Program 1:

java

import java.util.\*;

import java.util.stream.Collectors;

class Student {

public String name;

public int graduationYear;

public int score;

public Student(String name, int graduationYear, int score) {

this.name = name;

this.graduationYear = graduationYear;

this.score = score;

}

public String getName() {

return name;

}

public int getGraduationYear() {

return graduationYear;

}

public int getScore() {

return score;

}

@Override

public String toString() {

return "Student (name='" + name + "', graduationYear=" + graduationYear + ", score=" + score + ")";

}

}

class Implementation {

public static List<Student> studentsSorted(Collection<Student> students) {

return students.stream()

.sorted(Comparator.comparing(Student::getScore).reversed()

.thenComparing(Student::getName))

.collect(Collectors.toList());

}

public static Map<Integer, List<Student>> studentsByYear(Collection<Student> students) {

return students.stream()

.collect(Collectors.groupingBy(Student::getGraduationYear));

}

public static Student findOneBest(Collection<Student> students) {

return students.stream()

.max(Comparator.comparing(Student::getScore))

.orElse(null);

}

}

public class Main {

public static void main(String[] args) {

Collection<Student> students = Arrays.asList(

new Student("2014-11", 2014, 17),

new Student("2014-18", 2014, 20),

new Student("2011-18", 2013, 20)

);

List<Student> sortedStudents = Implementation.studentsSorted(students);

Map<Integer, List<Student>> studentsByYear = Implementation.studentsByYear(students);

Student bestStudent = Implementation.findOneBest(students);

System.out.println(sortedStudents);

System.out.println(studentsByYear);

System.out.println(bestStudent);

}

}

Program 2:

java

import java.util.\*;

import java.util.stream.Collectors;

class Car {

private String name;

private String carName;

private double price;

public Car(String name, String carName, double price) {

this.name = name;

this.carName = carName;

this.price = price;

}

public String getName() {

return name;

}

public String getCarName() {

return carName;

}

public double getPrice() {

return price;

}

public void setName(String name) {

this.name = name;

}

public void setCarName(String carName) {

this.carName = carName;

}

public void setPrice(double price) {

this.price = price;

}

}

class CarImplementation {

public static double sumOfPrices(List<Car> carList) {

return carList.stream()

.mapToDouble(Car::getPrice)

.sum();

}

public static List<String> printName(List<Car> carList) {

return carList.stream()

.filter(car -> car.getPrice() > 25000)

.map(Car::getCarName)

.collect(Collectors.toList());

}

public static double maxPrice(List<Car> carList) {

return carList.stream()

.mapToDouble(Car::getPrice)

.max()

.orElse(0.0);

}

}

public class Main {

public static void main(String[] args) {

List<Car> carList = new ArrayList<>();

carList.add(new Car("Car1", "Alfa Romeo", 35000.0));

carList.add(new Car("Car2", "Bugatti", 1000000.0));

carList.add(new Car("Car3", "Chrysler", 28000.0));

carList.add(new Car("Car4", "Dodge", 42000.0));

carList.add(new Car("Car5", "Essen", 910000.0));

double totalPrice = CarImplementation.sumOfPrices(carList);

List<String> carNameList = CarImplementation.printName(carList);

double maxCarPrice = CarImplementation.maxPrice(carList);

System.out.println(totalPrice);

System.out.println(carNameList);

System.out.println(maxCarPrice);

}

}