PRINTING EMPLOYEE LIST AS COLUMNAR TABLE

In this coding exercise you will be expected to develop a method that prints an array of Employees in a columnar list. Your method ***void printEmployees(Employee[] employees)*** will print the each Employee in the array in a table by following some formatting rules. The formatting rules, complete example main method, complete Employee class and an example of expected output will be provided.

The formatting rules are:

- *The table will have column names Id, Name, Birthdate, Salary and will be left justified with trailing spaces*
  - *Id column will be 5 chars in length*
  - *Name column will be 25 chars in length*
  - *Birthdate column be 15 chars in length*
  - *Salary will be 6 chars in length*
- *Id column will be right-justified with leading zeros through the column length*
- *Name column consists of name and surname including a space in between name and surname. Name column will be left-justified with trailing spaces through the column length. If the length of name and surname is longer than 25 chars, name will be shortened to its first letter and dot.*
- *Birthdate will have a format like 25/10/1990, Sun and left justified with trailing spaces through column length*
- *Salary fraction will be rounded to 2 digits by ceiling. Salary will be right-justified with leading zeros. Salary will be printed according to the UK localization. There will be a GBP symbol at the end.*
- *There will be a line of - (dashes) having the total length of columns in between column names and data lines*
- *There will be one space between each column value*

You can develop any helper methods or classes if you need.

The complete Employee class will be provided.

1. The complete Employee class is as below

```
package com.company;

import java.math.BigDecimal;
import java.time.LocalDate;

public class Employee {
    private static int idCounter = 10;
    private int id;
    private String name;
    private String surname;
    private LocalDate birthDate;
    private BigDecimal salary;

    public Employee(String name, String surname, LocalDate birthDate,
                    BigDecimal salary) {
        setId();
        setName(name);
        setSurname(surname);
        setBirthDate(birthDate);
```

```
        setSalary(salary);
    }

    private void setId() {
        this.id = ++idCounter;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSurname() {
        return surname;
    }

    public void setSurname(String surname) {
        this.surname = surname;
    }

    public LocalDate getBirthDate() {
        return birthDate;
    }

    public void setBirthDate(LocalDate birthDate) {
        this.birthDate = birthDate;
    }

    public BigDecimal getSalary() {
        return salary;
    }

    public void setSalary(BigDecimal salary) {
        if (salary.compareTo(BigDecimal.valueOf(1000)) > 0
                || salary.compareTo(BigDecimal.valueOf(0)) < 0)
            throw new IllegalArgumentException("Wrong salary: Salary can be " +
                    "in between 0 inclusive and 1000 inclusive");
        this.salary = salary;
    }

    public int getId() {
        return id;
    }

    @Override
    public String toString() {
        return "Employee{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", surname='" + surname + '\'' +
                ", birthDate=" + birthDate +
                ", salary=" + salary +
                '}';
    }
}
```

2. The main() method and the incomplete printEmployees() method are as below

```
public static void main(String[] args) {
   Employee[] employees = new Employee[3];
   Employee employeeA = new Employee("John", "Cohen", LocalDate.of(1965
            , 10,
            15), new BigDecimal("639.56"));

   Employee employeeB = new Employee("Michealhhhhhhhhhhhhhh", "Funny",
         LocalDate.of(1990
         , 5,
         29), new BigDecimal("39.123698"));

   Employee employeeC = new Employee("Leonard", "Adam",
         LocalDate.of(2000
         , 8,
         23), new BigDecimal("789.999999"));
   employees[0]=employeeA;
   employees[1]=employeeB;
   employees[2]=employeeC;

   printEmployees(employees);
}

public static void printEmployees(Employee[] employees) {


}
```

3. Output of the above main() method is as below

```
Id    Name                        Birthdate       Salary
-------------------------------------------------------
00011 John Cohen                  15/10/1965, Fri 639.56£
00012 M.Funny                     29/05/1990, Tue 039.13£
00013 Leonard Adam                23/08/2000, Wed 790.00£
```