

Proyecto 1 UT4 (para hacer en casa y entregar en GitHub)

Objetivos

Saber:

- definir objetos que contienen otros objetos
- definir constructores sobrecargados
- crear objetos a través del operador `new`
- definir y utilizar parámetros y valores de retorno que sean tipos objeto
- realizar llamadas externas (paso de mensajes entre objetos)
- realizar llamadas internas
- utilizar correctamente `this` y `null`
- formatear valores con `String.format()`
- generar valores aleatorios utilizando la clase `Random`

Repasar:

- `toString()`
- estructuras condicionales: `if` / `switch`

Antes de empezar

- Este ejercicio es para realizar de forma **individual** en casa.
- El proyecto de partida está en <https://github.com/montsemsanz/ENTRE-01-UT4-EnvioPaquete>. Deberás hacer un *fork* a tu cuenta y clonarlo en tu PC desde BlueJ tal y como se explicó en clase
- Una vez completado desde BlueJ haz un *push* del último *commit* a GitHub
- No olvides **entregar vía Moodle** el texto de la actividad “**Terminado proyecto 1 UT4 EnvioPaquete**” y pulsar *Enviar para calificar*
- Se valorará en la corrección que el programa esté probado (compila y ejecuta bien) y que esté claramente escrito y organizado (se respetan las reglas de estilo del lenguaje Java, nombres descriptivos, comentarios adecuados, código no duplicado, ...)

- La fecha tope de entrega es el **Jueves 4 de Noviembre** hasta las **23,30h**.

- Se anulará automáticamente la corrección del ejercicio y se **evaluará con un 0** si se detecta que ha sido copiado o dejado copiar a algún compañero/a
- Se penalizará si no se siguen las normas de entrega del ejercicio
 - x no se ha hecho un *fork* / no se sube vía *commit*
 - x hay algún *commit* posterior a esta fecha de entrega
 - x no se ha enviado el texto de la actividad vía Moodle
- El profesorado podrá convocar al alumno/a para defender oralmente el proyecto

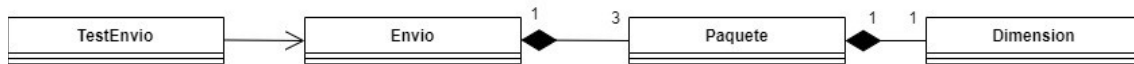
Especificaciones

En este proyecto vamos a completar una serie de clases que **modelan un envío de varios paquetes de determinada dimensión y peso**.

Haz el *fork* del proyecto **ENTRE-01-UT4-EnvioPaquete** desde <https://github.com/montsemsanz> a tu cuenta GitHub y desde BlueJ clona el proyecto a tu PC.

Abre el proyecto BlueJ. Tienes que completar las clases `Dimension`, `Paquete`, `Envio` y `TestEnvio`.

No olvides escribir tu nombre después de la etiqueta `@author` en todas ellas



clase Dimension

Esta clase guarda una dimensión (alto, ancho, largo en *cm*).

Completa:

- método **obtenerCopia()** - devuelve un nuevo objeto que es una copia exacta del objeto actual
- método **toString()** - devuelve una cadena formateada con los datos de la dimensión. Si se ejecuta el método **print()** proporcionado después de crear un objeto con valores 40, 30, 50 la salida debe ser:

```

Alto:    40,00(cm)
Ancho:   30,00(cm)
Largo:   50,00(cm)
  
```

Columna a 20 espacios
alineada a la derecha

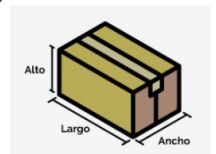
Columna de n°s a 10 espacios alineada a
la derecha (2 para decimales)

clase Paquete

Un objeto de esta clase guarda los datos de un paquete. Todo paquete tiene una dimensión y un peso real (en *Kg*).

Completa los tres constructores sobrecargados:

- *constructor1* - crea el paquete con valores 40 , 30, 50 para alto, ancho, largo (en *cm*) y peso 8 *Kg*
- *constructor2* - crea el paquete con los valores de alto, ancho, largo indicados por los parámetros y un peso aleatorio entre 2 y 8 (inclusive). Crea dentro de este constructor el generador
- *constructor3* - crea el paquete con la dimensión y peso indicados
- método **calcularVolumen()** - calcula y devuelve el volumen del paquete (en *cm³*). El volumen es *alto x ancho x largo*



- método **calcularPesoVolumetrico()** - calcula y devuelve el peso volumétrico del paquete (en *Kg*). *Peso volumétrico = volumen del paquete / 5000*
- método **calcularPesoFacturable()** - calcula y devuelve el peso facturable de un paquete. El peso facturable es el valor mayor entre su peso real y su peso volumétrico (es el **valor que se utiliza para calcular el coste de envío del paquete**)
- método **obtenerCopia()** - devuelve una copia exacta del objeto actual. Al hacer la copia se obtienen copias también de los objetos que contenga
- método **toString()** - devuelve una cadena formateada con los datos del paquete.

