

Aula 19 - Streams API e Desafio

Docupedia Export

Author: Sílio Leonardo (SO/OPM-TS21-BR)

Date: 16-Oct-2024 16:25

Table of Contents

Streams API é uma funcionalidade do Java que permite fazer processamento de listas com facilidade e de forma prática. Observe como criar uma lista a partir de uma Stream e o que pode ser feito a partir da mesma:

```
1  import java.io.IOException;
2  import java.nio.file.Paths;
3  import java.nio.file.Files;
4  import java.nio.charset.Charset;
5  import java.util.List;
6  import java.util.ArrayList;
7  import java.util.stream.Stream;
8  import java.util.stream.IntStream;
9  import java.util.stream.Collectors;
10
11  public class Main {
12      public static void main(String[] args) throws IOException {
13
14          ArrayList<Integer> list = new ArrayList<>();
15          for (int i = 1; i < 6; i++)
16              list.add(i);
17
18          List<Integer> result = list.stream()
19              .map(n -> n * n)
20              .filter(n -> n % 2 == 1)
21              .sorted((n, m) -> m - n)
22              .limit(2)
23              .collect(Collectors.toList());
24
25          for (Integer value : result)
26              System.out.println(value); // 25 9
27
28
29          Integer sum = Stream.of(1, 2, 3, 4, 5)
30              .map(n -> n * n)
31              .filter(n -> n % 2 == 1)
32              .sorted()
33              .skip(1)
34              .reduce(0, (acc, crr) -> acc + crr);
35
36          System.out.println(sum); // 34
```

```
37
38
39     boolean allPrimes = Stream.of(2, 3, 5, 7, 11)
40         .allMatch(n ->
41             {
42                 Integer a = n / 2;
43                 Integer r = ((int) Math.pow(a, n - 1) % n);
44                 return r == 1;
45             });
46
47     System.out.println(allPrimes); // true
48
49
50     Stream.of(3, 5, 7, 8, 11)
51         .filter(n -> n % 2 == 0)
52         .findAny()
53         .ifPresent(n -> System.out.println(n)); // 8
54
55
56     long odds = IntStream.range(1, 300)
57         .filter(n -> n % 2 == 1)
58         .count();
59
60     System.out.println(odds); // 150
61
62
63     // Lendo de Arquivos
64     long lines = Files.lines(Paths.get("Main.java"), Charset.defaultCharset())
65         .filter(line -> line.contains("filter"))
66         .count();
67
68     System.out.println(lines); // 5
69 }
70 }
```

A partir destes recursos, você deverá resolver o seguinte desafio:

Busque pelo SRAG 2021 (Síndrome Respiratória Aguda Grave). Esses dados trazem todos os casos de entrada no hospital com uma doença respiratória. Olhe o dicionário de dados para identificar cada coluna e poder ler os dados. Filtre os casos de Covid-19 e responda a seguinte pergunta: Onde a mortalidade é maior, nos vacinados ou nos não vacinados? Use Streams API para responder essa pergunta. Dicas:

1. Não acesse os dados de um servidor remoto, caso contrário a aplicação poderá ficar muito lenta. Tenha o .csv dos dados no Desktop.
2. Ao terminar tudo busque pelo fenômeno de Simpson.

Você pode encontrar os dados em: <https://opendatasus.saude.gov.br/dataset/srag-2021-a-2024>