширяющие классы;
- Self-типы;

7. Функции:

- Функции как переменные;
- Аргументы по умолчанию и именованные аргументы;
- Функции без возвращаемого значения;
- Анонимные функции;
- Функции с параметрами-функциями;
- Замыкания;
- Функциональные интерфейсы;
- Каррирование;
- Передача функции как параметра в виде блока кода;
- Использование return;

8. Коллекции:

- Трейты коллекций в Скала: Map, Set и Seq;
- Изменяемые и неизменяемые коллекции;
- Последовательности;
- Списки;
- Изменяемые списки;
- Множества;
- Операторы для работы с коллекциями;
- Стандартные методы работы с коллекциями;
- Использование функций map, reduce, fold и scan;
- Зиппирование;
- Итераторы;
- Потоки данных;
- Потокбезопасные коллекции;
- Параллельные вычисления в коллекциях;
- Взаимодействие с коллекциями Java;

9. Pattern matching:

- Использование паттерн матчинга;
- Переменные в паттернах;
- Паттерны-типы;

- Сравнение с массивами, списками и кортежами;
- Регулярные выражения;
- Экстракторы;
- Паттерны в определении переменных;
- Паттерны в выражениях for;
- Case классы, тип Option и partial функции;

10. Case-классы:

- Метод сору и именованные параметры;
- Защищенные классы;
- Тип Option;
- Частично определенные функции;

11. Generics:

- Дженоерик-классы;
- Дженоерик-функции;
- Ограничения типов, вариативные типы +T, -T;
- Использование _ вместо типа;

12. Неявные преобразования:

- Ключевое слово implicit для определения неявного приведения типа;
- Импортирование и использование неявных преобразований;
- Неявные параметры.

Целевая аудитория:

- Java-разработчики;
- Разработчики на других современных языках программирования (C++, C#, и т.д.).

Предварительная подготовка - общее:

Необходимо владение современным объектно-ориентированным языком программирования, желательно Java.

Рекомендуемые дополнительные материалы, источники:

- <http://www.scala-lang.org/documentation/>;
- Horstmann C. Scala for the Impatient. 2012.

Примечание:

Материалы курса представлены на английском языке.