

7. Aggregation

渐进聚合具有相同 Key 的数据流元素，以 min 和 minBy 为例，min 返回的是整个 KeyedStream 的最小值，minBy 按照 Key 进行分组，返回每个分组的最小值。在 KeyedStream 上应用聚合运算输出结果为 DataStream，如代码清单 2-7 所示。

代码清单 2-7 内置聚合运算代码示例

```
keyedStream.sum(0);
keyedStream.sum("key");
keyedStream.min(0);
keyedStream.min("key");
keyedStream.max(0);
keyedStream.max("key");
keyedStream.minBy(0);
keyedStream.minBy("key");
keyedStream.maxBy(0);
keyedStream.maxBy("key");
```

8. Window

对 KeyedStream 的数据，按照 Key 进行时间窗口切分，如每 5 秒钟一个滚动窗口，每个 key 都有自己的窗口。该类运算应用在 KeyedStream 上，输出结果为 WindowedStream。

输出结果的类泛型为 WindowedStream<T, K, W extends Window>，T 为 KeyedStream 中的元素数据类型，K 为指定 Key 的数据类型，W 为窗口类型，如代码清单 2-8 所示。

代码清单 2-8 Window 代码示例

```
//时间长度为 5 秒钟的 Tumble 窗口
dataStream.keyBy(0).window(TumblingEventTimeWindows.of(Time.seconds(5)));
```

关于窗口，第 4 章会有详细讲解。

9. WindowAll

对一般的 DataStream 进行时间窗口切分，即全局 1 个窗口，如每 5 秒钟一个滚动窗口。应用在 DataStream 上，输出结果为 AllWindowedStream，如代码清单 2-9 所示。

代码清单 2-9 WindowAll 代码示例

```
//时间长度为 5 秒钟的 Tumble 窗口
dataStream.windowAll(TumblingEventTimeWindows.of(Time.seconds(5)));
```



注意：在一般的 DataStream 上进行窗口切分，往往会导致无法并行计算，所有的数据会集中到 WindowAll 算子的一个 Task 上。

关于窗口请参照 Window 原理和机制章节。

10. Window Apply

将 Window 函数应用到窗口上，Window 函数将一个窗口的数据作为整体进行处理。Window Stream 有两种：分组后的 WindowedStream 和未分组的 AllWindowedStream。

(1) WindowedStream

在 WindowedStream 上应用的是 WindowFunction，在 WindowStream 应用此类运算，输出结果为 DataStream。WindowFunction<IN, OUT, KEY, W extends Window>中的 IN 表示输入值