

12월 5일 데이터를 수정하여

```
Trader john = new Trader("John", "New York");
```

```
Trader jane = new Trader("Jane", "London");
```

```
Trader mark = new Trader("Mark", "New York");
```

```
Trader emily = new Trader("Emily", "London");
```

```
List<Transaction> transactions = Arrays.asList(
```

```
    new Transaction(john, 2020, 1200),
```

```
    new Transaction(jane, 2019, 800),
```

```
    new Transaction(mark, 2021, 1500),
```

```
    new Transaction(emily, 2020, 700),
```

```
    new Transaction(john, 2019, 950),
```

```
    new Transaction(jane, 2020, 600)
```

```
);
```

2020년에 발생한 모든 트랜잭션을 찾아 금액 기준으로 내림차순 정렬하시오.

모든 거래자의 이름을 중복 없이 알파벳 순으로 정렬하여 출력하시오.

뉴욕(New York)에 거주하는 모든 거래자의 이름을 정렬하여 출력하시오.

트랜잭션의 최소 금액을 찾아라.

모든 거래가 발생한 도시를 중복 없이 출력하시오.

런던(London)에 거주하는 거래자가 하나라도 있는지 확인하시오.

모든 트랜잭션의 총합을 구하시오.

최댓값을 가진 트랜잭션을 출력하시오.

모든 거래자의 이름을 연결하여 하나의 문자열로 반환하시오. (예: "Emily,Jane,John,Mark")

2019년에 발생한 트랜잭션의 평균 금액을 계산하시오.

힌트

힌트

1. 스트림 생성

`transactions.stream()`

2. 필터링

`.filter(transaction -> transaction.getYear() == 2020)`

3. 정렬

`.sorted(Comparator.comparing(Transaction::getValue).reversed())`

4. 최댓값/최솟값

`.max(Comparator.comparing(Transaction::getValue))`

5. 맵핑

```
.map(transaction -> transaction.getTrader().getName())
```

6. 수집

```
.collect(Collectors.toList())
```

7. 집계

```
.mapToInt(Transaction::getValue).sum()
```

8. 문자열 연결

```
.collect(Collectors.joining(", "))
```