

## Ejercicio 2.1

Para cada una de las siguientes situaciones, realice en forma iterativa los siguientes pasos:

- (i) indique el mal olor,
- (ii) indique el refactoring que lo corrige,
- (iii) aplique el refactoring, mostrando el resultado final (código y/o diseño según corresponda).

Si vuelve a encontrar un mal olor, retorne al paso (i).

### 2.1 Empleados

```
public class EmpleadoTemporario {  
    public String nombre;  
    public String apellido;  
    public double sueldoBasico = 0;  
    public double horasTrabajadas = 0;  
    public int cantidadHijos = 0;  
    // .....  
  
    public double sueldo() {  
        return this.sueldoBasico  
            + (this.horasTrabajadas * 500)  
            + (this.cantidadHijos * 1000)  
            - (this.sueldoBasico * 0.13);  
    }  
}
```

```
public class EmpleadoPlanta {  
    public String nombre;  
    public String apellido;  
    public double sueldoBasico = 0;  
    public int cantidadHijos = 0;  
    // .....  
}
```

```

public double sueldo() {
    return this.sueldoBasico
        + (this.cantidadHijos * 2000)
        - (this.sueldoBasico * 0.13);
}
}

```

```

public class EmpleadoPasante {
    public String nombre;
    public String apellido;
    public double sueldoBasico = 0;
    // .....

    public double sueldo() {
        return this.sueldoBasico - (this.sueldoBasico * 0.13);
    }
}

```

## Code smells

### 1. Duplicated code

```

public class EmpleadoTemporario {
    public String nombre;
    public String apellido;
    public double sueldoBasico = 0;
    public double horasTrabajadas = 0;
    public int cantidadHijos = 0;
}

```

```

public class EmpleadoPlanta {
    public String nombre;
    public String apellido;
    public double sueldoBasico = 0;
    public int cantidadHijos = 0;
}

```

```
public class EmpleadoPasante {  
    public String nombre;  
    public String apellido;  
    public double sueldoBasico = 0;
```

## Solucion:

- Muevo estos atributos a una superclase abstracta Empleado

```
public abstract class Empleado{  
    public String nombre;  
    public String apellido;  
    public double sueldoBasico = 0;  
}
```

```
public class EmpleadoTemporario extends Empleado{  
    public double horasTrabajadas = 0;  
    public int cantidadHijos = 0;  
    // .....  
  
    public double sueldo() {  
        return this.sueldoBasico  
            + (this.horasTrabajadas * 500)  
            + (this.cantidadHijos * 1000)  
            - (this.sueldoBasico * 0.13);  
    }  
}
```

```
public class EmpleadoPlanta extends Empleado{  
    public int cantidadHijos = 0;  
    // .....  
  
    public double sueldo() {  
        return this.sueldoBasico  
            + (this.cantidadHijos * 2000)  
            - (this.sueldoBasico * 0.13);  
    }  
}
```

```
}  
}
```

```
public class EmpleadoPasante extends Empleado{  
    // .....  
  
    public double sueldo() {  
        return this.sueldoBasico - (this.sueldoBasico * 0.13);  
    }  
}
```

## Code smells

1. Atributos publicos en la clase abstracta Empleado, EmpleadoPlanta y EmpleadoTemporario. Esto viola el encapsulamiento de las variables.

## Solucion

- Pasarlas a privadas/protected (esto altera el comportamiento).

```
public abstract class Empleado{  
    private String nombre;  
    private String apellido;  
    protected double sueldoBasico = 0;  
}
```

```
public class EmpleadoTemporario extends Empleado{  
    private double horasTrabajadas = 0;  
    private int cantidadHijos = 0;  
    // .....  
  
    public double sueldo() {  
        return this.sueldoBasico  
            + (this.horasTrabajadas * 500)  
            + (this.cantidadHijos * 1000)  
            - (this.sueldoBasico * 0.13);  
    }  
}
```

```
}  
}
```

```
public class EmpleadoPlanta extends Empleado{  
    private int cantidadHijos = 0;  
    // .....  
  
    public double sueldo() {  
        return this.sueldoBasico  
            + (this.cantidadHijos * 2000)  
            - (this.sueldoBasico * 0.13);  
    }  
}
```

```
public class EmpleadoPasante extends Empleado{  
    // .....  
  
    public double sueldo() {  
        return this.sueldoBasico - (this.sueldoBasico * 0.13);  
    }  
}
```

