1. Gasiti toate numerele pare (TODO #1)
2. Gasiti elementul din steam care are numele “Jack” (TODO #2)
3. Gasiti toate elementele din stream care sunt not null (TODO #3)
4. Gasiti elementul din steam care are numele “James” si varsta “20” (TODO #4)
5. Dublati numerele din stream. (TODO #5)
6. Returnati doar numele persoanelor.(TODO #6)
7. Concatenati cele 2 liste din stream intr-una singura (TODO #7)
8. Calculati produsul numerelor din lista (TODO #8)
9. Gasiti prima persoana de sex masculin din lista (TODO #9)
10. Verificati daca toate persoanele din lista au varsta sub 70 de ani. (TODO #10)
11. Gasiti cel mai mic numar din stream-ul dat (TODO #11)
12. Gasiti cea mai tanara persoana din lista (TODO #12)
13. Calculati media de varsta a persoanelor din lista (TODO #13)
14. Gasiti toate nationalitatile pentru persoanele a caror nume incep cu ‘J’. (TODO #14)
15. Gasiti primul patrat perfect care se divide cu 3 din lista data. (TODO #15)
16. Calculati suma total a lugimilor cuvintelor din lista data.
17. Calculati salariul total al persoanelor din lista

Collectors

1. Partitionati lista dupa criteriul - calories > 380 (TODO #C1)
2. Grupati lista de dishes dupa ClaoricLevel: (TODO #C2)

* public enum CaloricLevel { DIET, NORMAL, FAT }
* 0 < DIET <= 400 calories
* 400 < NORMAL <= 700
* FAT > 700

1. Eliminiati duplicatele din lista de numere (TODO #C3)
   1. Folosind operatii streams
   2. Folosind collectors
2. Transformati Stream-ul intr-un TreeSet (TODO #C4)
3. Gasiti cel mai putin caloric dish de tipul MEAT (TODO #C5)
4. Returnati statistica (average, sum, min, max) pentru caloriile dish-urilor vegetariene (TODO #C6)
5. Returnati numarul de dishes care au numarul de calorii mai mare ca 140, grupate dupa tipul de dish.(TODO #C7)
6. Rescrieti functia de la exercitiul 12 folosind maxBy/minBy.

9\*) Generati un Map de aparatie a caracterelor intr-o propozitie: (TODO #C9)

E.g. “cool” -> {“c”:1, “o”:2, “l”: 1}

Puteti extinde functia pentru a prelucra liste de propozitii?