There is an Employee.java and ProcessingEmployees1.java as an attachment

There are 10 tasks need to be complete by writing Functional style of code in ProcessingEmployees1.java file. Type your code in the Eclipse IDE.

The 10 tasks are printed below.

1. Display all Employees using forEach

2. Display Employees with salaries in the range $4000-$6000, sorted order by salary.

3. Comparator for comparing Employees by first name then last name

4. Display unique employee last names in sorted order

5. group Employees by department to Map with Department name as a key and List of employees as values and print the result

6.Count number of Employees in each department to map with Department as key and department count as value and print the result

7. Calculate sum of Employee salaries with Stream reduce method.

8. Find an employee who has the first name "Andy" in the list, Check null using Optional isPresent() or orElse to avoid NullPointerException.

9. Create an object using ClassName::new for the Employee class by defining your own Interface.

10. Create a lambda library for the given query and test it.
    Hint : Lambda Library, identify the parameters that are combined together in your pipeline, and consider those to be arguments for some kind of Java functional interface [3 Points]

```
List<String> el = list.stream()
.filter((x) -> x.getLastName().startsWith("I"))
.peek(System.out::println)
.map(Employee::getLastName)
.sorted()
```

```
            .collect(Collectors.toList());
            System.out.println(el);
```
