Basic Scala

prepared by Vadim Shalts

5 августа 2013 г.

Содержание

1 Цель курса

Познакомиться с основами языка Scala. Получить представление о функциональных возможностях языка и получить практический опыт использования этих возможностей. Научиться использовать Scala REPL и другие инструменты из состава Scala.

2 Описание курса

Курс состоит из нескольких тем. Каждая тема имеет теоретическую часть и может включать набор упражнений для закрепления пройденного материала.

Курс является базовым и потому не включает в себя "сложные" элементы Scala. В курсе будут данны ссылки на материал для более углубленного изучения Scala, чтобы заинтересованные могли продолжить изучение Scala самостоятельно.

Хотя данный курс базируется на курсе Coursera "Functional Programming Principles in Scala", часть заданий была изменена и дополнена. Кроме того курс на Coursera рассчитан на 35-50 часов, в то время как данный курс исходит из 80-ти часовой нагрузки. Данный курс не содержит собственных видеоуроков, а потому предполагает гораздо больший объем самостоятельной работы.

Курс охватывает такие темы:

- Scala Overviews ans Scala Tools
- Functions and Closures (include Higher Order Functions)
- Classes and Objects
- Composition and Inheritance. Scala's Hierarchy
- Traits
- Case classes and Pattern Patching
- Type Parametrizations
- Collections
- For Expressions
- Lazy Evaluation

Упражнения к темам будут даны в виде заданий и шаблонов решений с тестами, где нужно будет исправить/доработать шаблоны так, чтобы они стали полными и правильными решениями заданий.

Данный курс не включает работу с актерами, аннотациями, XML и некоторыми другими частями базовой Scala. Предполагается, что изучающие курс знают один из языков OOP (C++, Java, C#). Опыт работы с Java будет очень полезен.

По окончанию курса все упражнения будут проверены на правильность выполнения и стиль. Для прохождения курса нужно иметь не меньше 80% заданий, решенных без нарекания к правильности и стилю.

3 Общие материалы для изучения

Здесь описаны ресурсы, которые будут полезны на протяжении всего курса. Эти ресурсы описывают больше одной темы Scala и могут изучаться частями по мере необходимости. В рамках каждого блока будут даваться уточнения: какие именно части данных ресурсов желательно прочитать для успешного освоения блока. Также в блоках могут встречаться уникальные ресурсы, которые описывают или разъясняют только тему конкретного блока.

Ресурсы обязательные для изучения:

- Learning Scala -http://www.scala-lang.org/node/1305
- Tutorial indroducing the main concepts of Scala - http://docs.scala-lang.org/tutorials/
- Scala Style Guide -http://docs.scala-lang.org/style/
- A Scala tutorial by Twitter -http://twitter.github.com/scala_school/

Очень желательные для изучения частично или полностью (не включены в обязательные в связи с ограниченностью курса по времени):

- Programming in Scala - http://www.artima.com/shop/programming_in_scala_2ed
- Effective Scala by Twitter - http://twitter.github.io/effectivescala/
- Scala By Example - http://www.scala-lang.org/docu/files/ScalaByExample.pdf

Интересные ресурсы для ознакомление с Scala (особенно если есть знание Java):

- Scala for the Intrigued - http://www.youtube.com/watch?v=grvvKURwGNg
- Scala Versus Java - http://www.youtube.com/watch?v=PKc5IwHG68k

- ScalaDays 2013 Keynote - Scala with Style - http://parleys.com/play/51c1994ae4b0d38b54f4621b/chapter0/about

Рекомендованные ресурсы для углубленного изучения (повышенный уровень сложности или же материал не входит в программу курса. Будут полезны для углубления понимания, но не являются необходимыми для прохождения курса):

- Scala in Depth - http://www.amazon.com/Scala-Depth-Joshua-D-Suereth/dp/1935182706
- Functional Programming in Scala - http://www.manning.com/bjarnason/
- NEScala Keynote: Effective Scala with Josh Suereth - http://www.youtube.com/watch?v=YZxL0alO1yc
- Simplicity In Scala Design - http://www.youtube.com/watch?v=Pqh_QFFzIlI
- Scala Overview on StackOverflow: A list of useful questions sorted by topic - http://stackoverflow.com/tags/scala/info

