



Scala

Tinkoff Fintech 2019

Контакты

Преподаватели:

Быков Данил

Ткачев Михаил

Telegramm канал

TFS19F Scala

SOFT

- JDK 11
- IntelliJ Idea + scala plugin

Цели курса

- Научиться писать и читать на Scala
- Получить представление о функциональном программировании
- Развить практические навыки программирования
- Познакомиться со стеком технологий
- Потренироваться работе в команде

Формат курса

- Занятия в офисе (лекция/практика)
- Домашние задания
 - Будут выкладываться в день занятия на gitlab
 - Будут оцениваться автоматизированной системой приемки
- Командный курсовой проект
- Телеграмм канал для связи с преподавателями (не злоупотребляйте!)

Как успешно сдать курс

- Выполнить домашние задания на не менее чем 80% от полного количества зачетных баллов
- Посещать занятия (не менее 10)
- Выполнить и защитить курсовой проект



Scala

Начало

Цели занятия

- Получить базовые представления о языке
- Настроить окружение
- Ознакомиться с базовыми конструкциями языка
- Написать HelloWorld.scala
- Написать игру висельник (домашнее задание)

Знакомство с языком

WIKI

Scala — мультипарадигмальный язык программирования, спроектированный кратким и типобезопасным для простого и быстрого создания компонентного программного обеспечения, сочетающий возможности функционального и объектно-ориентированного программирования.

Версии

Версии:

- 2003 - Scala 1.0
-
- Scala 2.11, Scala 2.12, Scala 2.13
- ...
- Dotty (Scala3)

Платформы:

- JVM
- JavaScript
- Native

Компиляция и runtime

- Компилируется в Java bytecode
 - Бежит в JVM, т.е. “езде”
 - Интеграция с Java
 - Переиспользование Java-инструментария
 - JIT-компиляция
- Компилируется в javascript (Scala.js)
 - Интеграция с javascript
- Компилируется в машинный код (Scala Native)
 - Интеграция с C
 - AOT-компиляция

Особенности языка

- Строгая статическая типизация
- Синтакс
- МультиПарадигма
 - OOP
 - FP

Типизация

- Статическая типизация
- Вывод типов (Хиндли-Милнер)

Синтакс

- Встроенный сахар
- Расширяемость

OOP

- вообще все в Scala программе - объект
- все вызовы являются операцией над некоторым объектом
- переопределение операций
- композиция программ с помощью классов, объектов, трейтов
- инкапсуляция / наследование / полиморфизм

FP

- функции и лямбды - граждане первого сорта
- упор на иммутабельность
- referential transparency
- ленивость и ленивые структуры данных
- все инструкции в языке являются выражениями

Почему scala

- совместимость Java
- борьба со сложностью
 - высокий уровень абстракции/композируемости
 - referential transparency
- статическая типизация + вывод типов
 - краткость
 - контроль корректности на этапе компиляции
 - упрощение рефакторинга

Среда разработки

- JDK11
- IntelliJ IDEA + scala plugin
- SBT

Репо лекции

Лекции будут размещаться в GitLab в группе **TFS19FScala**

Базовый синтаксис

Синтаксис

- литералы
- операторы
- объявление переменных (val/var/lazy)
- if
- объявление функций (def, args by name)
- while
- Tuples
- Map/List (без деталей)
- for (map/flatMap & foreach)
- try/catch
- классы трейты и объекты
- организация проекта: import/package



fintech.tinkoff.ru, 2019