

Flink DataStream API 编程实践

1. 使用多种算子完成 Flink 文本处理实验 (T1)

本实验将模拟一个简单的实时文本事件处理场景，通过创建一个 `Socket` 数据源并使用不同的 Flink 算子在文本数据流中统计每个含字母 a 的单词（不区分大小写）的出现次数，并将结果打印在终端，输入样例数据如下：

```
This is a sample input text with some words
It contains several words that have the letter a
These words will be counted by the Flink program
```

请根据给定的实验数据格式，按照以下实验步骤编写 Flink 代码完成实验要求。

- (1) 创建一个 Flink 应用程序，该程序从一个 `Socket` 源接收文本数据流。
- (2) 使用 `map` 算子将输入的文本转换为小写字母，以确保大小写不敏感。
- (3) 使用 `flatMap` 算子将每行文本拆分成单词。
- (4) 使用 `filter` 算子筛选出包含特定字母的单词，例如只保留包含字母 "a" 的单词。
- (5) 使用 `keyBy` 算子按字母对单词进行分组。
- (6) 使用 `reduce` 算子计算每个字母对应的单词数量，并输出结果。

2. 使用窗口计算函数 ReduceFunction 统计传感器温度最大值 (T2)

通过 `Socket` 模拟监听实时传感器温度数据流，创建一个滚动时间窗口，使用窗口计算函数 `ReduceFunction` 计算给定窗口时间内的对应传感器的最大温度值，`Socket` 中输入的传感器数据每行格式为 (`sensorId temperature`)，输入样例数据如下：

```
s1 2
s2 3
s3 4
s1 6
```

3. 使用窗口计算函数 AggregateFunction 计算给定窗口时间内的订单销售额 (T3)

通过 `Socket` 模拟在线销售订单的实时统计场景，创建一个滚动时间窗口，使用窗口计算函数 `AggregateFunction` 计算给定窗口时间内的订单销售额，`Socket` 中输入的订单数据每行格式为 (订单 ID, 产品名称, 销售数量)，样例数据如下：

```
s10235,towel,2
s10236,toothbrush,3
s10237,toothbrush,7
s10238,instantnoodles,5
s10239,towel,3
s10240,toothbrush,3
```

s10241,instantnoodles,3

4. 使用窗口计算函数 ProcessWindowFunction 完成车辆种类数量统计 (T4)

通过模拟监听实时车辆数据流，创建一个滚动时间窗口，使用窗口计算函数 `ProcessWindowFunction` 计算给定窗口时间内的不同车辆种类的总数量。每行输入数据格式为“种类 数量”，例如“卡车 10”表示有 10 辆卡车。在窗口时间内，对每个种类的车辆数量进行累加，并输出每个种类的总数量。样例数据如下：

```
卡车 10
燃油车 20
电动车 20
摩托车 10
摩托车 30
卡车 15
燃油车 20
```

5. 基于增量聚合和全窗口函数的水传感器最大水位值分析 (T5)

假设有一个实时数据流（用 `Socket` 模拟），数据流中包含水传感器的数据，每条数据包括传感器 ID、水位值，计算每个传感器在一个固定时间窗口内的最大水位值。要求使用增量聚合函数来计算窗口内的最大水位值，并使用全窗口函数来输出结果。增量聚合函数每次收到数据时都会更新最大水位值，而全窗口函数在窗口结束时将最大水位值输出，利用全窗口函数的上下文输出窗口开始时间以及结束时间。

本实验的样例输入如下：

```
s1,20
s1,30
s1,25
s2,10
s2,15
【下一个时间窗口】
s1,40
s1,35
s2,10
s2,5
```

6. 自定义触发器实现窗口数据求和 (T6)

设计一个自定义触发器，用于在数据流中的一定时间窗口内计算窗口数据的和。数据流中的每个元素代表一个整数，通过自定义触发器，当累计到一定数量的数据时，触发窗口计算并输出窗口内数据的总和。

本实验的样例输入为自定义的数据源，在一定时间内随机生成整数，样例输入和输出如下：

```
Input Data:3> 10
Input Data:4> 1
Input Data:5> 6
Input Data:6> 3
Input Data:7> 7
Sum of window:2> 27
Input Data:8> 9
Input Data:1> 2
Input Data:2> 8
Input Data:3> 1
Input Data:4> 4
Sum of window:3> 24
```