La salida mostrada si se ejecuta el método **print()** proporcionado después de crear un objeto **Paquete** es:

Con el constructor 1

```

Descripción del paquete
Alto:    40,00(cm)
Ancho:   30,00(cm)
Largo:   50,00(cm)
Peso real:  8,00(Kg)
Volumen:  60000,00(cm3)
Peso volumétrico:  12,00(Kg)
  
```

Columna a 20 espacios
alineada a la derecha

Con el constructor 2

```

Descripción del paquete
Alto:    40,00(cm)
Ancho:   60,00(cm)
Largo:   40,00(cm)
Peso real:  4,00(Kg)
Volumen:  96000,00(cm3)
Peso volumétrico:  19,20(Kg)
  
```

Columna de n°s a 10 espacios alineada a
la derecha (2 para decimales)

Con el constructor 3

```

Descripción del paquete
Alto:    50,00(cm)
Ancho:   50,00(cm)
Largo:   70,00(cm)
Peso real:  13,00(Kg)
Volumen:  175000,00(cm3)
Peso volumétrico:  35,00(Kg)
  
```

clase Envio

La clase describe un envío de varios paquetes. Un envío contiene tres paquetes como máximo. No se necesitan más atributos que los indicados. El constructor de esta clase crea el envío con todos los paquetes inicialmente a *null*.

Completa:

- método **getNumeroPaquetes()** - devuelve el nº de paquetes que incluye el envío (dependerá de cuántos paquetes estén a *null*)
- método **envioCompleto()** - devuelve *true* si el envío está completo, es decir, tiene exactamente 3 paquetes, *false* en otro caso (hacer sin *if*)
- método **addPaquete(Paquete paquete)** - añade un nuevo paquete al envío. Si el envío está completo se muestra el mensaje "*No se admiten más paquetes en el envío*". Si no está completo se añade el paquete teniendo en cuenta si se añade como primero, segundo o tercero (no han de quedar huecos)
- método **calcularCosteTotalEnvio()** - calcula y devuelve el coste total del envío. Para hacer este cálculo:
 - x se obtiene el peso facturable de cada paquete
 - x se suman los pesos facturables de todos los paquetes
 - x se calcula el coste en euros según el precio del Kg (cada Kg no completo se cobra entero, 5.8 Kg. se cobran como 6, 5.3 Kg. se cobran como 6)

- método **toString()** - devuelve la representación textual del envío con el formato exacto mostrado en la figura

```
Nº de paquetes: 3
Descripción del paquete
    Alto:      40,00(cm)
    Ancho:     30,00(cm)
    Largo:     50,00(cm)
    Peso real:  8,00(Kg)
    Volumen:   60000,00(cm3)
    Peso volumétrico: 12,00(Kg)

Descripción del paquete
    Alto:      40,00(cm)
    Ancho:     60,00(cm)
    Largo:     40,00(cm)
    Peso real:  8,00(Kg)
    Volumen:   96000,00(cm3)
    Peso volumétrico: 19,20(Kg)

Descripción del paquete
    Alto:      50,00(cm)
    Ancho:     50,00(cm)
    Largo:     70,00(cm)
    Peso real:  13,00(Kg)
    Volumen:   175000,00(cm3)
    Peso volumétrico: 35,00(Kg)

Coste total envío: 147,40€
```

clase TestEnvio

Añade al proyecto la clase `TestEnvio` que **incluya solo el método `main()`**. No definas ningún atributo. La utilizaremos para probar el resto de clases.

Incluye un breve comentario *javadoc* y la etiqueta *@author*.

- método **main()** - dentro de este método:
 - x crea *paquete1* con el primer constructor
 - x crea *paquete2* de dimensiones 40, 60, 40 con el segundo constructor
 - x crea un envío y añade *paquete1* y *paquete2*
 - x muestra en pantalla el envío
 - x crea *paquete3* de dimensiones 50, 50, 70 con el tercer constructor
 - x añade el paquete al envío y muestra este último de nuevo en pantalla
 - x obtén una copia de *paquete2* y añádela al envío

Los resultados a obtener son los de la siguiente figura:

Nº de paquetes: 2		Nº de paquetes: 3	
Descripción del paquete		Descripción del paquete	
Alto:	40,00(cm)	Alto:	40,00(cm)
Ancho:	30,00(cm)	Ancho:	30,00(cm)
Largo:	50,00(cm)	Largo:	50,00(cm)
Peso real:	8,00(Kg)	Peso real:	8,00(Kg)
Volumen:	60000,00(cm3)	Volumen:	60000,00(cm3)
Peso volumétrico:	12,00(Kg)	Peso volumétrico:	12,00(Kg)
Descripción del paquete		Descripción del paquete	
Alto:	40,00(cm)	Alto:	40,00(cm)
Ancho:	60,00(cm)	Ancho:	60,00(cm)
Largo:	40,00(cm)	Largo:	40,00(cm)
Peso real:	5,00(Kg)	Peso real:	5,00(Kg)
Volumen:	96000,00(cm3)	Volumen:	96000,00(cm3)
Peso volumétrico:	19,20(Kg)	Peso volumétrico:	19,20(Kg)
Coste total envío: 70,40€		Descripción del paquete	
-----		Alto:	50,00(cm)
		Ancho:	50,00(cm)
		Largo:	70,00(cm)
		Peso real:	13,00(Kg)
		Volumen:	175000,00(cm3)
		Peso volumétrico:	35,00(Kg)
		Coste total envío: 147,40€	

		No se admiten más envíos	

Rúbrica evaluación	
clase Dimension	6
clase Paquete	20
clase Envío	60
clase TestEnvío	14
	100
Penalización (no compila, no probado)	-0,5 (sobre 10)