4 Подготовка среды

До того, как приступить к выполнению задания, нужно:

1. Установить Scala 2.10.x (используйте "Current Stable Release"). Скачать Scala можно с официального сайта: http://www.scala-lang.org/downloads. Для упрощения работы со Scala желательно прописать её в PATH

2. Скачать и установить SBT согласно инструкций с официального сайта http://www.scala-sbt.org/release/docs/Getting-Started/Setup.html. SBT также желательно прописать в PATH

3. Выбрать с установить одну из IDE.
   Выполнить курс можно в любой из IDE, которая для вас более удобна или знакома. Шаблоны позволяют сгенерировать проектные файлы для IntelliJ IDEA и Eclipse. Для прохождения курса рекомендуется использовать Eclipse, чтобы не сталкиваться с некоторыми проблемами в поддержке Scala со стороны IntelliJ IDEA. В дальнейшем предполагается, что вы выбрали Eclipse для прохождения курса.

   Scala IDE на базе Eclipse находиться тут: http://scala-ide.org/
   Scala IDE на базе IntelliJ IDEA Conunity Edition тут: http://www.jetbrains.com/idea/download/

Для работы из IDEA также понадобиться Scala Plugin. Подробности установки и настройки плагина находятся на официальной странице: http://confluence.jetbrains.com/display/SCA/Getting+Started+with+IntelliJ+IDEA+Scala+Plugin.

Для удобной работы с тестами из Eclipse рекомендуется установить плагин ScalaTest - http://www.scalatest.org/user_guide/using_scalatest_with_eclipse.

Будет также полезно использовать Scala Worksheet (существует как для Eclipse так и для IDEA), но это не является требованием.

5 Учебные блоки и упражнения

5.1 Scala Overviews ans Scala Tools

5.1.1 Цель

Цель данного блока в том, чтобы вы получили начальное представление о возможностях языка и освоили базовые инструменты Scala разработчика.

Вы должны научиться работать с одной из IDE, поддерживающих Scala. Также вы должны освоить работу с Scala REPL. Понять базовую структуру Scala проектов. Научиться использовать базовые команды SBT для компиляции, тестирования и запуска Scala кода.

5.1.2 Материалы для блока

Обязательные метериалы:

- Learning Scala - http://www.scala-lang.org/node/1305
- Scala: A Scalable Language - http://www.artima.com/scalazine/articles/scalable-language.html
- Introduction to the Scala REPL - http://www.youtube.com/watch?v=tRD8hhMxrVI
- Very short SBT Tutorial - https://gist.github.com/wilmoore/3748139
- Прочитать части из SBT Getting Started (http://www.scala-sbt.org/release/docs/Getting-Started/index.html):
  - Setup
  - Hello, World
  - Directory structure
  - Running

Рекомендованные материалы:

- Scala Style Guide - http://docs.scala-lang.org/style/
- С 1-й по 3-ю главы из “Programming in Scala 2-nd Edition”
- Learning Scala by Joel Abrahamsson - http://joelabrahamsson.com/learning-scala/
- Видео лекции из секции Getting Started курса “Functional Programming Principles in Scala” на coursera
- Zero to a Scala SBT Project - http://www.bks2.com/blog/2013/02/23/zero-to-a-scala-sbt-project/
- Настроить и использовать Scala Worksheet - https://github.com/scala-ide/scala-worksheet/wiki/Getting-Started

Необязательный материал повышенной сложности для углубленного изучения вне курса:

- SBT Introduction & Cookbook for Scala -http://www.youtube.com/watch?v=V2rl62CZPVc

5.1.3 Задание

Просмотрите и прочитайте обязательный материал для данного блока. Задание для данного блока находится в папке assignments/example. Запустите SBT из каталога example.

Для начала вам нужно сгенерировать файлы проекта для IDE, в которой вы будете работать. Для этого нужно из SBT выполнить команду eclipse

```
$ sbt
> eclipse
```

или gen-idea, в зависимости от выбранной IDE.

```
$ sbt
> gen-idea
```

Запустите вашу IDE и откройте сгенирированнй проект.

