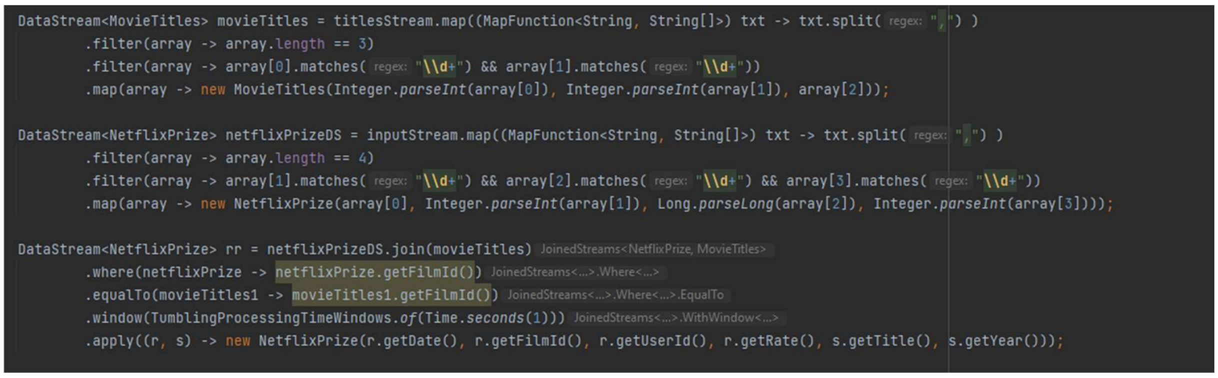
1. Skrypty – osobny plik
2. Utrzymanie czasu rzeczywistego, transformacje

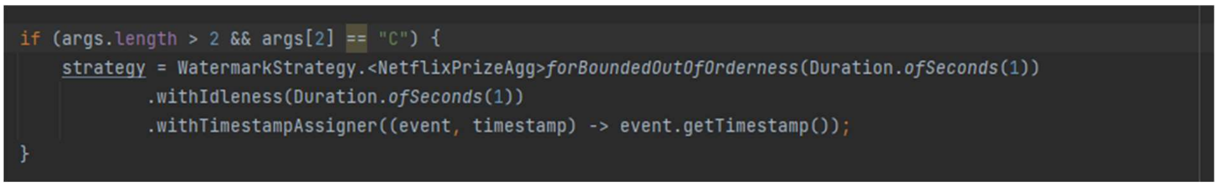


```java
DataStream<MovieTitles> movieTitles = titlesStream.map((MapFunction<String, String[]>) txt -> txt.split( regex: ",") )
        .filter(array -> array.length == 3)
        .filter(array -> array[0].matches( regex: "\\d+") && array[1].matches( regex: "\\d+"))
        .map(array -> new MovieTitles(Integer.parseInt(array[0]), Integer.parseInt(array[1]), array[2]));

DataStream<NetflixPrize> netflixPrizeDS = inputStream.map((MapFunction<String, String[]>) txt -> txt.split( regex: ",") )
        .filter(array -> array.length == 4)
        .filter(array -> array[1].matches( regex: "\\d+") && array[2].matches( regex: "\\d+") && array[3].matches( regex: "\\d+"))
        .map(array -> new NetflixPrize(array[0], Integer.parseInt(array[1]), Long.parseLong(array[2]), Integer.parseInt(array[3])));

DataStream<NetflixPrize> rr = netflixPrizeDS.join(movieTitles) JoinedStreams<NetflixPrize, MovieTitles>
        .where(netflixPrize -> netflixPrize.getFilmId()) JoinedStreams<...>.Where<...>
        .equalTo(movieTitles1 -> movieTitles1.getFilmId()) JoinedStreams<...>.Where<...>.EqualTo
        .window(TumblingProcessingTimeWindows.of(Time.seconds(1))) JoinedStreams<...>.WithWindow<...>
        .apply((r, s) -> new NetflixPrize(r.getDate(), r.getFilmId(), r.getUserId(), r.getRate(), s.getTitle(), s.getYear()));
```

3. Obsługa trybu C



```java
if (args.length > 2 && args[2] == "C") {
    strategy = WatermarkStrategy.<NetflixPrizeAgg>forBoundedOutOfOrderness(Duration.ofSeconds(1))
            .withIdleness(Duration.ofSeconds(1))
            .withTimestampAssigner((event, timestamp) -> event.getTimestamp());
}
```