

# Ejercicios POO

## Ejercicio 1: Persona

```
Welcome POO.py M dni.py M X persona.py M
F: > asignaturas_segundo > desarrollo_de_interfaces > Tema_1 > actividades >
1 class Dni:
2     def __init__(self, numero):
3         self.numero=numero
4
5     def __calcular_letra(self):
6         letras='TRWAGMYFPDXBNJZSQVHLCKE'
7         return letras[int(self.numero)%23]
8
9     @property
10    def numero(self):
11        return self.__numero
12
13    @numero.setter
14    def numero(self, numero):
15        if len(numero)==8 and numero.isdigit():
16            self.__numero=numero
17            self.__letra=self.__calcular_letra()
18        else:
19            self.__numero=0
20            self.__letra=""
21            print("DNI Incorrecto")
22
23    @property
24    def letra(self):
25        return self.__letra
26
27    def mostrar(self):
28        return self.numero+self.letra
```

```

from dni import Dni

class Persona:

    def __init__(self,nombre="",edad="",dni=""):
        self.nombre=nombre
        self.edad=edad
        self.dni=dni

    @property
    def nombre(self):
        return self.__nombre

    @nombre.setter
    def nombre(self,nombre):
        self.__nombre=nombre

    @property
    def edad(self):
        return self.__edad

    @edad.setter
    def edad(self,edad):
        if(edad.isdigit() and int(edad)>=0):
            self.__edad=edad
        else:
            self.__edad=""
            print("LA EDAD DEBE SER UN NÚMERO ENTERO POSITIVO")

    @property
    def dni(self):
        return self.__dni

    @dni.setter
    def dni(self,dni):
        if (len(dni)==9):
            numero=""
            i=0
            while(i<8):
                numero=numero+dni[i]
                i+=1
            letra=dni[8]
            dniCorrecto=Dni(numero)
            if(dniCorrecto.letra==letra):
                self.__dni=dni
            else:
                self.__dni="*FALTA DNI*"
                print("DNI INCORRECTO")
        else:
            self.__dni="*FALTA DNI*"
            print("DNI INCORRECTO")

    def mostrar(self):
        return self.nombre+" (" +str(self.edad)+") "+self.dni

    def esMayorDeEdad(self):
        esMayor=False
        if(self.__edad.isdigit() and int(self.edad)>=18):
            esMayor=True
        return esMayor

```

## Ejercicio 2: Cuenta

```

Welcome  POO.py M  dni.py M  persona.py M  cuenta.py M X  cuentajc
F: > asignaturas_segundo > desarrollo_de_interfaces > Tema_1 > actividades > DI-primeros_ejercicios_python > ejercicios

1  from persona import Persona
2
3  class Cuenta:
4
5      def __init__(self, titular, cantidad=0):
6          self.titular=titular
7          self.__cantidad=cantidad
8
9      @property
10     def titular(self):
11         return self.__titular
12
13     @titular.setter
14     def titular(self, titular):
15         self.__titular=titular
16
17     @property
18     def cantidad(self):
19         return self.__cantidad
20
21     def retirar(self, importe):
22         if(importe>0):
23             self.__cantidad=self.__cantidad-importe
24             print("Importe retirado: "+str(importe)+"€")
25         else:
26             self.__cantidad=self.__cantidad
27             print("No es posible retirar "+str(importe)+"€")
28
29     def ingresar(self, importe):
30         if(importe<0):
31             self.__cantidad=self.__cantidad
32             print("No es posible ingresar "+str(importe)+"€")
33         else:
34             self.__cantidad=self.__cantidad+importe
35             print("Importe ingresado: "+str(importe)+"€")
36
37     def mostrar(self):
38         return self.__titular.mostrar()+" "+str(self.__cantidad)+"€"
```