В проекте присутствуют файлы Lists и ListsSuite. Lists содержит описание двух функций, которые вам нужно реализовать:

```
// This method computes the length of the list xs.
def length(xs: List[Int]): Int = ???
```

и

```
// This method returns the smallest element in a list of integers.
def min(xs: List[Int]): Int = ???
```

А ListsSuite содержит примеры тестов и тесты для функций length и min. Вам нужно реализовать функции, заменив ??? на правильный Scala код. Кроме того вам нужно исправить тесты так, чтобы они стали верными. Дополните тесты своими для более качественного тестирования функций length и min. Реализуя задание, постарайтесь не пользоваться циклами while. Вместо этого используйте рекурсию.

Проверьте ваше решение на правильность стиля. В этом вам может помочь команда scalastyle

```
$ sbt
> scalastyle
```

scalastyle проверяет не все аспекты стиля. Отсутствие предупреждений необязательно означает правильно выполненное задание. Используйте scalastyle только как вспомогательный инструмент диагностики вашего решения.

5.2 Functions and Closures

5.2.1 Цель

Цель этого блока – освоить работу с функциями в Scala. Научиться использовать функции как минимальные элементы абстракции. Научиться использовать рекурсию и хвостовую рекурсию. Понять и научиться использовать функции высшего порядка. Разбросаться с Currying, Partially Applied функциями, замыканиями и передачей параметров по имени и по значению.

5.2.2 Материалы для блока

Рекомендованные материалы:

- 8-я и 9-я главы из “Programming in Scala 2-nd Edition”
- Видео лекции из секций Functions & Evaluations и Higher Order Functions “Functional Programming Principles in Scala” на coursera
- Basics из Twitter Scala School
- Functional programming principles in Scala – Week 1: Functions & evaluations - http://www.albertmata.net/2012/11/13/functional-programming-principles-in-scala-week-1-functions-evaluations/
- Functional programming principles in Scala – Week 2: Higher order functions - http://www.albertmata.net/2012/11/13/functional-programming-principles-in-scala-week-2-higher-order-functions/

5.2.3 Задание

Задачи взяты из курса "Functional Programming Principles in Scala" coursera (Week 1: Functions & Evaluations и Week 2: Higher Order Functions). Права на тексты заданий и код этого задания принадлежат coursera и автору задания Martin Odersky.

Задачи данного блока находятся в папке assignments/functions. Ознакомиться с описанием задания можно на сайте coursera или используя данные копии описания заданий:

- Week 1: Functions & Evaluations - https://www.evernote.com/shard/s46/sh/09ff5a4b-bd7c-4490-8677-5e9dd7dfd50d/d474b42e6da5b758056b7db1c90881b3
- Week 2: Higher Order Functions - https://www.evernote.com/shard/s46/sh/4204f1eb-03ea-4c66-96f8-c1ff1af0782b/2362c14ada2e1fdb6985ff2921142857

Для всех заданий нужно реализовать тесты и проверить стиль посредством scalastyle

5.3 Classes and Objects

5.3.1 Цель

В рамках этого блока нужно разобраться с понятиями класса, объекта и метода Scala. Изучить как в Scala определяются поля класса, конструкторы и как контролируется уровень доступа к членам класса. Изучить операторы в Scala. Разобраться с перегрузкой методов.

5.3.2 Материалы для блока

Рекомендованные материалы:

- 4-я и 6-я главы из “Programming in Scala 2-nd Edition”
- Видео лекции из секций Data and Abstraction “Functional Programming Principles in Scala” на coursera
- Basics continued из Twitter Scala School
- Learning Scala part four – Classes and Constructors - http://joelabrahamsson.com/learning-scala-part-four-classes-and-constructors/
- Learning Scala part five – Methods - http://joelabrahamsson.com/learning-scala-part-five-methods/
- Learning Scala part nine – Uniform Access - http://joelabrahamsson.com/learning-scala-part-nine-uniform-access/
- Functional programming principles in Scala – Week 3: Data & abstraction - http://www.albertmata.net/2012/11/13/functional-programming-principles-in-scala-week-3-data-abstraction/

5.3.3 Задание

Задачи взяты из курса "Functional Programming Principles in Scala" coursera (Week 3: Data and Abstraction). Права на тексты заданий и код этого задания принадлежат coursera и автору задания Martin Odersky.

