1. First exmple

import java.util.\*;

import java.util.stream.\*;

import java.util.function.\*;

import java.util.Comparator;

class Person {

    private String name;

    private int age;

    public Person(String name, int age) {

        this.name = name;

        this.age = age;

    }

    public String getName() { return name; }

    public int getAge() { return age; }

    public String toString() {

        return name + " (" + age + ")";

    }

}

public class AdvancedStreamExample {

    public static void main(String[] args) {

        List<Person> people = Arrays.asList(

            new Person("ABC", 30),

            new Person("PQR", 20),

            new Person("XYZ", 25),

            new Person("LMN", 18),

            new Person("DMQ", 30)

        );

1. MAP: Get a list of names
2. FILTER: People over 21
3. SORTED: Sort by age
4. GROUPING BY AGE
5. Sum of all ages
6. FIND FIRST person over 25

2)

import java.util.\*;

import java.util.stream.\*;

import java.util.function.\*;

class Product {

    private String name;

    private double price;

    private String category;

    public Product(String name, double price, String category) {

        this.name = name;

        this.price = price;

        this.category = category;

    }

    public String getName() { return name; }

    public double getPrice() { return price; }

    public String getCategory() { return category; }

    @Override

    public String toString() {

        return name + " - " + price + " (" + category + ")";

    }

}

public class ProductStreamExample {

    public static void main(String[] args) {

        List<Product> products = Arrays.asList(

            new Product("Laptop", 1200.0, "Electronics"),

            new Product("Smartphone", 800.0, "Electronics"),

            new Product("Coffee", 5.0, "Groceries"),

            new Product("Notebook", 3.0, "Stationery"),

            new Product("Pen", 1.5, "Stationery"),

            new Product("Laptop", 1200.0, "Electronics")  // Duplicate product

        );

1. DISTINCT: Remove duplicates based on name (by using Set)
2. GROUPING BY category
3. Most expensive product
4. Cheapest product
5. If any product is having price is less than 100
6. if all Electronics product having price is greater than 100