### Ejercicio 3: Cuenta joven

```
1 from cuenta import Cuenta
2
3 class CuentaJoven(Cuenta):
4
5     def __init__(self, titular, cantidad=0, bonificacion=0):
6         super().__init__(titular, cantidad)
7         if (super().titular.esMayorDeEdad()):
8             self.__cuentaValida=True
9             print("Titular válido")
10        else:
11            self.__cuentaValida=False
12            print("No es un titular válido")
13
14        self.bonificacion=bonificacion
15
16    @property
17    def cuentaValida(self):
18        return self.__cuentaValida
19
20    @property
21    def bonificacion(self):
22        return self.__bonificacion
23
24    @bonificacion.setter
25    def bonificacion(self, bonificacion):
26        self.__bonificacion=bonificacion
27
28    def esTitularValido(self):
29        titularValido=False
30        if super().titular.esMayorDeEdad() and int((super().titular.edad)<25:
31            titularValido=True
32        return titularValido
33
34    def retirar(self, importe):
35        if (self.esTitularValido()):
36            super().retirar(importe)
37        else:
38            print("Este titular NO puede retirar dinero de esta cuenta.")
39
40    def ingresar(self, importe):
41        super().ingresar(importe)
42
43    def mostrar(self):
44        return "Cuenta joven"+"nTitular: "+super().titular.mostrar()+"nBonificacion:"+str(self.__bonificacion)+"%"+nCantidad:"+str((super().cantidad))+€"
```

### Ejecución:

```
1
2 from persona import Persona
3 from cuentajoven import CuentaJoven
4
5 print("CREACION DE CUENTA JOVEN\n")
6
7 print("Introduzca los datos del cliente.")
8 nombre=(str(input("Nombre:")))
9 edad=(str(input("Edad:")))
10 dni=(str(input("DNI:")))
11
12 persona=Persona(nombre,edad,dni)
13
14 cuentaJoven=CuentaJoven(persona,0,2)
15 if(cuentaJoven.cuentaValida):
16
17     print("\nSe ha creado una cuenta joven.\n")
18     print(cuentaJoven.mostrar())
19
20     ingreso=(int(input("\nCantidad a ingresar(€):")))
21     cuentaJoven.ingresar(ingreso)
22     print()
23     print(cuentaJoven.mostrar())
24
25     retirar=(int(input("\nCantidad a retirar(€):")))
26     cuentaJoven.retirar(retirar)
27     print()
28     print(cuentaJoven.mostrar())
29 else:
30     del cuentaJoven
```

```
python\ejercicios_P00\Programa.py
CREACION DE CUENTA JOVEN
```

```
Introduzca los datos del cliente.
Nombre:Juan
Edad:23
DNI:53191630J
Titular válido
```

Se ha creado una cuenta joven.

```
Cuenta joven
Titular: Juan (23) 53191630J
Bonificacion:2%
Cantidad:0€
```

```
Cantidad a ingresar(€):200
Importe ingresado: 200€
```

```
Cuenta joven
Titular: Juan (23) 53191630J
Bonificacion:2%
Cantidad:200€
```

```
Cantidad a retirar(€):300
Importe retirado: 300€
```

```
Cuenta joven
Titular: Juan (23) 53191630J
Bonificacion:2%
Cantidad:-100€
```

```
PS E:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_python\ejercicios_P00> █
```

```
bugpy\launcher - 5/23/21 -- e:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_python\
CREACION DE CUENTA JOVEN
```

```
Introduzca los datos del cliente.
Nombre:Juan
Edad:17
DNI:53191630J
No es un titular válido
```

```
PS E:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_python\ejercicios_P00> █
```

```
bugpy\launcher - 5/23/21 -- e:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_python\
CREACION DE CUENTA JOVEN
```

```
Introduzca los datos del cliente.
Nombre:Juan
Edad:45
DNI:123432
DNI INCORRECTO
Titular válido
```

Se ha creado una cuenta joven.

```
Cuenta joven
Titular: Juan (45) *FALTA DNI*
Bonificacion:2%
Cantidad:0€
```

```
Cantidad a ingresar(€):300
Importe ingresado: 300€
```

```
Cuenta joven
Titular: Juan (45) *FALTA DNI*
Bonificacion:2%
Cantidad:300€
```

```
Cantidad a retirar(€):100
Este titular NO puede retirar dinero de esta cuenta.
```

```
Cuenta joven
Titular: Juan (45) *FALTA DNI*
Bonificacion:2%
Cantidad:300€
```

```
PS E:\asignaturas_segundo\desarrollo_de_interfaces\Tema_1\actividades\DI-primeros_ejercicios_python\ejercicios_P00> █
```