

【Kotlin】Sequence

Sequence

在 Kotlin 中，`Sequence` 是一種延遲計算（lazy evaluation）的集合類型，用於處理大量或潛在無限數據的情況。`Sequence` 的主要特點是每次需要元素時才會進行計算，這有助於優化性能並節省內存。以下是 `Sequence` 的主要特點及用法：

特點

1. **延遲計算**：只有在需要元素時才會計算，這與立即計算（eager evaluation）不同。
2. **中間操作和終端操作**：中間操作（如 `map`、`filter`）是延遲計算的，終端操作（如 `toList`、`sum`）會觸發整個計算過程。
3. **無限序列**：可以處理無限長度的序列，只要終端操作不需要遍歷整個序列即可。

用法

創建 Sequence

可以從現有集合或使用生成函數來創建 `Sequence`。

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

從集合創建：

```
val list = listOf(1, 2, 3, 4, 5)
val sequenceFromList = list.asSequence()
```

使用生成函數創建：

```
val generatedSequence = generateSequence(1) { it + 1 }
```

中間操作

中間操作是延遲計算的，只有在終端操作觸發時才會計算。

```
val sequence = sequenceOf(1, 2, 3, 4, 5)
    .map { it * 2 }
    .filter { it > 5 }
```

上面的 `map` 和 `filter` 操作都不會立即執行。

終端操作

終端操作會觸發整個計算過程，並返回一個結果。

```
val result = sequence.toList() // [6, 8, 10]
```

常見的終端操作有 `toList()`、`sum()`、`first()`、`count()` 等。

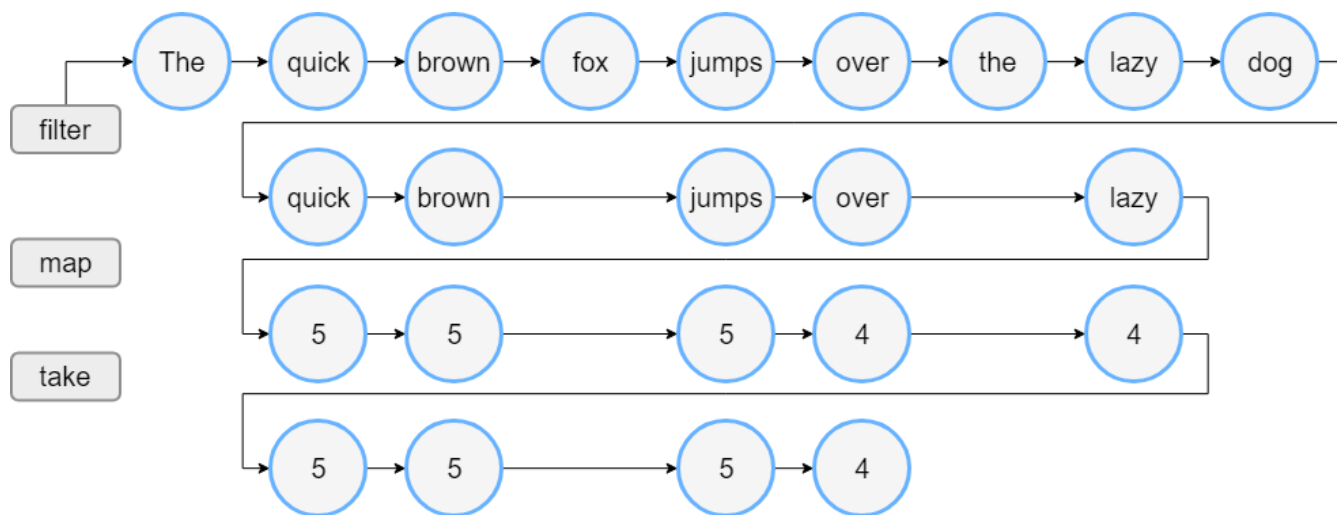
範例

```
fun main() {
    val sequence = generateSequence(1) { it + 1 }
        .map { it * 2 }
        .filter { it % 3 == 0 }

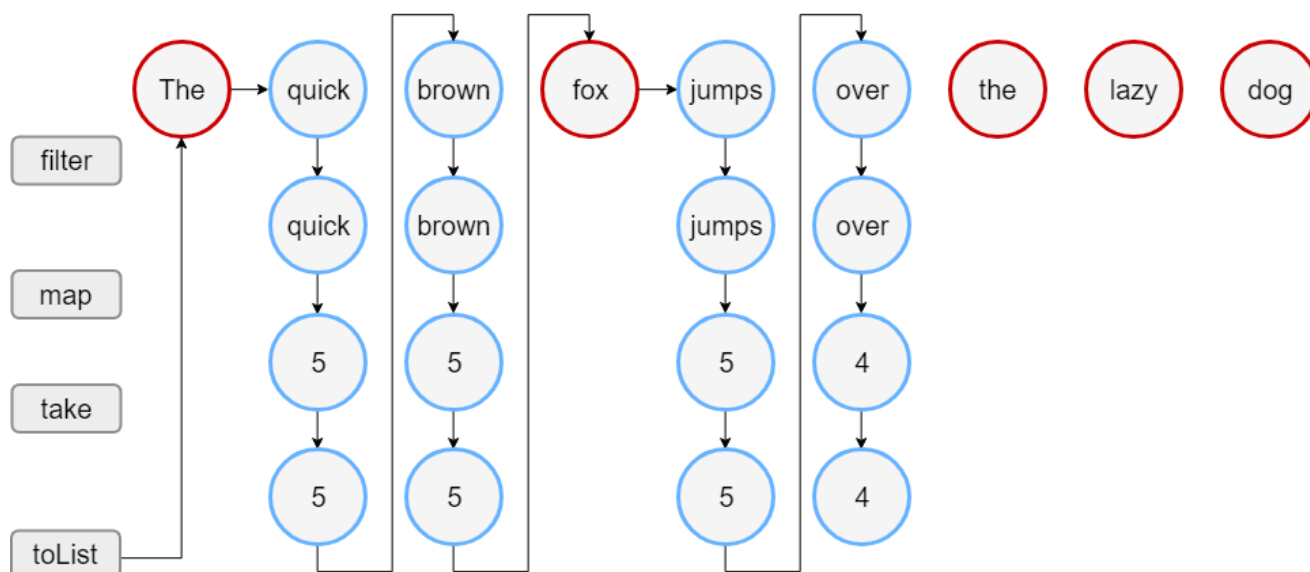
    // 取前五個符合條件的數字
    val result = sequence.take(5).toList()

    println(result) // [6, 12, 18, 24, 30]
}
```

在這個範例中，我們生成一個從 1 開始的無限序列，然後對每個數字乘以 2，再篩選出能被 3 整除的數字。最後使用 `take(5)` 取出前五個符合條件的數字並轉換成列表。



Sequence 在需要處理大量數據或無限數據流時特別有用，因為它可以有效地管理計算資源並避免不必要的計算。



🔄修訂版本 #3

★由 treeman 建立於 29 🕒 2024 17:55:59

✍由 treeman 更新於 31 🕒 2024 10:35:50