Ejercicios adicionales UT3 – *Clases y estructuras condicionales*

Ejercicio AD01.

Una biblioteca quiere modelar los libros que guarda.

- De cada libro se quiere saber:
  - autor
  - título
  - número de páginas
  - número de referencia (ISBN)
  - veces que se ha prestado
- Todos los libros comienzan sin haber sido prestados nunca.
- Cada vez que se presta un libro, se incrementa el valor de las veces que se ha prestado.
- Se desea mostrar la información de un libro con el siguiente formato:

Titulo: XXXXXXXXXX
Autor: XXXXXXXXXX
Nº páginas : XXX
Nº de referencia: XXX
Nº veces prestado: XXX

1. Abre el paquete EjercicioAD01
2. Rellena la **clase Libro** con los atributos y métodos necesarios
3. Implementa el método **main de TestLibro** para obtener la salida que se ve en la imagen cumpliendo con:

- Utiliza un único objeto Libro
- No utilices ningún System.out excepto para mostrar las líneas de asteriscos
  System.*out*.println("*".repeat(60));

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Progr
************************************************************
Título: El imperio final
Autor: Brandon Sanderson
N° de páginas: 688
N° de referencia:
N° de veces prestado: 0
************************************************************
Título: El imperio final
Autor: Brandon Sanderson
N° de páginas: 688
N° de referencia: 9788498726138
N° de veces prestado: 0
************************************************************
Título: El imperio final
Autor: Brandon Sanderson
N° de páginas: 688
N° de referencia: 9788498726138
N° de veces prestado: 2
************************************************************
```

Ejercicio 2.

Queremos modelar el comportamiento de una hucha que guarda una determinada cantidad de dinero.

- La hucha siempre comienza vacía.
- Podemos meter más dinero en la hucha y podemos sacar dinero de ella.
- No se pueden introducir en la hucha cantidades negativas.
- No se puede sacar más dinero del que hay en la hucha.
- Se desea mostrar la información sobre la hucha con el siguiente formato:

