Best Salesperson

By Kenneth Rodriguez

A company sells 10 different products, and has a staff of 8 people. To incentivize the employees, the company plans to give its best 3 salespersons a raise of $1000. To qualify for the raise, the employees must have individual sales of at least $700. If more than 3 employees qualify for the raise, the top 3 with the highest individual sales will earn the reward.

Classes

This program uses 3 classes: Salesperson, Product, and Sale.

Salesperson

The Salesperson class has 3 attributes: the name, the employee ID, and the salary, and a static variable totalEmployees to keep track of the number of salespersons working for the company. It also has a single method raiseSalary(), which takes an argument (a double value) and increases the current salary by said value.

```
public class Salesperson {
	static int totalEmployees;

	public String name;
	public int employeeId;
	public double salary;

	public Salesperson(String name, double salary) {
		Salesperson.totalEmployees += 1;

		employeeId = Salesperson.totalEmployees;
		this.name = name;
		this.salary = salary;
	}

	public void raiseSalary(double raise) {
		this.salary += raise;
	}
}
```

Product

The Product class only has 2 attributes: the product ID, and the price of the product. Similar to Saleman, Product has a static variable totalProducts to keep track of the amount of products and avoid repeating the product ID.

```
public class Product {
	static int totalProducts;

	public int productId;
	public double price;

	public Product(double price) {
		Product.totalProducts += 1;

		this.productId = Product.totalProducts;
		this.price = price;
	}
}
```

Sale

The Sale class has 3 attributes: the product that was sold, the salesperson that sold it, and the amount of units that were sold.

```
public class Sale {
	Product product;
	Salesperson salesperson;
	int quantity;

	public Sale(Product product, Salesperson salesperson, int quantity) {
		this.product = product;
		this.salesperson = salesperson;
		this.quantity = quantity;
	}
}
```

Solution

We use a list, such as **List<Sale> salesData**, to keep track of all the individual sales. By creating a Stream from this list, we can achieve our desired goal of finding the best salespeople.

After creating the stream, we must first filter the individual sales that reach the goal of at least $700. Then, we sort the sales in decreasing order of value (product price times units sold). The sorted sales are then mapped to their respective salespeople, of which only the distinct salespersons are used. With this we have obtained the salespeople that fit the criteria.

However, we must also limit the amount of salespeople, in case more than 3 have met the criteria. After that, we can finally collect the remaining employees to a list.

```
List<Salesperson> bestSalespersons = salesData.stream()
		.filter( (s) -> s.product.price * s.quantity >= 700.0)
		.sorted( (sale1, sale2) -> (int) (sale2.product.price * sale2.quantity - 
sale1.product.price * sale1.quantity))
		.map( (s) -> s.salesperson)
		.distinct()
		.limit(3)
		.collect(Collectors.toList());
```

Finally, we can create another stream from the newly created list to raise the selected employees' salaries.

```
bestSalespersons.stream()
		.peek( (s) -> s.raiseSalary(1000.0))
		.forEach(
				(s) -> {
					System.out.println(s + " deserves a raise.");
					System.out.println("New salary: $" + s.salary);
				}
		);
```