## Code smells

1. Duplicated code en el metodo sueldo la sentencia "- (this.sueldoBasico \* 0.13);"de las 3 clases hijas de Empleado

```
public double sueldo() {  
    return this.sueldoBasico  
        + (this.horasTrabajadas * 500)  
        + (this.cantidadHijos * 1000)  
        - (this.sueldoBasico * 0.13);  
}
```

```
public double sueldo() {  
    return this.sueldoBasico
```

```
        + (this.cantidadHijos * 2000)
        - (this.sueldoBasico * 0.13);
    }
```

```
public double sueldo() {
    return this.sueldoBasico - (this.sueldoBasico * 0.13);
}
```

## Solucion

- Extract Method

```
public abstract class Empleado{
    private String nombre;
    private String apellido;
    protected double sueldoBasico = 0;
}
```

```
public class EmpleadoTemporario extends Empleado{
    private double horasTrabajadas = 0;
    private int cantidadHijos = 0;
    // .....

    public double descuentoAportes (){
        return this.sueldoBasico * 0.13;
    }

    public double sueldo() {
        return this.sueldoBasico
            + (this.horasTrabajadas * 500)
            + (this.cantidadHijos * 1000)
            - this.descuentoAportes();
    }
}
```

```

public class EmpleadoPlanta extends Empleado{
    private int cantidadHijos = 0;
    // .....

    public double descuentoAportes (){
        return this.sueldoBasico * 0.13;
    }

    public double sueldo() {
        return this.sueldoBasico
            + (this.cantidadHijos * 2000)
            - this.descuentoAportes();
    }
}

```

```

public class EmpleadoPasante extends Empleado{
    // .....

    public double descuentoAportes (){
        return this.sueldoBasico * 0.13;
    }

    public double sueldo() {
        return this.sueldoBasico - this.descuentoAportes();
    }
}

```

## Code smells

1. Código Repetido en las 3 subclases empleado

```

public double descuentoAportes (){
    return this.sueldoBasico * 0.13;
}

```

## Solucion

- Extract method de descuentoAportes()

```
public abstract class Empleado{
    private String nombre;
    private String apellido;
    protected double sueldoBasico = 0;

}
```

```
public class EmpleadoTemporario extends Empleado{
    private double horasTrabajadas = 0;
    private int cantidadHijos = 0;
    // .....

    public double descuentoAportes (){
        return this.sueldoBasico * 0.13;
    }

    public double sueldoConDescuento (){
        return this.sueldoBasico - this.descuentoAportes();
    }

    public double sueldo() {
        return this.sueldoConDescuento()
            + (this.horasTrabajadas * 500)
            + (this.cantidadHijos * 1000)
    }
}
```

```
public class EmpleadoPlanta extends Empleado{
    private int cantidadHijos = 0;
    // .....

    public double descuentoAportes (){
        return this.sueldoBasico * 0.13;
    }
}
```



```

public double sueldoConDescuento (){
    return this.sueldoBasico - this.descuentoAportes();
}

public double sueldo() {
    return this.sueldoConDescuento()
        + (this.cantidadHijos * 2000)
        - this.descuentoAportes();
}
}

```

```

public class EmpleadoPasante extends Empleado{
    // .....

    public double descuentoAportes (){
        return this.sueldoBasico * 0.13;
    }

    public double sueldoConDescuento (){
        return this.sueldoBasico - this.descuentoAportes();
    }

    public double sueldo() {
        return this.sueldoConDescuento();
    }
}

```

## Code smells

1. Código duplicado del método `descuentoAportes()` y `sueldoConDescuento()` de las 3 clases hijas de `Empleado`.

```

public double descuentoAportes (){
    return this.sueldoBasico * 0.13;
}

public double sueldoConDescuento (){

```

```
        return this.sueldoBasico - this.descuentoAportes();  
    }  
}
```

## Solucion

- Pull up method de los metodos descuentoAportes() y sueldoConDescuento()

```
public abstract class Empleado{  
    private String nombre;  
    private String apellido;  
    protected double sueldoBasico = 0;  
  
    public double descuentoAportes(){  
        return this.sueldoBasico * 0.13;  
    }  
  
    public double sueldoConDescuento (){  
        return this.sueldoBasico - this.descuentoAportes();  
    }  
}
```

```
public class EmpleadoTemporario extends Empleado{  
    private double horasTrabajadas = 0;  
    private int cantidadHijos = 0;  
    // .....  
  
    public double sueldo() {  
        return this.sueldoConDescuento()  
            + (this.horasTrabajadas * 500)  
            + (this.cantidadHijos * 1000);  
    }  
}
```

```
public class EmpleadoPlanta extends Empleado{  
    private int cantidadHijos = 0;
```

```
// .....

public double sueldo() {
    return this.sueldoConDescuento();
        + (this.cantidadHijos * 2000)
}
}
```

```
public class EmpleadoPasante extends Empleado{
    // .....

    public double sueldo() {
        return this.sueldoConDescuento();
    }
}
```

