

2018年2月13日

株式会社インプレスR&D

<https://nextpublishing.jp/>

技術書典シリーズ最新刊！

『Scalaをはじめよう！ —マルチパラダイム言語への招待—』発行

Scala(スカラ)はこわくない！

インプレスグループで電子出版事業を手がける株式会社インプレスR&Dは、『Scalaをはじめよう！ —マルチパラダイム言語への招待—』(著者:伊藤 竜一)を発行いたしました。

Scalaをはじめよう！ —マルチパラダイム言語への招待—』

<https://nextpublishing.jp/isbn/9784844398127>



著者:伊藤 竜一

小売希望価格:電子書籍版 1600円(税別)／印刷書籍版 1800円(税別)

電子書籍版フォーマット:EPUB3／Kindle Format8

印刷書籍版仕様:B5判／カラー／本文 86ページ

ISBN:978-4-8443-9812-7

発行:インプレス R&D

<< 発行主旨・内容紹介 >>

【マルチパラダイム言語・Scala の、プログラマ向け入門書！】

本書はオブジェクト指向と関数型の特徴を併せ持つマルチパラダイム言語 Scala(スカラ)のプログラマ向け入門書です。

「コードが難しい」「コンパイルが遅い」といった Scala に対する誤解を解き、JVM 上でバイナリが動作し、Java のライブラリをシームレスに利用可能な特徴を活かしたプログラミングを行うためのガイドブックとしておすすめします。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

Scala を学習するために必要な環境、参考になる情報源なども掲載

第1章 Scalaの世界へようこそ

1.1 この本について

この本は、あまり難しいことを考えずに一通りScalaを触れるようになることを目的としています。簡単なサンプルコードをベースにゆる〜く、でも出来る限りScalaの本質を損なわないように構文や機能を紹介します。流れに沿って必要十分なスピードで進んでいくため、辞書的に利用するのではなく順を追って読むことをおすすめします（サンプルコード内のコメントにも重要な情報があります）。Scalaやその他の小難しい関数型プログラミング等々の事前知識は一切必要ありませんが、Javaまたは相当するオブジェクト指向プログラミング言語のある程度理解していることを前提とします。

1.2 Scalaとは？

Scala (スカラ) はオブジェクト指向言語と関数型言語の特徴を統合したマルチパラダイムのプログラミング言語である。名前の「Scala」は英語の「scalable language」に由来するものである。

Wikipediaではこのように紹介されています。2.12系がメインストリームで、最新のリリースは2.12.4となっています（2018年2月現在）。本書では2.12系を前提に進めます。本題に入る前にScalaの「推しポイント」をいくつか紹介します。

- Scalaはオブジェクト指向と関数型の特徴を併せ持つプログラミング言語です。関数型は難しく敬遠してしまう方も多いと思いますが、Scalaでは**関数型の知識をほとんど必要とせずに思惑だけを得ることができる**のです。ベースはオブジェクト指向ですので、Javaの延長としてプログラミングしながら関数型由来の便利ツールで楽をすることができます。
- 構文・標準ライブラリが共に洗練されており、簡潔なコードを記述できます。強い静的な型付けにより多くの**バグをコンパイル時に発見しやすい**というメリットと、照準論や襷衣構文といった補助機能により、RubyやPythonといった**スクリプト言語のようにスッキリとしたコーディングが可能**というメリットを併せ持ちます。他の言語ではもどかしかったことがScalaでならサクッとできちゃいます。
- JVM上で動作するため、**様々なプラットフォームで同じバイナリを利用できます**。また、Javaのライブラリ資源もオーバヘッドなくシームレスに利用できます。逆にJavaからScalaのコードを呼び出すことも可能なため、プロジェクトの一部にだけScalaを利用する

といったこともできます。

・**必要なツールはsbt（とJDK）だけ**です。「コンパイラは？言語仮想環境構築ツールは？ビルドツールは？」とあれこれ迷う必要はありません。sbtひとつでビルドやライブラリの管理はもちろん、REPLと呼ばれる対話的な実行環境やプロジェクトで利用するScalaのバージョン管理まで可能です。ちなみに最近ではScalaをネイティブ環境（LLVM）やJavaScript環境（as an AltJS）向けにコンパイルして利用できる環境も整いつつあります。

