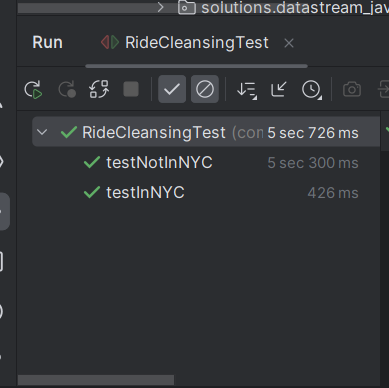
**Задание №1 (Ride Cleansing)**

Необходимо очистить поток событий, связанных с поездками на такси, удалив события, которые начинаются или заканчиваются за пределами города Нью-Йорк.

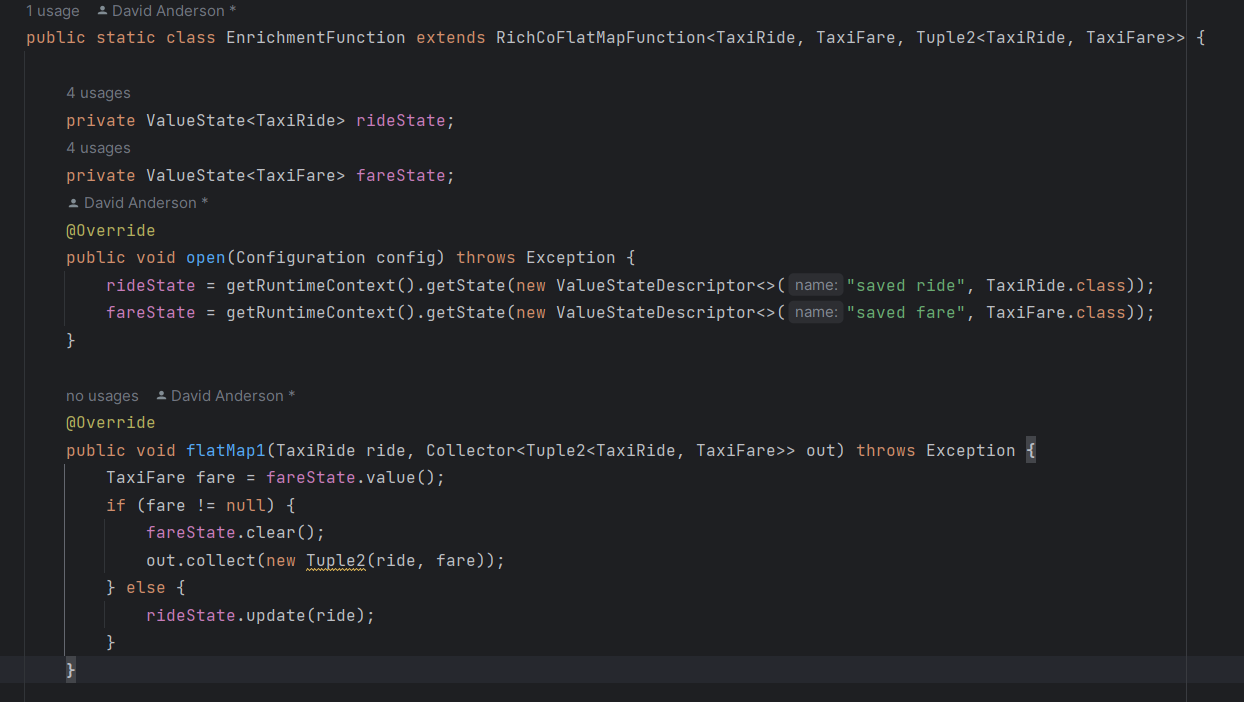


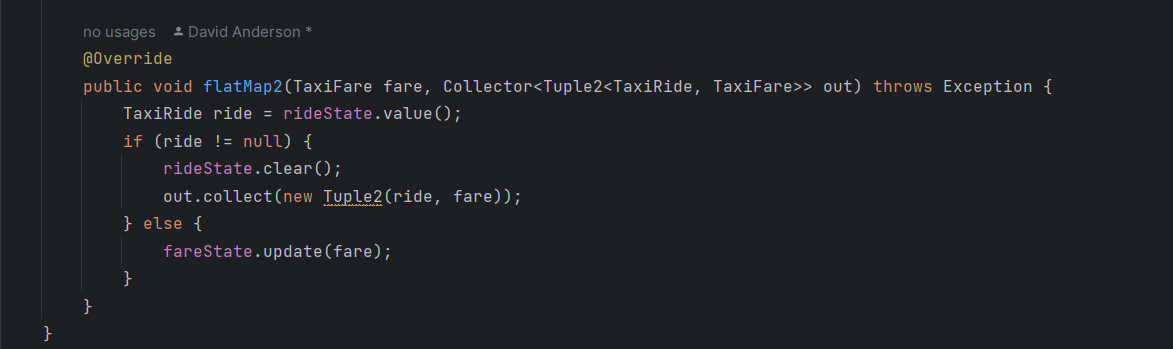
При решении использовался метод isInNYC, определяющей нахождение локации в городе Нью-Йорк.