```
******** HUCHA **************
Hay XXX euros
********************************
```

1. Abre el paquete EjercicioAD02
2. Rellena la **clase Hucha** con los atributos y métodos necesarios.
   - Utiliza los operadores += y -=
3. Implementa el método **main de TestHucha** para obtener la salida que se ve en la imagen cumpliendo con:
   - Utiliza un único objeto Hucha
   - No utilices ningún System.out en Test Libro excepto para mostrar las líneas de guiones
     System.*out*.println("-".repeat(60));

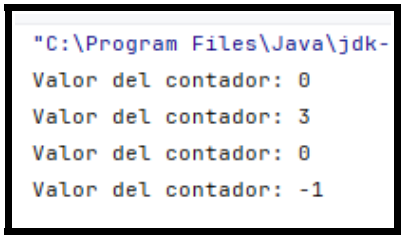
```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Pro
------------------------------------------------------------
******** HUCHA **************
Hay 20 euros
****************************
------------------------------------------------------------
No puede introducir una cantidad negativa en la hucha.
******** HUCHA **************
Hay 20 euros
****************************
------------------------------------------------------------
No puede sacar más dinero del que hay en la hucha.
******** HUCHA **************
Hay 20 euros
****************************
------------------------------------------------------------
******** HUCHA **************
Hay 5 euros
****************************
------------------------------------------------------------
```

Ejercicio 3.

Queremos modelar el comportamiento de un contador de personas que acceden a un museo cada día.

- El contador siempre comienza en 0
- Hay un botón que permite añadir una persona más al contador
- Hay un botón que permite eliminar una persona del contador, por si te has equivocado al pulsar
- Hay un botón que permite poner el contador a 0
- Se desea mostrar la información sobre el contador con el siguiente formato:
  Valor del contador: XXX

1. Abre el paquete EjercicioAD03
2. Rellena la **clase Contador** con los atributos y métodos necesarios.
   - Utiliza los operadores ++ y --
3. Implementa el método **main de TestContador** para obtener la salida que se ve en la imagen cumpliendo con:
   - Utiliza un único objeto Contador
   - No utilices ningún System.out

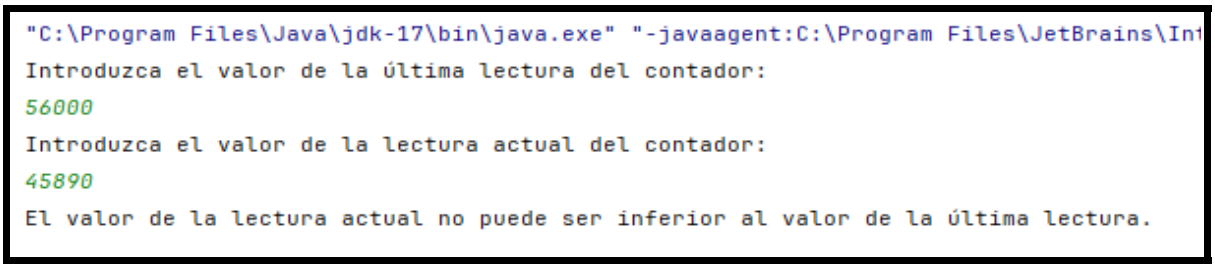


Ejercicio 4.

Queremos modelar el comportamiento de una factura de la luz.

- Una factura guarda:
  - la última lectura del contador en KW.
  - La lectura actual del contador en KW.
- En la factura se utilizan dos datos fijos (**constantes)**:
  - PRECIO_KW con un valor de 8.6 cents. (es el precio del KW consumido – suponemos que todos los KW tienen el mismo precio)
  - IVA es el 16% que hay que aplicar a la factura
- Se necesita saber el consumo actual de la factura.
- Se necesita saber el importe a pagar por lo que se ha consumido incluyendo el IVA

1. Abre el paquete EjercicioAD04
2. Rellena la **clase FacturaLuz** con los atributos y métodos necesarios.
   - Utiliza las constantes que ya están definidas en la clase
   - No utilices ningún System.out
3. Implementa el método **main de TestFacturaLu**z para obtener la salida que se ve en las **imágenes** cumpliendo con:
   - Utiliza un Scanner para solicitar los datos al usuario
   - Utiliza un objeto de tipo FacturaLuz
   - No utilices ninguna expresión matemática
   - No permitas que el usuario introduzca una lectura actual inferior a la ultima lectura



```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Progr
Introduzca el valor de la última lectura del contador:
14567
Introduzca el valor de la lectura actual del contador:
23387
El consumo de la factura 1 es: 8820
El importe total de la factura 1 es: 88704.50399999999
```

Ejercicio 5.

Queremos modelar el comportamiento de una máquina registradora que indica cómo devolver los cambios de la forma más óptima.

de Crea en el paquete EjercicioAD05 una nueva clase Dinero

- La máquina guarda la cantidad de dinero en euros que se quiere devolver.
- La máquina muestra la descomposición en billetes y monedas con el siguiente formato:

  XX € son
  Billetes de 50 = X
  Billetes de 10 = X
  Billetes de 5 = X
  Monedas de 2 = X
  Monedas de 1 = X

1. Abre el paquete EjercicioAD05
2. Rellena la **clase Dinero** con los atributos y métodos necesarios.
   - Utiliza las constantes que ya están definidas en la clase
3. Implementa el método **main de TestDinero** para obtener la salida que se ve en las imágenes cumpliendo con:
   - Utiliza un Scanner para solicitar los datos al usuario
   - Utiliza un objeto de tipo Dinero
   - No utilices ninguna expresión matemática
   - No permitas que el usuario introduzca valores negativos ni superiores a 100€.

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\P
Introduce una cantidad de € no superior a 100€:
-23
No ha introducido una cantidad positiva no superior a 100€
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-jav
Introduce una cantidad de € no superior a 100€:
78
78 € son
Billetes de 50 = 1
Billetes de 10 = 2
Billetes de 5 = 1
Monedas de 2 = 1
Monedas de 1 = 1
```