

PROGRAMACIÓN

UNIDAD 4: DESARROLLO DE CLASES

CASO 3

Realizado por:
María Gómez Irujo.
Primer curso de DAW.

TÍTULO:

DESARROLLO DE CLASES

SITUACIÓN:

Tras haber realizado la clase del caso práctico anterior (CP2) de este tema.

INSTRUCCIONES:

Vamos a definir ahora una "Cuenta Joven", para ello vamos a crear una nueva clase CuentaJoven 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.
- Los setters y getters para el nuevo atributo.
- 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.

RESOLUCIÓN:

He creado una clase Cuentajoven que es subclase/clase hija de la clase Cuenta creada en el caso práctico anterior, esto se indica mediante la palabra reservada extends.

```
public class Cuentajoven extends Cuenta {
```

La clase hija Cuentajoven hereda de la clase Cuenta sus atributos y métodos, para que estos atributos puedan ser heredados los hemos tenido que cambiar de private a protected. Hemos creado otros dos atributos más: bonificación y edad, ambos como privados ya que no tienen que ser heredados por ninguna otra clase.

```
private double bonificacion= 10;  
private int edad;
```

He creado un constructor sin parámetros y los métodos getters y setters para los nuevos atributos especificados:

```

//constructor

public Cuentajoven () {

}

//metodos getter y setter

public double getBonificacion() {
    return bonificacion;
}

public void setBonificacion() {
    this.bonificacion=bonificacion;
}

public int getEdad() {
    return edad;
}

public void setEdad(int edad) {
    this.edad = edad;
}

```

He creado un método de tipo booleano llamado `esTitularValido`, para comprobar si el titular reúne las condiciones para tener una cuenta joven, es decir mayor de edad pero menor de 25 años.

```

//metodo esTitularValido para saber si el titular cumple o no la condicion para tener cuenta joven

boolean r =esTitularValido (edad);

public boolean esTitularValido(int edad) {
    boolean resultado;
    if ( edad>=18 && edad<25) {
        resultado=true;
    }
    else {
        resultado = false;
    }
    return resultado;
}

```

El método retirar lo hemos pasado de la clase Cuenta a Cuentajoven ya que tiene una características especiales (tiene que ser titular válido) que no tienen que cumplirse en otro tipo de cuenta. He incorporado un if para que solo nos deje retirar dinero si el titular es válido

```
/*metodo retirar, lo he cambiado de la clase padre a la hija, ya que la condicion de titula valido  
* solo es para la cuenta joven y no para todas las cuentas en general*/
```

```
public void retirar (double cantidad) {  
  
    if (r== true) {  
        this.cantidad=this.cantidad-cantidad;  
    }  
  
    else {  
        this.cantidad=this.cantidad;  
    }  
  
}
```

También he movido del método mostrar datos de la clase padre a la clase hija ya que en este método se muestran características de este tipo de cuentas, como indicar que se trata de una cuenta joven y por ello se tiene una bonificación.

```
//metodo mostrar datos
```

```
public void mostrar () {  
    System.out.println("titular "+ titular + " tiene en su cuenta: " + cantidad + " euros"+  
    " Se trata de una Cuenta Joven con una bonificacion del: " + bonificacion + "%" );  
}
```

Por último he creado una clase principal, en ella probaremos nuestro programa, para ellos tenemos dos objetos de la clase Cuentajoven (cuentajoven1 y cuentajoven2); he invocados a los métodos setters para darle parámetros a sus atributos (titular, cantidad y edad) y posteriormente a los métodos ingresar y retirar para modificar las cantidades. En cuentajoven2 he puesto una edad menor a 18 años de forma que el titular no es válido y vemos como la operación de retirar dinero no se ejecuta

Finalmente he invocado al método mostrar para que estos datos se nos muestren por pantalla.

```
public class Principal {  
    public static void main(String[] args) {  
        //creamos dos objetos de la clase cuenta joven  
  
        Cuentajoven cuentajoven1= new Cuentajoven();  
        Cuentajoven cuentajoven2=new Cuentajoven();  
  
        //damos los valores a cada cuenta  
  
        cuentajoven1.setTitular("Jose Ruiz Lopez");  
        cuentajoven1.setCantidad(700);  
        cuentajoven1.setEdad(21);  
  
        cuentajoven2.setTitular("Luisa Perez Dominguez");  
        cuentajoven2.setCantidad(500);  
        cuentajoven2.setEdad(17);  
  
        //vamos a hacer operaciones de ingreso y retirada de dinero  
  
        cuentajoven1.ingresar(200);  
        cuentajoven1.retirar(100);  
  
        cuentajoven2.ingresar(100);  
        cuentajoven2.retirar(300);  
  
        //vamos a mostrar nuestros valores  
  
        cuentajoven1.mostrar();  
        cuentajoven2.mostrar();  
    }  
}
```