

Scala Collections

How many ways are there to say "Multiple Things"

Julian Bieber

December 3, 2018

Sequence

```
val sequence = Seq(1, 2, 3, 4, 5)  
println(sequence)
```

Sequence

```
val sequence = Seq(1, 2, 3, 4, 5)  
println(sequence)
```

```
List(1, 2, 3, 4, 5)
```

Stream

```
val stream = Stream(1, 2, 3, 4, 5)
println(stream)
val seq: Seq[Int] = stream.toSeq
seq.map(println)
```

Stream

```
val stream = Stream(1, 2, 3, 4, 5)
println(stream)
val seq: Seq[Int] = stream.toSeq
seq.map(println)
```

```
Stream(1, ?)
1
```

Stream Consumption

```
val stream = Stream(1, 2, 3)
stream.foreach(println)
val streamPlusOne = stream.map(_ + 1)
println(stream.size)
streamPlusOne.foreach(println)
```

Stream Consumption

```
val stream = Stream(1, 2, 3)
stream.foreach(println)
val streamPlusOne = stream.map(_ + 1)
println(stream.size)
streamPlusOne.foreach(println)
```

1
2
3
3
2
3
4

Sequence Order of Execution

```
Seq(1, 2, 3).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}
```


Sequence Order of Execution

```
Seq(1, 2, 3).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}
```

```
(method1,1)
(method1,2)
(method1,3)
(method2,1)
(method2,2)
(method2,3)
```

Stream Order of Execution

```
Stream(1, 2, 3).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}
```

(method1,1) (method2,1)

Stream Order of Execution

```
Stream(1, 2, 3).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}
```

(method1,1) (method2,1)

Stream.force?

```
Stream(1, 2, 3).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}.force
```

Stream.force?

```
Stream(1, 2, 3).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}.force
```

```
(method1,1)
(method2,1)
(method1,2)
(method2,2)
(method1,3)
(method2,3)
```

Infinite Streams

```
val stream = Stream.from(1).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}

println(stream.sum)
```

Infinite Streams

```
val stream = Stream.from(1).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}

println(stream.sum)
```

...

(method1,2517132)

(method2,2517132)

Exception: java.lang.OutOfMemoryError

Infinite Iterators

```
val iterator = Stream.from(1).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}.tolerator

println(iterator.sum)
```


Infinite Iterators

```
val iterator = Stream.from(1).map{ i =>
  println("method1", i)
  i
}.map{ i =>
  println("method2", i)
  i
}.tolerator

println(iterator.sum)
```

```
...
(method1,2517132)
(method2,2517132)
...
```