**Employee** {

int id;

String name;

int age;

String gender;

String department;

int yearOfJoining;

double salary;

}

Suppose we want objects related to specific values of a column from object list i.e. grouping elements based on some column, use below in **collect** method

***Collectors.groupingBy(ClassName::propertyName)*** in collect method

For ex. No of employees in each dept

**Collectors.*groupingBy*(Employee::getDepartment,Collectors.*counting*())**

if we want the some count or average of one column with respect to other col values.

For ex. want age count with respect to each Department.

we need two arguments to Collectors.groupingBy().

1. method reference to property by which we are grouping . in this case Dept

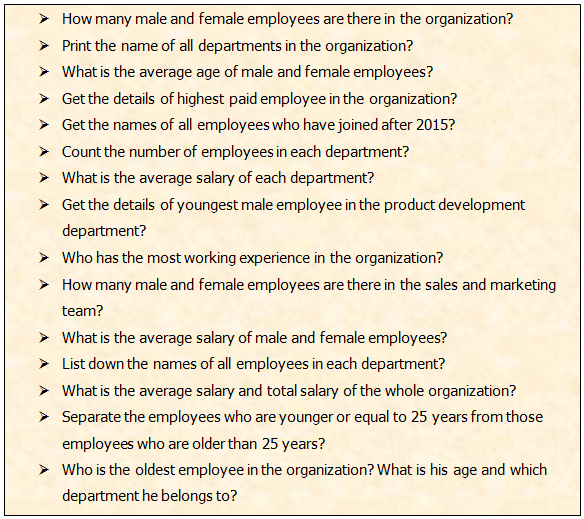
2. Collectors.counting()/Collectors.averageInt(method reference to property whose count/average we have to calculate)

For ex. What is the average salary of each department

Here we have to group by department and calculate the average

Collectors.groupingBy(Employee::getDepartment, **Collectors.averagingDouble(Employee::getSalary)**));

Example list:



Link for the reference :

<https://javaconceptoftheday.com/solving-real-time-queries-using-java-8-features-employee-management-system/>