Scalaの良いところばかりを挙げましたが、一方でScalaについてよく誤解されることがあるので、こちらも触れておきたいと思います。既にScalaを調べたことがある方であれば「Scalaのコンパイルは遅い」という噂を耳にしたことがあるのではないのでしょうか。これは一部正解ですが、誤解によるものが多く含まれます。コンパイル時に必要な処理が多く結果として比較的遅いことは事実ですが、それ以上にビルドツールの誤った利用法によりコンパイル速度が低下している場合が多いです。正しく利用していればそれほどたつくことはないでしょう。

また、コンパイル速度自体も日々改善が進んでいます。ビルドについて詳しくは2章で取り上げます。もう1つ、「Scalaのコードは難しい」と先入観を持たれることがあります。実際には基本的な構文（特に一部の構文）さえ知ってしまえば不自由することはないはずで、その量も少なく抑えられています。コーディングの指針としては、「基本はオブジェクト指向であること」「出来る限り値を不変にすること」の2つだけ意識していればとらえず十分でしょう。これらについても本書の中で追い追い紹介していきます。

さて、Scalaを始める前にこのような誤い文句をいくつも並べても実感が湧かないと思いますので早速本編に入っていきます。

【コラム】参考になるウェブサイト・本

■おすすめ

Scala Standard Library API

<http://www.scala-lang.org/api/current/index.html>

Scala標準ライブラリの公式ドキュメント。Google先生が苦手な記号でも検索可能。

Scala Tutorials

<http://docs.scala-lang.org/tutorials/>

Scala公式のチュートリアル。項目ごとに列挙されている。かなりニッチで応用的な項目も絡んでいるので注意。

ひしだま's 技術メモ Scala

<http://www.np.jp/asahi/hishidama/home/tech/scala/index.html>

網羅的にまとめられている。

実際にScalaでプログラミングする過程を、いわゆる「FizzBuzz」をつかって解説

第6章 FizzBuzzを作ってみよう

パラメータ多相・コレクション・関数

4章ではFizzBuzzを出力してみました。この章では同じFizzBuzzを作っていきます。具体的には、1からnの整数リストを作成し、そこからFizzBuzzを施した文字列リストに変換していきます。

これまでのサンプルコードで度々出てきていましたが、Scalaの標準ライブラリから多数のコレクション（配列やリストといった多数の値を扱うデータ構造）とそのコレクションを操作するためのメソッドが提供されているので利用していこうと思います。

6.1 コレクションを扱う前に……パラメータ多相

その前にコレクションやその他でも多く利用されているScalaのパラメータ多相について紹介します。パラメータ多相とは、例えばリストという1つのデータ構造で整数リストや文字列リストといった様々な要素に対応するために利用される機能です。ScalaではList[Int]やList[String]のように[]がパラメータ多相のための記号になり、中に型がパラメータとして入ります。もちろんクラスだけでなくトレイトやメソッドにも適用でき、それぞれ以下のように定義・呼び出しができます。

リスト6.1: 型パラメータを持つクラス

```
1: class Box[T](var element: T) {
2:   def get(): T = element
3:   def set(newElement: T): Unit = {
4:     element = newElement
5:   }
6: }
7:
8: val intBox = new Box[Int](10) // 'Box[Int]'
9: println(intBox.get())
10: /* 出力
11: 10
12: */
13: intBox.set(0)
14: println(intBox.get())
15: /* 出力
16: 0
```

```
17: */
18:
19: sealed abstract class Animal(val cry: String)
20: case object Cat extends Animal("にゃー")
21: case object Dog extends Animal("わんわん")
22:
23: val animalBox = new Box[Animal](Cat) // 'Box[Animal]'
24: println(animalBox.get())
25: /* 出力
26: Cat
27: */
28: animalBox.set(Dog) // 'Dog' は 'Animal' を継承しているのでok
29: println(animalBox.get())
30: /* 出力
31: Dog
32: */
33:
34: // 型パラメータの省略
35: val intBox2 = new Box(10) // 'Box[Int]'
36: val catBox = new Box(Cat) // 'Box[Animal]' ではなく 'Box[Cat]'
37: // catBox.set(Dog) // Error: 'Box[Cat]' に 'Dog' は入らない
```