Задачи данного блока находятся в папке assignments/classes. Ознакомиться с описанием задания можно на сайте coursera или используя данные копии описания заданий:

- Week 3: Data and Abstraction - https://www.evernote.com/shard/s46/sh/cd512dff-d0b3-48b2-957e-0debe578e8d8/993e94074020ed787e7967deade55f32

Для всех заданий нужно реализовать тесты и проверить стиль посредством scalastyle

5.4 Composition and Inheritance. Scala's Hierarchy

5.4.1 Цель

В рамках данного блока вам предстоит понять как ключевые принципы ООП реализуются в Scala. Нужно разобраться со способами реализации наследования и композиции в Scala. Изучить абстрактные классы и переопределение методов (overriding). Понять как реализуется полиморфизм в Scala. Изучить иерархию классов Scala, включая особые случаи: `scala.Any`, `scala.AnyRef`, `scala.AnyVal`, `scala.Null` и `scala.Nothing`. Разобраться с реализацией сравнения объектов в случае наследования: `equals`, `canEquals`, `hashCode`.

5.4.2 Материалы для блока

Рекомендованные материалы:

- 10-я, 11-я, 20-я и 30-я главы из "Programming in Scala 2-nd Edition"
- Видео лекции из секций Data and Abstraction и секции Types and Pattern Matching курса "Functional Programming Principles in Scala" на coursera
- Learning Scala part eight – Scala's type hierarchy and object equality - http://joelabrahamsson.com/learning-scala-part-eight-scalas-type-hierarchy-and-object-equality/

Необязательный материал повышенной сложности для углубленного изучения вне курса:

- The Typeclass Pattern - An Alternative to Inheritance - http://www.youtube.com/watch?v=yYoOgANYViE
- Tutorial: Typeclasses in Scala with Dan Rosen - http://www.youtube.com/watch?v=sVMES4RZF-8

5.5 Traits

5.5.1 Цель

Требуется разобраться с идеей Traits, как обобщенных интерфейсов. Обратить внимание на отличия Traits от обычных классов. Понять, в чем отличие использования Traits от классического множественного наследования (C++). Изучить правила линеаризации для Traits. Разобраться с использованием Traits для декомпозиции приложения на примере Cake Pattern.

5.5.2 Материалы для блока

Рекомендованные материалы:

- 12-я глава из "Programming in Scala 2-nd Edition"
- Видео лекции из секций Data and Abstraction "Functional Programming Principles in Scala" на coursera
- Learning Scala part seven – Traits - http://joelabrahamsson.com/learning-scala-part-seven-traits/
- Basics continued из Twitter Scala School - http://twitter.github.io/scala_school/basics2.html
- Real-World Scala: Dependency Injection - http://jonasboner.com/2008/10/06/real-world-scala-dependency-injection-di/
- DI in Scala: Cake Pattern pros & cons - http://www.warski.org/blog/2011/04/di-in-scala-cake-pattern-pros-cons/

Необязательный материал повышенной сложности для углубленного изучения вне курса:

- Cake Pattern: The Bakery from the Black Lagoon - http://www.youtube.com/watch?v=yLbdw06tKPQ
- Dependency Injection in Scala: Cake Pattern and Typeclasses - http://vimeo.com/23421254
- Component Based Dependency Injection in Scala - http://scabl.blogspot.com/2013/02/cbdi.html

5.6 Case classes and Pattern Matching

5.6.1 Цель

В данном блоке вам следует разобраться с тем, чем case классы отличаются от обычных классов. Изучить какие методы генерирует Scala для case классов и как ими пользоваться (`copy`, `toString`, `equals`, `hashCode`). Разобраться с использованием pattern matching в Scala. Познакомиться с применением патерна `Option`.

