

第 5 章 第一級の関数 (First-Class Functions)

Scala の関数は「第一級の値」です。他の値と同じように、関数のパラメータとして渡したり結果として返したりできます。他の関数をパラメータとしてとったり、関数を結果として返す関数は、**高階関数** と呼ばれます。この章では高階関数を紹介し、それがプログラム作成にどれほど柔軟なメカニズムを与えるか示します。

興味をそそる例として、次の 3 つの関連した課題を考えます。

1. 2 つのあたえられた数 a と b の間にある、すべての整数の和を求める関数を書きなさい。

```
def sumInts(a: Int, b: Int): Int =
  if (a > b) 0 else a + sumInts(a + 1, b)
```

2. 2 つのあたえられた数 a と b の間にある、すべての整数の 2 乗の和を求める関数を書きなさい。

```
def square(x: Int): Int = x * x
def sumSquares(a: Int, b: Int): Int =
  if (a > b) 0 else square(a) + sumSquares(a + 1, b)
```

3. 2 つのあたえられた数 a と b の間にある、すべての整数 n について 2^n の和を求める関数を書きなさい。

```
def powerOfTwo(x: Int): Int = if (x == 0) 1 else 2 * powerOfTwo(x - 1)
def sumPowersOfTwo(a: Int, b: Int): Int =
  if (a > b) 0 else powerOfTwo(a) + sumPowersOfTwo(a + 1, b)
```

これらの関数はすべて、異なる値 f に対する $\sum_a^b f(n)$ のインスタンスです。関数 `sum` を定義して、共通パターンを括り出せます。

```
def sum(f: Int => Int, a: Int, b: Int): Int =
  if (a > b) 0 else f(a) + sum(f, a + 1, b)
```

型 `Int => Int` は、型 `Int` の引数をとって、型 `Int` の結果を返す関数の型です。したがって `sum` は、他の関数をパラメータとしてとる関数です。別の言葉で言えば、`sum` は **高階関数** です。

`sum` を使って、和を求める 3 つの関数を次のように書けます。

```
def sumInts(a: Int, b: Int): Int = sum(id, a, b)
def sumSquares(a: Int, b: Int): Int = sum(square, a, b)
def sumPowersOfTwo(a: Int, b: Int): Int = sum(powerOfTwo, a, b)
```

ただし、

```
def id(x: Int): Int = x
def square(x: Int): Int = x * x
def powerOfTwo(x: Int): Int = if (x == 0) 1 else 2 * powerOfTwo(x - 1)
```

- [5.1 無名関数 \(Anonymous Functions\)](#)
- [5.2 カリー化 \(Currying\)](#)
- [5.3 例：関数の不動点探索](#)
- [5.4 まとめ](#)
- [5.5 ここまでの構文](#)

[前ページ](#) [5 章 目次](#) [次ページ](#)

- 冒頭の「他の他の値」は「他の値」の typo でしょうか。 -- ryugate (2008-05-19 22:19:55)
- 「first-class」は、「ファーストクラス」でいいんです？ 補足説明は必要ですが、無理に「第一級」と訳す必要はないと思われ。 -- murase_syuka (2008-09-25 01:12:53)

名前:	<input type="text"/>
コメント:	<input type="text"/>

投稿