## Code smells

1. Duplicated code en la sentencia "+ (this.cantidadHijos \* 1000);" del metodo sueldo en la clase EmpleadoPlanta y EmpleadoTemporario

## Solucion

- Extract method de la sentencia "+ (this.cantidadHijos \* 1000);"

```
public abstract class Empleado{
    private String nombre;
    private String apellido;
    protected double sueldoBasico = 0;

    public double descuentoAportes(){
        return this.sueldoBasico * 0.13;
    }

    public double sueldoConDescuento (){
        return this.sueldoBasico - this.descuentoAportes();
    }
}
```

```
}  
}
```

```
public class EmpleadoTemporario extends Empleado{  
    private double horasTrabajadas = 0;  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 1000;  
    }  
  
    public double sueldo() {  
        return this.sueldoConDescuento()  
            + (this.horasTrabajadas * 500)  
            + this.asignacionPorHijo();  
    }  
}
```

```
public class EmpleadoPlanta extends Empleado{  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 2000;  
    }  
  
    public double sueldo() {  
        return this.sueldoConDescuento()  
            + this.asignacionPorHijo();  
    }  
}
```

```
public class EmpleadoPasante extends Empleado{  
    // .....
```

```
public double sueldo() {  
    return this.sueldoConDescuento();  
}  
}
```

## Code smells

### 1. Duplicated code en las clases EmpleadoTemporario y EmpleadoPlanta

```
public class EmpleadoTemporario extends Empleado{  
    private double horasTrabajadas = 0;  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 1000;  
    }  
  
    public double sueldo() {  
        return this.sueldoConDescuento()  
            + (this.horasTrabajadas * 500)  
            + this.asignacionPorHijo();  
    }  
}
```

```
public class EmpleadoPlanta extends Empleado{  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 2000;  
    }  
  
    public double sueldo() {  
        return this.sueldoConDescuento()  
            + this.asignacionPorHijo();  
    }  
}
```

```
}  
}
```

## Solucion

- Pull Up method del metodo asignacionPorHijo(). Convertirlo en metodo abstracto ya que se comporta de distinta formas en las distintas subclases de Empleado

```
public abstract class Empleado{  
    private String nombre;  
    private String apellido;  
    protected double sueldoBasico = 0;  
  
    public double descuentoAportes(){  
        return this.sueldoBasico * 0.13;  
    }  
  
    public double sueldoConDescuento (){  
        return this.sueldoBasico - this.descuentoAportes();  
    }  
  
    public abstract int asignacionPorHijo();  
}
```

```
public class EmpleadoTemporario extends Empleado{  
    private double horasTrabajadas = 0;  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 1000;  
    }  
  
    public double sueldo() {  
        return this.sueldoConDescuento()  
            + (this.horasTrabajadas * 500)  
            + this.asignacionPorHijo();  
    }  
}
```

```
}  
}
```

```
public class EmpleadoPlanta extends Empleado{  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 2000;  
    }  
  
    public double sueldo() {  
        return this.sueldoConDescuento()  
            + this.asignacionPorHijo();  
    }  
}
```

```
public class EmpleadoPasante extends Empleado{  
    // .....  
  
    public int asignacionPorHijo(){  
        return 0;  
    }  
  
    public double sueldo() {  
        return this.sueldoConDescuento();  
    }  
}
```

## Code smell

1. Duplicated code

## Solucion

Extract method de la sentencia "(this.horasTrabajadas \* 500)" de la clase EmpleadoTemporario

```

public abstract class Empleado{
    private String nombre;
    private String apellido;
    protected double sueldoBasico = 0;

    public double descuentoAportes(){
        return this.sueldoBasico * 0.13;
    }

    public double sueldoConDescuento (){
        return this.sueldoBasico - this.descuentoAportes();
    }

    public abstract int asignacionPorHijo();
}

```

```

public class EmpleadoTemporario extends Empleado{
    private double horasTrabajadas = 0;
    private int cantidadHijos = 0;
    // .....

    public int asignacionPorHijo(){
        return this.cantidadHijos * 1000;
    }

    public int calcularHorasTrabajadas(){
        return this.horasTrabajadas * 500;
    }

    public double sueldo() {
        return this.sueldoConDescuento()
            + this.calcularHorasTrabajadas()
            + this.asignacionPorHijo();
    }
}

```



```

public class EmpleadoPlanta extends Empleado{
    private int cantidadHijos = 0;
    // .....

    public int asignacionPorHijo(){
        return this.cantidadHijos * 2000;
    }

    public double sueldo() {
        return this.sueldoConDescuento()
            + this.asignacionPorHijo();
    }
}

