Creado por:

Isabel Maniega

Ejercicios

1) Crea el siguiente programa:

- Una clase de nombre Librería
- Inicia los siguientes atributos: nombre, sección, editorial y año
- Crea una segunda clase con nombre Rosalia que herede la clase librería.
- En esta clase Rosalia, crea una función "result" cuyo resultado sea los datos de los libros.
- declara los Objetos siguientes:
  - libro1 --> Oceanarium, Ciencia, Impedimenta, 2021
  - libro2 --> 33 Botones, Novela negra, Atlantis, 2022
  - libro3 --> Venganza en Compostela, Historia, Universo de letras, 2022

In [1]:
```
class Libreria:
    def __init__(self, nombre, seccion, editorial, año):
        self.nombre = nombre
        self.seccion = seccion
        self.editorial = editorial
        self.año = año
```

In [2]:
```
class Rosalia(Libreria):

    def result(self):
        print("Información del libro es %s, de la editorial %s, cuya secc
              % (self.nombre, self.editorial, self.seccion, self.año))
```

In [3]:
```
libro1 = Rosalia("Oceanarium", "Ciencia", "Impedimenta", 2021)
libro2 = Rosalia("33 Botones", "Novela negra", "Atlantis", 2022)
libro3 = Rosalia("Venganza en Compostela", "Historia", "Universo de letra
```

In [4]:
```
libro1.result()
```

```
Información del libro es Oceanarium, de la editorial Impedimenta, cuya sec
cion es Ciencia, del año 2021
```

2) Crea otra libraría de nombre MiLibro, que corresponde a una nueva clase, define una función de nombre misLibros, cuyo resultado sea los datos de los libros:

- libro4 --> Mi primera Novela, Novela, Bruño, 2019
- libro5 --> Gatos, Literatura, Listado, 2018

In [5]:
```
class MiLibro(Libreria):

    def misLibros(self):
        print("Información del libro es %s, de la editorial %s, cuya secc
              % (self.nombre, self.editorial, self.seccion, self.año))
```

```
    def años(self):
        return self.año
```

In [6]:
```
libro4 = MiLibro("Mi primera Novela", "Novela", "Bruño", 2019)
libro5 = MiLibro("Gatos", "Literatura", "Listado", 2018)
```

In [7]:
```
libro4.misLibros()
```

```
Información del libro es Mi primera Novela, de la editorial Bruño, cuya se
ccion es Novela, del año 2019
```

- Realiza la media de los años de los libros 4 y 5

In [8]:
```
media = (libro4.años() + libro5.años())/2
media
```

Out[8]: 2018.5

3) Crea una clase llamada Persona. Sus atributos son: nombre, edad y DNI. Construye los siguientes métodos para la clase:

- Un constructor, donde los datos pueden estar vacíos.
- mostrar(): Muestra los datos de la persona.
- esMayorDeEdad(): Devuelve un valor indicando si es mayor de edad.

In [2]:
```
class Persona:

    def __init__(self, nombre, edad, DNI):
        self.nombre = nombre
        self.edad = edad
        self.DNI = DNI

    def mostrar(self):
        print("Nombre:"+self.nombre+" - Edad:"+str(self.edad)+" - DNI:"+s


    def esMayorDeEdad(self):
        return self.edad>=18
```

In [10]:
```
persona1 = Persona("Juan García", 15, "74152864W")
persona2 = Persona("María Perez", 20, "29152875W")
```

In [11]:
```
persona1.mostrar()
```

```
Nombre:Juan García - Edad:15 - DNI:74152864W
```

In [12]:
```
persona2.mostrar()
```

```
Nombre:María Perez - Edad:20 - DNI:29152875W
```

In [13]:
```
persona1.esMayorDeEdad()
```

Out[13]: False

In [14]:
```
persona2.esMayorDeEdad()
```

Out[14]: True

4) Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Construye los siguientes métodos para la clase:

- Un constructor, donde los datos pueden estar vacíos.
- El atributo no se puede modificar directamente, sólo ingresando o retirando dinero.
- mostrar(): Muestra los datos de la cuenta.
- ingresar(cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
- retirar(cantidad): se retira una cantidad a la cuenta. La cuenta puede estar en números rojos si es saldo negativo.