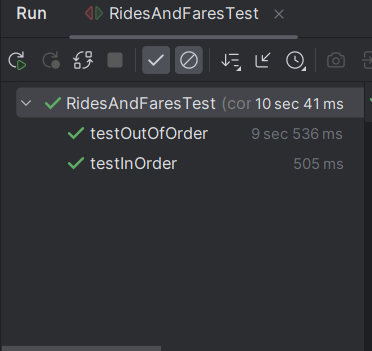
**Задание №2 (Rides and Fares)**

В данной упражнении необходимо объединить данные с TaxiRide и записи с TaxiFare



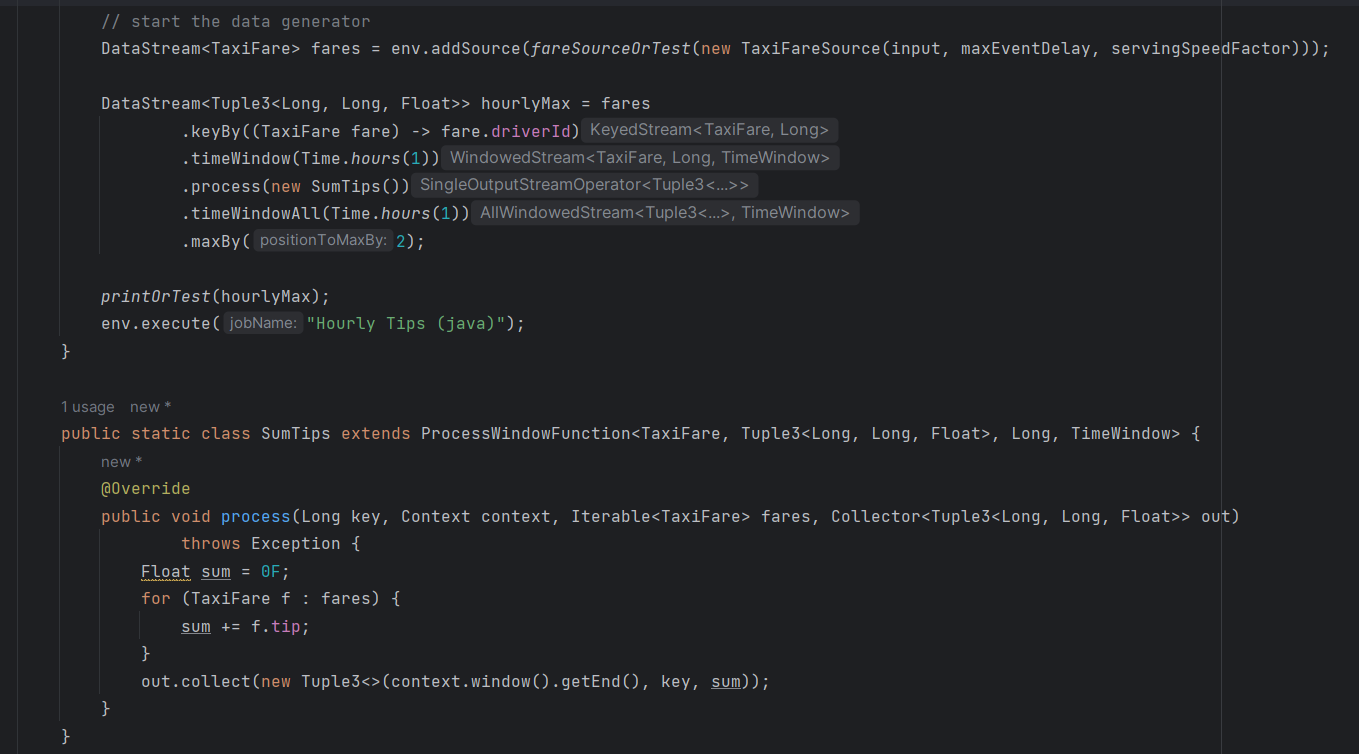


В решении в функции flatMap1 при поступлении TaxiRide идёт проверка на существование стоимости проезда, если она существует, то создаётся соответствующий кортеж. В функции flatMap2 происходит то же самое, только поступает TaxiFare и проверяется наличие самой поездки.

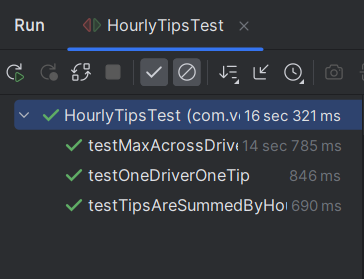


**Задание №3 (Hourly Tips)**

Необходимо определить водителя, получающего наибольшее количество чаевых за каждый час.