型パラメータは省略できますが、意図しない推論をされる可能性もあるので注意してください。以下に一般的な形も示します。

リスト6.2: 型パラメータを持つクラス・トレイト・メソッド

```
class クラス名 [型パラメータ, ...] (コンストラクタ引数: コンストラクタ引数の型, ...) {
  メンバの定義
}

trait トレイト名 [型パラメータ, ...] {
  メンバの定義
}

def メソッド名 [型パラメータ, ...] (引数: 引数の型, ...): 返り値の型 = 式
```

型パラメータにはそのまま任意の型を渡すことができますが、特定クラスの子クラス/継クラスといった制約を加えることもできます。[T]の代わりに[T <: A]とすることでTがAかAの子クラス、[T >: B]とすることでTがBかBの継クラスに制限できます。

<<目次>>

第1章 Scalaの世界へようこそ

1.1 この本について

1.2 Scalaとは?

【コラム】参考になるウェブサイト・本

第2章 Scalaの環境を作る

2.1 sbtをインストールする

2.2 sbtでテンプレートからプロジェクトを作る

2.3 sbtでプログラムを実行する

2.4 sbtでScala REPLを使う

【コラム】Scastie

【コラム】サンプルコード

第3章 Hello Worldを噛み砕こう—エントリポイント・メソッド・式と文・ブロック式

3.1 Hello World—エントリポイント

3.2 Hello Worldの構成要素 —メソッド・式と文・ブロック式

【コラム】「???」を利用する

第4章 FizzBuzzしてみよう—for式・if式といった基本的な制御構文

4.1 FizzBuzzに必要な道具を揃える① — for式

4.2 FizzBuzzに必要な道具を揃える② — if式

4.3 FizzBuzzの趣向を変えてみる① — match式

4.4 FizzBuzzの趣向を変えてみる②—再帰

第5章 オブジェクト指向で多角形を扱おう—クラスやトレイトといったオブジェクト指向に関わる構文

5.1 多角形を表現してみる—クラス

5.2 便利機能を追加してみる—オブジェクト

【コラム】ケースクラス

5.3 多角形に色と透明度を付けてみる—トレイト

【コラム】タプル

第6章 FizzBuzzを作ってみよう—パラメータ多相・コレクション・関数

6.1 コレクションを扱う前に…—パラメータ多相

6.2 FizzBuzzを作る—Range・List・関数

【コラム】主なコレクション

【コラム】主なコレクション操作メソッド

第7章 安全第一—エラーハンドリング・Option・Either

7.1 その引数は安全ですか?① —Option

7.2 その引数は安全ですか?② —Either

第8章 らくらく非同期処理 —Future

【コラム】ファンクタ・モナド・モノイド

第9章 またFizzBuzzしてみよう—IO・JSON・implicit・テスト

9.1 JSONファイルでFizzBuzz —IO・JSON

9.2 縁の下の力持ち—implicit

【コラム】implicit class

9.3 チェックチェック! —テスト

【コラム】よく利用されるその他のライブラリ

<< 著者紹介 >>

伊藤 竜一

大阪大学情報科学研究科博士前期課程在籍。高並列環境でのデータ分析の最適化・高速化の研究に従事。研究の傍ら、未踏事業にて「様々なデータソースに対応するグラフ処理エンジンの開発」を行う。プログラミング言語である Scala 好きが高じて Scala 関西勉強会や Scala 関西 Summit にて登壇。またその運営にも携わる。

<< 販売ストア >>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple iBookstore、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【株式会社インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D（本社：東京都千代田区、代表取締役社長：井芹昌信）は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>



株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:唐島夏生、証券コード:東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「モバイルサービス」を主要テーマに専門性の高いコンテンツ+サービスを提供するメディア事業を展開しています。2017年4月1日に創設25周年を迎えました。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

〒101-0051 東京都千代田区神田神保町 1-105

TEL 03-6837-4820

電子メール: np-info@impress.co.jp