

Ejercicio DNI

Desarrolla una clase que represente un DNI español. Tendrá las siguientes características:

- La numeración se almacenará en un atributo de tipo int privado.
- Para acceder al DNI, tendremos dos getters:
 - getDNI() devuelve un entero con los números del DNI.
 - getNIF() devuelve un String con el NIF completo.
- Para grabar en el atributo, usaremos un setter con sobrecarga:
 - setDNI(String NIF)
 - setDNI(int DNI)
- Usaremos otros métodos para operaciones internas:
 - calcularLetraNIF (int dni)
 - validarNIF (String nif)
 - extraerLetraNIF (String nif)
 - extraerNumeroNIF (String nif)

Cómo calcular la letra del DNI: [Info](#)

Resuelto en la siguiente página. Inténtalo antes de mirar, en el examen no estará resuelto... 🤖💀😁

Clase DNI

```
public class DNI {

    // Atributos
    private int numDni;
    private static final String LETRAS = "TRWAGMYFPDXBNJZSQVHLCKE";

    // Métodos GETTERS
    public int getDNI(){
        return numDni;
    }
    public String getNIF() {
        char letra;
        letra = DNI.calcularLetraNIF(numDni);
        String nif = Integer.toString(this.numDni) + letra;
        return nif;
    }

    // Métodos SETTERS
    public boolean setDNI(int numDni){
        this.numDni = numDni;
        return true;
    }
    public boolean setDNI(String nif){
        boolean resultado = true;
        nif = nif.toUpperCase();
        if (!this.validarNIF(nif)){
            resultado = false;
        }else {
            int numDni = this.extraerNumeroNIF(nif);
            this.numDni = numDni;
        }
        return resultado;
    }

    // Otros métodos internos
    private static char calcularLetraNIF(int dni){
        int resto = dni % 23;
        char letra = LETRAS.charAt(resto);
        return letra;
    }
    private boolean validarNIF(String nif){
        char letra;
        int num;
        boolean valido = false;
        //Compruebo si longitud es 9 y si el ultimo caracter es una letra
        if (nif.length() != 9 || Character.isLetter(nif.charAt(8)) == false ){
            return valido;
        }
        letra = extraerLetraNIF(nif);
        num = extraerNumeroNIF(nif);
        if (letra == calcularLetraNIF(num)){
            valido = true; }
        return valido;
    }
    private static char extraerLetraNIF(String nif){
        char letra;
        letra = nif.charAt(8);
        return letra;
    }
    private static int extraerNumeroNIF(String nif){
        int numero;
        numero = Integer.parseInt(nif.substring(0,8));
        return numero;
    }
}
```

Clase Main

```
public class Main {  
  
    public static void main(String[] args){  
  
        DNI dni1 = new DNI();  
  
        if(!dni1.setDNI("77322947V")){  
            System.out.println("El DNI 77322947V no es válido");  
        }else{  
            System.out.println("El DNI 77322947V es válido");  
            System.out.println("dni1.getNIF() = " + dni1.getNIF());  
            System.out.println("dni1.getDNI() = " + dni1.getDNI());  
        }  
  
        if(!dni1.setDNI("77322947Y")){  
            System.out.println("El DNI 77322947Y no es válido");  
        }else{  
            System.out.println("El DNI 77322947Y es válido");  
            System.out.println("dni1.getNIF() = " + dni1.getNIF());  
            System.out.println("dni1.getDNI() = " + dni1.getDNI());  
        }  
    }  
}
```