



Ejercicio 01.

A la vista del siguiente código, identifique y aplique las refactorizaciones que considere más convenientes. (Contestar un texto con una explicación de las mejoras y volver a generar el código con las mejoras propuestas).

```
public class Persona {
    String numeroDeTelefono;

    public Persona(String numeroDeTelefono) {
        super();
        this.numeroDeTelefono = numeroDeTelefono;
    }

    public String getNumeroDeTelefono() {
        return numeroDeTelefono;
    }

    public void setNumeroDeTelefono(String numeroDeTelefono) {
        this.numeroDeTelefono = numeroDeTelefono;
    }
}

public class Profesor extends Persona {

    String str;
    int edad;
    String numeroDeTelefono;
    List<Prestamo> prestamos;

    public Profesor(String numeroDeTelefono) {
        super(numeroDeTelefono);
    }

    public void printInformacionPersonal() {
        System.out.println("Nombre: " + str);
        System.out.println("Edad: " + edad);
        System.out.println("Teléfono: " + numeroDeTelefono);
    }

    public void printTodaLaInformacion() {
        System.out.println("Nombre: " + str);
        System.out.println("Edad: " + edad);
        System.out.println("Teléfono: " + this.numeroDeTelefono);
        for (Prestamo p: prestamos) {
            System.out.println(p);
        }
    }
}
```



Ejercicio 02.

A continuación tiene un fragmento de un juego, en concreto la clase que se encarga de ver el movimiento que se desea hacer y mover las coordenadas del jugador en dicha dirección (considerando que el punto 0,0 está arriba a la izquierda).

```
public class Game {  
    Player p;  
    //...  
    public void movement(String m) {  
        if (m.equalsIgnoreCase("Derecha")) {  
            p.setX(p.getX()+1);  
        }  
        if (m.equalsIgnoreCase("Izquierda")) {  
            p.setX(p.getX()-1);  
        }  
        if (m.equalsIgnoreCase("Arriba")) {  
            p.setY(p.getY()-1);  
        }  
        if (m.equalsIgnoreCase("Abajo")) {  
            p.setY(p.getY()+1);  
        }  
    }  
}  
  
public class Player {  
    int x, y;  
    public int getX() {  
        return x;  
    }  
    public void setX(int x) {  
        this.x = x;  
    }  
    public int getY() {  
        return y;  
    }  
    public void setY(int y) {  
        this.y = y;  
    }  
}
```

Identifique qué refactorizaciones puede realizar en ambas clases. (Contestar un texto con una explicación de las mejoras y volver a generar el código con las mejoras propuestas).