

Проблемы и недостатки

- Часто идут дебаты на тему «Scala is too complex»
- Да, Scala вряд ли подойдёт для компании, построенной по принципу «100 китайцев рано или поздно напишут»
- Но практика показывает, что лучшие программисты могут быть во много раз эффективнее, чем средние, Scala – это одновременно и путь и инструмент к такого рода решениям

Проблемы и недостатки

- Scala всё ещё достаточно молодой язык – поэтому «best practices» ещё не все выработаны и не расписаны в десятках книг
- Несложно написать программу с очень плохими показателями по производительности, нужно разбираться и профилировать
- IDE всё ещё не на уровне Java, хотя и колоссально продвинулись за последний год

Typesafe

- Уже около 30 специалистов высочайшего класса
- Typesafe stack: Akka, SBT, Play, IDE
- Консультации, тренинги, конференции
- Прямой контакт с компаниями, использующими Scala в production

Профессиональный уровень

- Многие Java-библиотеки используют элементы функционального подхода (Spring templates, Wicket, Guava и многие другие)
- В Scala многие привычные «design patterns» и «best practices» – часть языка, после программирования на Scala, Вы получите более широкий взгляд на Java и OOP
- Oracle развивают Java, планируют добавить новые концепции (например, лямбда-выражения). Знание Scala может сильно помочь в освоении и понимании сути и ограничений этих концепций

Ресурсы

- Офф. сайт: <http://www.scala-lang.org/>
- Google groups:
<https://groups.google.com/forum/?fromgroups=#!forum/scala-user>
- Twitter: <https://twitter.com/search?q=%23scala>
- This week in #Scala:
<http://www.cakesolutions.net/teamblogs/tag/scala/>
- <http://habrahabr.ru/hub/scala/>

Что такое Scala?

- Полностью открытая (**open source**) технология
- Эффективный, ёмкий язык для профессионалов
- Единственный популярный язык, гармонично объединяющий статическую типизацию, **FP** и **OOP**
- Промышленный язык, широко применяющийся на практике в серьёзных проектах и имеющий полноценную коммерческую поддержку
- Язык (вместе с платформой **JVM**), нацеленный на решение самых насущных проблем ближайшего будущего: параллельные и распределённые вычисления, нагруженные системы