In [3]:
```
class Cuenta(Persona):

    def __init__(self,titular,cantidad=0):
        self.titular=titular
        self.cantidad=cantidad

    def mostrar(self):
        return "Titular de la cuenta: "+self.titular +" - Cantidad: "+str

    def ingresar(self,cantidad):
        if cantidad > 0:
            self.cantidad = self.cantidad + cantidad

    def retirar(self,cantidad):
        if cantidad > 0:
            self.cantidad = self.cantidad - cantidad

        if self.cantidad < 0:
            return f"ATENCIÓN! La cuenta está en números rojos: {self.can
```

In [16]:
```
titular1 = Cuenta("José Rodríguez")
```

In [17]:
```
titular1.mostrar()
```

Out[17]: 'Titular de la cuenta: José Rodríguez - Cantidad: 0'

- Ingresar en positivo

In [18]:
```
titular1.ingresar(20)
```

In [19]:
```
titular1.mostrar()
```

Out[19]: 'Titular de la cuenta: José Rodríguez - Cantidad: 20'

- Ingresar en negativo

In [20]:
```
titular1.ingresar(-10)
```

In [21]:
```
titular1.mostrar()
```

Out[21]:
```
'Titular de la cuenta: José Rodríguez - Cantidad: 20'
```

- Retirar dinero

In [22]:
```
titular1.retirar(5)
```

In [23]:
```
titular1.mostrar()
```

Out[23]:
```
'Titular de la cuenta: José Rodríguez - Cantidad: 15'
```

- Retirar dinero en numeros rojos

In [24]:
```
titular1.retirar(20)
```

Out[24]:
```
'ATENCIÓN! La cuenta está en números rojos: -5'
```

In [25]:
```
titular1.mostrar()
```

Out[25]:
```
'Titular de la cuenta: José Rodríguez - Cantidad: -5'
```

5) Vamos a definir ahora una "Cuenta Joven", para ello vamos a crear una nueva clase CuantaJoven que deriva de la anterior. Cuando se crea esta nueva clase, además del titular y la cantidad se debe guardar una bonificación que estará expresada en tanto por ciento.Construye los siguientes métodos para la clase:

- Un constructor.
- En esta ocasión los titulares de este tipo de cuenta tienen que ser mayor de edad., por lo tanto hay que crear un método esTitularValido() que devuelve verdadero si el titular es mayor de edad pero menor de 25 años y falso en caso contrario.
- Además la retirada de dinero sólo se podrá hacer si el titular es válido.
- El método mostrar() debe devolver el mensaje de "Cuenta Joven" y la bonificación de la cuenta.
- Piensa los métodos heredados de la clase madre que hay que reescribir.

In [7]:
```
class CuentaJoven(Cuenta):

    def __init__(self,titular, edad, cantidad=0, bonificacion=0):
        super().__init__(titular,cantidad)
        self.bonificacion=bonificacion
        self.edad = edad

    def mostrar(self):
        return "-Cuenta Joven- "+"Titular:"+self.titular +" - Cantidad:"+

    def esTitularValido(self):
        return self.edad < 25 and self.esMayorDeEdad()

    def retirar(self,cantidad):
        if not self.esTitularValido():
            print ("No puesdes retirar el dinero. titular no válido")
```

```
        elif cantidad > 0:
            super().retirar(cantidad)
```

In [8]:
```
titular2 = CuentaJoven("Pedro Alonso", 30)
titular3 = CuentaJoven("Lorena García", 20, 200, 5)
```

- Mayor de edad

In [9]:
```
titular2.mostrar()
```

Out[9]:
```
'-Cuenta Joven- Titular:Pedro Alonso - Cantidad:0- Bonificación:0%'
```

In [10]:
```
titular2.ingresar(100)
```

In [11]:
```
titular2.mostrar()
```

Out[11]:
```
'-Cuenta Joven- Titular:Pedro Alonso - Cantidad:100- Bonificación:0%'
```

In [12]:
```
titular2.retirar(10)
```

```
No puesdes retirar el dinero. titular no válido
```

- Menor de edad

In [32]:
```
titular3.ingresar(1000)
```

In [33]:
```
titular3.mostrar()
```

Out[33]:
```
'-Cuenta Joven- Titular:Lorena García - Cantidad:1200- Bonificación:5%'
```

In [34]:
```
titular3.retirar(200)
```

In [35]:
```
titular3.mostrar()
```

Out[35]:
```
'-Cuenta Joven- Titular:Lorena García - Cantidad:1000- Bonificación:5%'
```

Creado por:

Isabel Maniega