

## Ejercicios adicionales UT3 ( I ) - Entendiendo la definición de una clase

### Ejercicio 1.

- Abre el proyecto Libro. Analiza el código que contiene.
- Añade dos métodos accesoros, `getAutor()` y `getTitulo()`. Testea la clase creando varias instancias y llamando a estos métodos.
- Añade dos métodos `printAutor()` y `printTitulo()` que escriben el autor y el título, respectivamente, en la pantalla.
- Añade un atributo adicional `numeroPaginas` de tipo entero. Su valor inicial se pasará como parámetro al constructor junto con el autor y el título.
- Incluye el accesor correspondiente al n° de páginas.
- Añade el método `printDetalles()` que escribe la información completa del libro en pantalla:

```
Titulo:  XXXXXXXXXXXX
Autor:   XXXXXXXXXXXX
Número páginas :  XXX
```

- Añade un nuevo atributo `numeroReferencia` de tipo String. Se inicializa como "" (cadena vacía) dentro del constructor, no se pasa como parámetro.
- Define un mutador `setNumeroReferencia()` para este nuevo atributo.
- Incluye el correspondiente accesor para el n° de referencia.
- Modifica el método `printDetalles()` para que aparezca el n° de referencia.
- Añade el atributo `vecesPrestado` que indicará la cantidad de veces que el libro ha sido prestado. Inicializa a 0 el atributo en el constructor.
- Añade el mutador `prestar()`. No tiene parámetros, solo incrementa el atributo `vecesPrestado` (para el incremento utiliza el operador ++).
- Incluye para este último atributo el correspondiente accesor.
- Modifica el método `printDetalles()` para que aparezca el valor de este atributo.

### Ejercicio 2.

- Crea en C:\Programacion\UT3 un nuevo proyecto Hucha y añade una clase con el mismo nombre. La clase modela el comportamiento de una hucha que guarda una determinada cantidad de dinero. Podemos meter más dinero en la hucha y podemos sacar dinero de ella.
- Define el atributo de la clase que representa el dinero (supondremos que son euros)
- Define un constructor sin parámetros que inicializa cada objeto Hucha con un dinero inicial o
- Añade un accesor que permita consultar cuánto dinero hay
- Añade los mutadores que permiten meter y sacar dinero de la hucha en una determinada cantidad (de momento no haremos comprobaciones acerca de si es posible o no sacar lo que se pide)
- Incluye el método `printDetalles()` para visualizar la cantidad de dinero que hay en la hucha.
- Prueba la clase creando varias instancias y llamando a los métodos para verificar que todo funciona bien

### Ejercicio 3.

Contador
- valor: int
+ Contador()
+ getValor():int
+ incrementar():void
+ decrementar():void
+ reset():void
+ printContador():void

- Crea en C:\Programacion\UT3 un nuevo proyecto Contador
- Añade al proyecto la clase que muestra el siguiente diagrama de clases UML y complétala
- Indica qué métodos son los accesoros y los mutadores.
- Testea la clase desde BlueJ

#### Ejercicio 4.

- Abre el proyecto Factura Luz y haz las modificaciones que se piden a la clase FacturaLuz. Los objetos de esta clase guardan información acerca del consumo de luz en una factura de electricidad y obtienen el importe total facturado.
- Añade a la clase las siguientes constantes:
  - VALOR\_KW con un valor de 8.6 cents. (es el valor del KW consumido – suponemos que todos los KW tienen el mismo precio)
  - IVA es el 16% que hay que aplicar a la factura
- Modifica el constructor añadiendo los parámetros adecuados para inicializar los atributos que incluye la clase adecuadamente
- Añade el método obtenerConsumo() que devuelve el nº de Kw consumidos
- Incluye el método obtenerImporteTotal() que devuelve el importe a pagar por lo que se ha consumido incluyendo el IVA.
- Prueba la clase.

#### Ejercicio 5.

Convertor
- grados: double
+ Convertor() + getGrados():double + setGrados(queGrados:double):void + aCentigrados():double + aFahrenheit():double

- Crea en C:\Programacion\UT3 un nuevo proyecto Convertor
- Define la clase que muestra el diagrama. La clase Convertor modela un convertor de temperatura, de grados centígrados a grados Fahrenheit y viceversa.
- La fórmula de conversión entre temperaturas es la siguiente:

$$^{\circ}F = \frac{9}{5}^{\circ}C + 32$$

#### Ejercicio 6.

```
BlueJ: Ventana de Terminal - ...
Opciones
32 € son
Billetes de 10 = 3
Billetes de 5 = 0
Monedas de 2 = 1
Monedas de 1 = 0
Can only enter input while your
```

- Crea en C:\Programacion\UT3 un nuevo proyecto Dinero
- Define una clase Dinero con un atributo entero, euros. Asumiremos que el valor de este atributo siempre será un valor positivo y menor que 100
- Define dentro de la clase cinco constantes que lleven asociado los valores 10, 5, 2 y 1
- Incluye un constructor con parámetros y un accesor y mutador para el valor que representa el dinero
- Añade un método printDescomposicionMonedas() que muestre en pantalla la descomposición del dinero que almacenan los objetos de la clase en el mínimo nº de billetes de 10, billetes de 5, monedas de 2€ y monedas de 1€. (La figura muestra un ejemplo de los resultados a obtener)

```
BlueJ: Ventana de Terminal - ...
Opciones
78 € son
Billetes de 10 = 7
Billetes de 5 = 1
Monedas de 2 = 1
Monedas de 1 = 1
Can only enter input while your
```

- Testea la clase