1)

**package** streams;

**import** java.util.ArrayList;

**import** java.util.Comparator;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.stream.Collectors;

**public** **class** StreamFruit1 {

**public** **static** **void** main(String[] args) {

List<Fruit> fruit=**new** ArrayList<Fruit>();

fruit.add(**new** Fruit("Apple",200,500,"Red"));

fruit.add(**new** Fruit("Banana",150,200,"Yellow"));

fruit.add(**new** Fruit("Watermelon",50,100,"Green"));

fruit.add(**new** Fruit("Strawberry",20,50,"Pink"));

fruit.add(**new** Fruit("Pomogranate",100,150,"Red"));

//Display fruits of low calories in descending order

List<Fruit> fruitcalories = fruit.stream()

.filter(f -> f.calories < 100)

.sorted(Comparator.*comparing*(Fruit::getCalories).reversed())

.toList();

System.***out***.println("low calorie fruits:\n" + fruitcalories);

//color wise list

Map<String, List<String>> fruitcolor = fruit.stream()

.collect(Collectors.*groupingBy*(Fruit::getColor,

Collectors.*mapping*(Fruit::getName, Collectors.*toList*())));

System.***out***.println("color wise list:\n" + fruitcolor);

//sort red fruits in ascending order

List<Fruit> redfruit = fruit.stream()

.filter(f -> f.getColor().equals("Red"))

.sorted(Comparator.*comparing*(Fruit::getPrice))

.toList();

System.***out***.println("red fruits:\n" + redfruit);

}

}

**package** streams;

**public** **class** Fruit {

String name;

**int** calories;

**int** price;

String color;

**public** Fruit(String name, **int** calories, **int** price, String color) {

**super**();

**this**.name = name;

**this**.calories = calories;

**this**.price = price;

**this**.color = color;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getCalories() {

**return** calories;

}

**public** **void** setCalories(**int** calories) {

**this**.calories = calories;

}

**public** **int** getPrice() {

**return** price;

}

**public** **void** setPrice(**int** price) {

**this**.price = price;

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

@Override

**public** String toString() {

**return** "Fruit [name=" + name + ", calories=" + calories + ", price=" + price + ", color=" + color + "]";

}

}

2)

**package** streams;

**public** **class** News {

**int** newsId;

String postedByUser;

String commentByUser;

String comment;

**public** News(**int** newsId, String postedByUser, String commentByUser, String comment) {

**super**();

**this**.newsId = newsId;

**this**.postedByUser = postedByUser;

**this**.commentByUser = commentByUser;

**this**.comment = comment;

}

@Override

**public** String toString() {

**return** "News [newsId=" + newsId + ", postedByUser=" + postedByUser + ", commentByUser=" + commentByUser

+ ", comment=" + comment + "]";

}

**public** **int** getNewsId() {

**return** newsId;

}

**public** String getPostedByUser() {

**return** postedByUser;

}

**public** String getCommentByUser() {

**return** commentByUser;

}

**public** String getComment() {

**return** comment;

}

**public** **void** setNewsId(**int** newsId) {

**this**.newsId = newsId;

}

**public** **void** setPostedByUser(String postedByUser) {

**this**.postedByUser = postedByUser;

}

**public** **void** setCommentByUser(String commentByUser) {

**this**.commentByUser = commentByUser;

}

**public** **void** setComment(String comment) {

**this**.comment = comment;

}

}

**package** streams;

**import** java.util.ArrayList;

**import** java.util.Collection;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Map.Entry;

**import** java.util.stream.Collectors;

**public** **class** StreamNews1 {

**public** **static** **void** main(String[] args) {

List<News>newList=**new** ArrayList<News>();

newList.add(**new** News(100,"Tom","Rocky","First comment"));

newList.add(**new** News(101,"Ram","Ravi","Second comment"));

newList.add(**new** News(102,"Tonny","","Third comment"));

newList.add(**new** News(103,"Missa","liva","Fourth comment"));

newList.add(**new** News(104, "Rohit", "Priya", "Surprising"));

Map<Integer,List<News>> newsGroup = newList.stream()

.collect(Collectors.*groupingBy*(News::getNewsId));

// System.out.println(newsGroup);

**int** maxComments = 0;

**int** newsId = 0;

**for**(Entry<Integer, List<News>> e: newsGroup.entrySet()) {

**if**(e.getValue().size() > maxComments) {

newsId = e.getKey();

maxComments = e.getValue().size();

}

}

System.***out***.println(newsId + " has max comments of " + maxComments);

}

}

3)

**package** streams;

**public** **class** Trader {

String name;

String city;

**public** Trader(String name, String city) {

**super**();

**this**.name = name;

**this**.city = city;

}

@Override

**public** String toString() {

**return** "Trader [name=" + name + ", city=" + city + "]";

}

}

4)

**package** streams;

**public** **class** Transaction {

Trader trader;

**int** year;

**int** value;

**public** Transaction(Trader trader, **int** year, **int** value) {

**super**();

**this**.trader = trader;

**this**.year = year;

**this**.value = value;

}

@Override

**public** String toString() {

**return** "Transaction [trader=" + trader + ", year=" + year + ", value=" + value + "]";

}

}