5.6.2 Материалы для блока

Рекомендованные материалы:

- 15-я глава из “Programming in Scala 2-nd Edition”
- How does a case class differ from a normal class? - http://www.scala-lang.org/node/258
- Видео лекции из секций Types and Pattern Matching “Functional Programming Principles in Scala” на coursera
- Playing with Scala’s pattern matching - http://kerflyn.wordpress.com/2011/02/14/playing-with-scalas-pattern-matching/

Необязательный материал повышенной сложности для углубленного изучения вне курса:

- 26-я глава из “Programming in Scala 2-nd Edition”

5.6.3 Задание

Задачи взяты из курса "Functional Programming Principles in Scala" coursera (Week 4: Types and Pattern Matching). Права на тексты заданий и код этого задания принадлежат coursera и автору задания Martin Odersky.

Задачи данного блока находятся в папке assignments/patmat. Ознакомиться с описанием задания можно на сайте coursera или используя данные копии описания заданий:

- Week 4: Types and Pattern Matching - https://www.evernote.com/shard/s46/sh/f7290371-f90a-498b-8f42-7f081fe456b3/a3a2c9a1711b105d8b30257cf42e719a

Для всех заданий нужно реализовать тесты и проверить стиль посредством scalastyle

5.7 Type Parametrizations

5.7.1 Цель

В рамках этого блока следуем изучить особенности работы с обобщенными типами Scala. Нужно разобраться с различными видами ограничений на типы: верхняя и нижняя границы. Изучить различные виды вариации типов: Invariant, Covariant, Contravariant. Следует уделить внимание тому, как поведение Scala отличается от поведения Java в аналогичных ситуациях (на примере массивов).

5.7.2 Материалы для блока

Рекомендованные материалы:

- 19-я глава из “Programming in Scala 2-nd Edition”
- Covariance and Contravariance in Scala - http://blogs.atlassian.com/2013/01/covariance-and-contravariance-in-scala/
- wikipedia: Covariance and contravariance - http://en.wikipedia.org/wiki/Covariance_and_contravariance_(computer_science)
- Type & polymorphism basics из Twitter Scala School - http://twitter.github.io/scala_school/type-basics.html
- Видео лекции из секций Types and Pattern Matching “Functional Programming Principles in Scala” на coursera

5.8 Collections

5.8.1 Цель

В рамках блока нужно изучить базовые коллекции: Sequences, Sets, Maps. Изучить иерархию классов коллекций: `Traversable`, `Iterable`, `Seq`, `Set`, `Map`. Уяснить различиях между применением мутабельных и иммутабельных коллекций. Изучить наиболее часто применяющиеся операции для коллекций. Разобраться с кортежами (tupples), массивами и строками. Изучить работу с итераторами. Обратить внимание на скоростные характеристики разных видов коллекций.

5.8.2 Материалы для блока

Рекомендованные материалы:

- 16-я, 17-я и 24-я главы из “Programming in Scala 2-nd Edition”
- Collections из Twitter Scala School - http://twitter.github.io/scala_school/collections.html
- Collections из Twitter Effective Scala - http://twitter.github.io/effectivescala/index.html#Collections
- Collections by Martin Odersky, and Lex Spoon - http://docs.scala-lang.org/overviews/collections/introduction.html
- Видео лекции из секций Lists и Collections “Functional Programming Principles in Scala” на coursera

Необязательный материал повышенной сложности для углубленного изучения вне курса:

- 25-я главы из “Programming in Scala 2-nd Edition”
- The Architecture of Scala Collections - http://docs.scala-lang.org/overviews/core/architecture-of-scala-collections.html
- Parallel Collections - http://docs.scala-lang.org/overviews/parallel-collections/overview.html

5.8.3 Задание

В данном блоке вам нужно будет найти максимальную сумму для треугольника из чисел. Текст задания можно прочитать тут: http://projecteuler.net/problem=67

У данной задачи существует оптимальное ранение, которое пропорционально по времени количеству чисел в треугольнике. Ваша задача найти и реализовать такое решение. Задание нужно реализовать используя списки (List) и не используя другие виды коллекций (Array, Vector, etc). Списков достаточно для построения оптимального решения этой задачи.

Вам нужно реализовать функции определенные в MaximumPathSum так, чтобы тесты исполнялись без ошибок.

Треугольник описан как список списков (примеры в PathData). Вы можете рассчитывать, что исходные данные всегда заданы верно. Проверять правильность исходных данные не нужно.