```

```

public class EmpleadoPasante extends Empleado{
    // .....

    public int asignacionPorHijo(){
        return 0;
    }

    public double sueldo() {
        return this.sueldoConDescuento();
    }
}

```

## Code smell

1. Subo metodo calcularHorasTrabajadas() para poder llegar al patter design "Template method"

## Solucion

- Pull Up method

```

public abstract class Empleado{
    private String nombre;
    private String apellido;

```

```

protected double sueldoBasico = 0;

public double descuentoAportes(){
    return this.sueldoBasico * 0.13;
}

public double sueldoConDescuento (){
    return this.sueldoBasico - this.descuentoAportes();
}

public abstract int asignacionPorHijo();
public int calcularHorasTrabajadas();
}

```

```

public class EmpleadoTemporario extends Empleado{
    private double horasTrabajadas = 0;
    private int cantidadHijos = 0;
    // .....

    public int asignacionPorHijo(){
        return this.cantidadHijos * 1000;
    }

    public int calcularHorasTrabajadas(){
        return this.horasTrabajadas * 500;
    }

    public double sueldo() {
        return this.sueldoConDescuento()
            + this.calcularHorasTrabajadas()
            + this.asignacionPorHijo();
    }
}

```

```

public class EmpleadoPlanta extends Empleado{
    private int cantidadHijos = 0;
    // .....

```

```

public int asignacionPorHijo(){
    return this.cantidadHijos * 2000;
}

public int calcularHorasTrabajadas(){
    return 0;
}

public double sueldo() {
    return this.sueldoConDescuento()
        + this.asignacionPorHijo();
}
}

```

```

public class EmpleadoPasante extends Empleado{
    // .....

    public int asignacionPorHijo(){
        return 0;
    }

    public int calcularHorasTrabajadas(){
        return 0;
    }

    public double sueldo() {
        return this.sueldoConDescuento();
    }
}

```

## Code smell

1. Duplicated code en el metodo sueldo de las 3 subclases de Empleado

```

public class EmpleadoTemporario extends Empleado{
    private double horasTrabajadas = 0;
    private int cantidadHijos = 0;
}

```

```

// .....

public int asignacionPorHijo(){
    return this.cantidadHijos * 1000;
}

public int calcularHorasTrabajadas(){
    return this.horasTrabajadas * 500;
}

public double sueldo() {
    return this.sueldoConDescuento()
        + this.calcularHorasTrabajadas()
        + this.asignacionPorHijo();
}
}

```

```

public class EmpleadoPlanta extends Empleado{
    private int cantidadHijos = 0;
    // .....

    public int asignacionPorHijo(){
        return this.cantidadHijos * 2000;
    }

    public int calcularHorasTrabajadas(){
        return 0;
    }

    public double sueldo() {
        return this.sueldoConDescuento()
            + this.asignacionPorHijo();
    }
}

```

```

public class EmpleadoPasante extends Empleado{
    // .....

```

```

public int asignacionPorHijo(){
    return 0;
}

public int calcularHorasTrabajadas(){
    return 0;
}

public double sueldo() {
    return this.sueldoConDescuento();
}
}

```

## Solucion

- Pull up method del metodo sueldo()

```

public abstract class Empleado{
    private String nombre;
    private String apellido;
    protected double sueldoBasico = 0;

    public double descuentoAportes(){
        return this.sueldoBasico * 0.13;
    }

    public double sueldoConDescuento (){
        return this.sueldoBasico - this.descuentoAportes();
    }

    public double sueldo(){
        return this.sueldoConDescuento + this.asignacionPorHijo() + this.calc
ularHorasTrabajadas();
    }

    public abstract int asignacionPorHijo();
}

```

```
    public int calcularHorasTrabajadas();  
}
```

```
public class EmpleadoTemporario extends Empleado{  
    private double horasTrabajadas = 0;  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 1000;  
    }  
  
    public int calcularHorasTrabajadas(){  
        return this.horasTrabajadas * 500;  
    }  
}
```

```
public class EmpleadoPlanta extends Empleado{  
    private int cantidadHijos = 0;  
    // .....  
  
    public int asignacionPorHijo(){  
        return this.cantidadHijos * 2000;  
    }  
  
    public int calcularHorasTrabajadas(){  
        return 0;  
    }  
}
```

```
public class EmpleadoPasante extends Empleado{  
    // .....  
  
    public int asignacionPorHijo(){  
        return 0;  
    }  
}
```

```
public int calcularHorasTrabajadas(){  
    return 0;  
}  
}
```