

## SALARY CALCULATOR

Se quiere construir un programa que calcule los costes de personal dentro de una empresa. El coste de cada trabajador viene determinado por la siguiente información: identificador, nombre, nivel, horas trabajadas y coste por hora. Se pretende que el programa calcule los costes de los trabajadores que se han registrado.

Se dispone de la siguiente implementación:

```
public class DeveloperReport {  
  
    private int id;  
    private String nombre;  
    private String nivel;  
    private int horastrabajadas;  
    private double costehora;  
  
    public DeveloperReport(int id, String nombre, String  
nivel, int horastrabajadas, double coste) {  
        this.id = id;  
        this.nombre = nombre;  
        this.nivel = nivel;  
        this.horastrabajadas = horastrabajadas;  
        this.costehora = coste;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getNivel() {  
        return nivel;  
    }  
  
    public void setNivel(String nivel) {  
        this.nivel = nivel;  
    }  
}
```

```

    public int getHorastrabajadas() {
        return horastrabajadas;
    }

    public void setHorastrabajadas(int horas) {
        this.horastrabajadas = horas;
    }

    public double getCostehora() {
        return costehora;
    }

    public void setCostehora(double coste) {
        this.costehora = coste;
    }
}

public class SalaryCalculator {
    private ArrayList<DeveloperReport> reports;

    public SalaryCalculator() {
        this.reports = new ArrayList<>();
    }

    public ArrayList<DeveloperReport> getReports() {
        return reports;
    }

    public void setReports(ArrayList<DeveloperReport> reports)
{
        this.reports = reports;
    }

    public void addReport(DeveloperReport report) {
        this.reports.add(report);
    }

    public double getTotalSalaries() {
        double total = 0;
        for (DeveloperReport report : reports) {
            total += report.getCostehora() *
report.getHorastrabajadas();
        }
        return total;
    }
}

public static void main(String[] args) {
    SalaryCalculator application = new
SalaryCalculator();
    DeveloperReport report1 = new DeveloperReport(1,
"Dev1", "Senior", 160, 30.5);

```

```

        DeveloperReport report2 = new DeveloperReport(1,
"Dev2", "Junior", 140, 18);
        DeveloperReport report3 = new DeveloperReport(1,
"Dev3", "Senior", 180, 30.5);
        application.addReport(report1);
        application.addReport(report2);
        application.addReport(report3);
        System.out.println("El coste de los salarios es: " +
application.getTotalSalaries());
    }

```

#### PREGUNTA 1

Se nos pide modificar el programa para que los trabajadores "Senior" cobren un 20% más como complemento de productividad. Como resultado, un desarrollador propone re-definir la función getTotalSalaries de la siguiente manera:

```

public double getTotalSalaries() {
    double total = 0;
    for (DeveloperReport report : reports) {
        if (report.getNivel().equals("Senior"))
            total += report.getCostehora() *
report.getHorastrabajadas() * 1.2;
        else
            total += report.getCostehora() *
report.getHorastrabajadas();
    }
    return total;
}

```

¿Es correcta? ¿Este código cumple el principio abierto/cerrado?

#### PROBLEMA 2

Modifique el código con los cambios pertinentes para que esté lo más desacoplado posible. ¿Qué principios SOLID se están aplicando?