Обратите внимание на определение пути до некоторого числа в треугольнике:

```
// sum - is maximum sum of numbers along this path.
// value - is last number in this path.
// previous - is link to previous number from this path.
case class Path(sum:Long, value: Int, previous: Option[Path])
```

Используя это определение реализуйте:

```
// Extends previous path by one number of triangle
def extendPath(newValue: Int, previous: Option[Path]): Path = ???

// Find optimal solution (path with maximum sum) for every number from the bottom line of
    triangle.
// You can split this function to several ones.
def calculateOptimalPaths(data:List[List[Int]]): List[Path] = ???

// Return list of number along the path (staring from top of triangle to bottom).
def numbersForPath(path:Option[Path]): List[Int] = ???

// Select path that have maximum sum. Should return empty list if paths is empty list.
def bestPath(paths:List[Path]): Option[Path] = ???

// Return sum for best path from paths. Should return zero if paths is empty list. Reuse
    bestPath to find best path.
def bestSum(paths:List[Path]): Long = ???

// Return number along best path. Should return empty list if paths is empty list. Reuse both
    bestPath and numbersForPath.
def bestSumNumbers(paths:List[Path]): List[Int] = ???
```

5.9 For Expressions

5.9.1 Цель

Изучить синтаксис for выражения. Разобраться с синтаксическим сахаром, который применяет компилятор при трансляции for выражений. Научиться самостоятельно преобразовывать функции высшего порядка (map, flatMap, withFilter) в for выражения и обратно. Понять для чего и как используется yield в for выражении.

5.9.2 Материалы для блока

Рекомендованные материалы:

- 23-я глава из “Programming in Scala 2-nd Edition”

- Learning Scala part six – If statements and Loops - http://joelabrahamsson.com/learning-scala-part-six-if-statements-and-loops/

5.9.3 Задание

Задачи взяты из курса "Functional Programming Principles in Scala" coursera (Week 6: Collections). Права на тексты заданий и код этого задания принадлежат coursera и автору задания Martin Odersky.

Задачи данного блока находятся в папке assignments/forcomp. Ознакомиться с описанием задания можно на сайте coursera или используя данные копии описания заданий:

- Week 6: Collections - https://www.evernote.com/shard/s46/sh/58635581-eddd-461d-a8a2-1ed943fef52f/0d930c90210a476404abb273dea1d1c5

Для всех заданий нужно реализовать тесты и проверить стиль посредством scalastyle

5.10 Lazy Evaluation

5.10.1 Цель

Разобраться что такое ленивые вычисления и зачем они нужны. Изучить какие средства для ленивых вычислений присутствуют в Scala. Изучить применения модификатора lazy. Изучить работу с потоками (Streams) и представлениями (views).

5.10.2 Материалы для блока

Рекомендованные материалы:

- Functional programming principles in Scala – Week 7: Lazy evaluation - http://www.albertmata.net/2012/11/13/functional-programming-principles-in-scala-week-7-lazy-evaluation/
- Видео лекции из секций Lazy Evaluation “Functional Programming Principles in Scala” на coursera

Необязательный материал повышенной сложности для углубленного изучения вне курса:

- Play Iteratees: Handling data streams reactively - http://www.playframework.com/documentation/2.0/Iteratees
- Understanding Play2 Iteratees for Normal Humans - http://mandubian.com/2012/08/27/understanding-play2-iteratees-for-normal-humans/
- Iteratees for imperative programmers - http://jazzy.id.au/default/2012/11/06/iteratees_for_imperative_programmers.html

5.10.3 Задание

Задачи взяты из курса "Functional Programming Principles in Scala" coursera (Week 7: Lazy Evaluation). Права на тексты заданий и код этого задания принадлежат coursera и автору задания Martin Odersky.

Задачи данного блока находятся в папке assignments/collections. Ознакомиться с описанием задания можно на сайте coursera или используя данные копии описания заданий:

- Week 7: Lazy Evaluation - https://www.evernote.com/shard/s46/sh/58635581-eddd-461d-a8a2-1ed943fef52f/0d930c90210a476404abb273dea1d1c5

Для всех заданий нужно реализовать тесты и проверить стиль посредством scalastyle