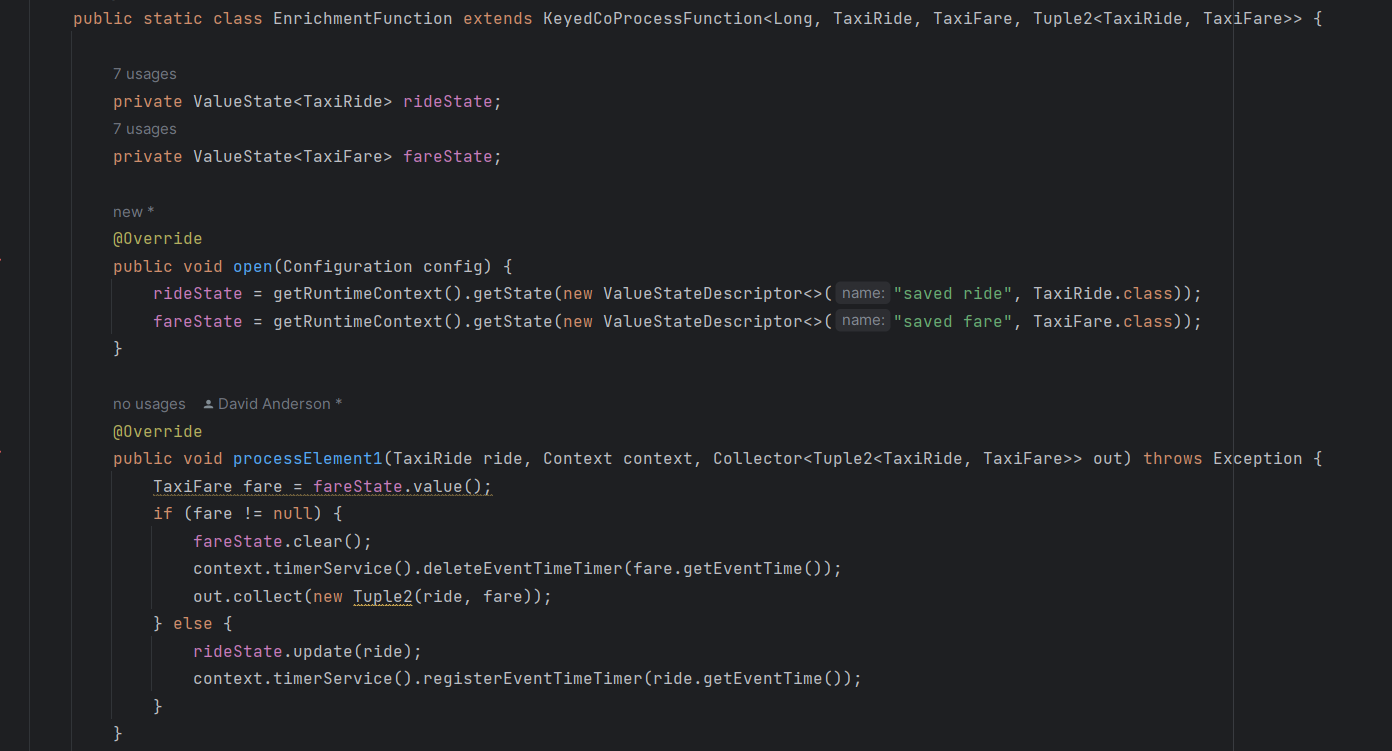


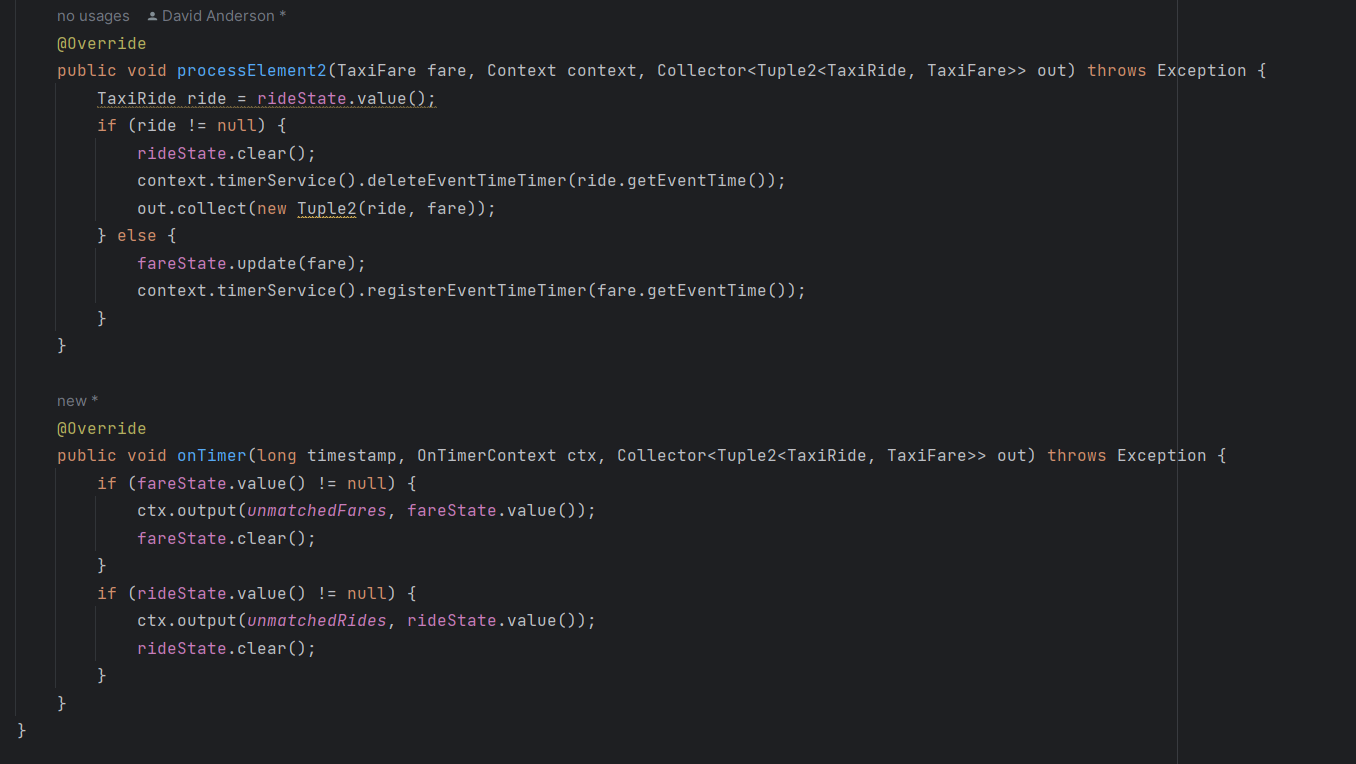
В решении класс SumTips ищет сумму чаевых для каждого водителя, а в основной функции данные группируются и ищутся водители с максимальными почасовыми чаевыми.



**Задание №4 (Expiring State)**

Необходимо дополнить поездки на такси информацией о стоимости проезда.





Решение приблизительно такое же, как и в задании 2, только для данных, которые не получилось составить, если, например, платёж пришёл поздно, то они отправляются в sideOutput.

