

## 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()`

#### 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/progdaw1/ENTRE-01-UT4-HojaCalculo>. 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 HojaCalculo”** 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, código no duplicado, ...)

- La fecha tope de entrega es el **Lunes 2 de Noviembre** hasta las **23,59h**.

- 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
  - ➔ 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

### Especificaciones

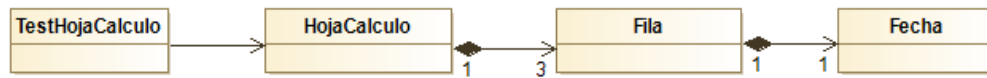
En este proyecto vamos a completar una serie de clases que modelan una sencilla hoja de cálculo que contiene un máximo de tres filas. Cada fila guarda información sobre un apunte contable de una empresa.

	FECHA	INGRESOS	GASTOS
Fila1	4/10/2020	25,50	132,00
Fila2	5/10/2020	300,00	350,00
Fila3	1/1/2020	0,00	0,00
		325,50	482,00

Haz el *fork* del proyecto **ENTRE-01-UT4-HojaCalculo** desde <https://github.com/progdaw1> a tu cuenta GitHub y desde BlueJ clona el proyecto a tu PC.

Abre el proyecto BlueJ. Tienes que completar las clases Fecha, Fila, HojaCalculo y Test HojaCalculo.

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



### clase Fecha

Esta clase representa la fecha del apunte contable.

Completa:

- método **obtenerCopia()** - devuelve un nuevo objeto que es una copia exacta del objeto actual

### clase Fila

Esta representa a una fila de la hoja de cálculo. Una fila guarda además de su identificador (su nombre), la fecha en que se “anota” el apunte y los ingresos y gastos en esa fecha de la empresa.

Completa:

- constructor1 - recibe como parámetro el identificador de la fila y crea e inicializa la fila con ingresos y gastos a 0 y como fecha 1/1/2020
- constructor2 - recibe como parámetros todos los datos que se necesitan para crear e inicializar la fila
- método **duplicar()** - crea y devuelve una nueva fila que es una copia idéntica de la actual. La fecha que incluye la fila duplicada también es una copia
- método **toString()** - devuelve una cadena formateada con los datos de la fila y una nueva columna calculada que indica el beneficio en ese fecha

Fila1	4/10/2020	25,50€	132,00€	-106,50€ **
8 espacios alineado a la derecha	15 posiciones alineado a la derecha	15 posiciones de las cuales 2 decimales alineado a la derecha	15 posiciones de las cuales 2 decimales alineado a la derecha	15 posiciones de las cuales 2 decimales alineado a la derecha. Observa que si el beneficio es negativo se añaden ** al final
Fila1	7/10/2020	260,00€	125,00€	135,00€

## clase HojaCalculo

La clase describe una hoja de cálculo. Una hoja de cálculo tiene tres atributos: su nombre y las tres filas de que consta como máximo. No se necesitan más atributos. El constructor de esta clase crea la hoja con su nombre pero las filas están inicialmente “vacías” (a *null*).

Completa:

- método **getNumeroFilas()** - devuelve el nº de filas de la hoja (las que están creadas) Dependerá de cuántas filas estén a *null*.
- método **hojaCompleta()** - devuelve *true* si la hoja está completa, es decir, tiene exactamente 3 filas creadas.
- método **addFila(Fila fila)** - Se añade una nueva fila que se pasa como parámetro a la hoja. Si la hoja está completa se muestra el mensaje "*FilaX no se puede añadir en HOJAX*". Si no está completa se añade la fila a la hoja teniendo en cuenta si se añade como primera, segunda o tercera fila (no han de quedar huecos)
- método **addFila(String id, Fecha fecha, double ingresos, double gastos)** - dados los datos necesarios para crear una fila, se crea la fila y se añade a la hoja. (evita repetir código)
- método **getTotalIngresos()** - calcula y devuelve el total de ingresos entre todas las filas que incluye la hoja
- método **getTotalGastos()** - calcula y devuelve el total de gastos entre todas las filas que incluye la hoja
- método **getBeneficio()** - calcula y devuelve el total de beneficio entre todas las filas que incluye la hoja
- método **toString()** - Representación textual de la hoja con el formato exacto que se indica en la figura siguiente

HOJA1				
	FECHA	INGRESOS	GASTOS	BENEFICIO
Fila1	4/10/2020	25,50€	132,00€	-106,50€ **
Fila2	5/10/2020	300,00€	350,00€	-50,00€ **
Fila3	1/1/2020	0,00€	0,00€	0,00€
		325,50€	482,00€	-156,50€

- método **duplicarHoja()** - devuelve un duplicado de la hoja actual. El nombre de la hoja duplicada será "*Duplicada HojaX*". Al duplicar la hoja se duplicarán también las filas que contenga

## clase TestHojaCalculo

En esta clase incluiremos el código para probar el resto. Omitiremos atributos y constructor y el código para probar lo haremos dentro de los métodos.

- el método **test1()** - define y crea todos los objetos necesarios para obtener en pantalla la hoja de cálculo indicada en la *figura 1* incluyendo la hoja duplicada. La fila4 que no se podrá guardar puedes crearla con el constructor por defecto de fila (el que no tiene parámetros)

Fila4 no se puede añadir en HOJA1

HOJA1				
	FECHA	INGRESOS	GASTOS	BENEFICIO
Fila1	4/10/2020	25,50€	132,00€	-106,50€ **
Fila2	5/10/2020	300,00€	350,00€	-50,00€ **
Fila3	1/1/2020	0,00€	0,00€	0,00€
		325,50€	482,00€	-156,50€
Duplicada HOJA1				
	FECHA	INGRESOS	GASTOS	BENEFICIO
Fila1	4/10/2020	25,50€	132,00€	-106,50€ **
Fila2	5/10/2020	300,00€	350,00€	-50,00€ **
Fila3	1/1/2020	0,00€	0,00€	0,00€
		325,50€	482,00€	-156,50€

figura 1)

- el método **test2()** - define y crea todos los objetos necesarios para obtener en pantalla la hoja de cálculo indicada en la figura 2) incluyendo la hoja duplicada

HOJA2				
	FECHA	INGRESOS	GASTOS	BENEFICIO
Fila1	7/10/2020	260,00€	125,00€	135,00€
Fila2	8/10/2020	125,00€	245,00€	-120,00€ **
		385,00€	370,00€	15,00€
Duplicada HOJA2				
	FECHA	INGRESOS	GASTOS	BENEFICIO
Fila1	7/10/2020	260,00€	125,00€	135,00€
Fila2	8/10/2020	125,00€	245,00€	-120,00€ **
		385,00€	370,00€	15,00€

figura 2)

- el método **test3()** - define y crea todos los objetos necesarios para obtener en pantalla la hoja de cálculo indicada en la figura 3) incluyendo la hoja duplicada

HOJA3				
	FECHA	INGRESOS	GASTOS	BENEFICIO
Fila1	8/10/2020	670,00€	234,00€	436,00€
		670,00€	234,00€	436,00€
Duplicada HOJA3				
	FECHA	INGRESOS	GASTOS	BENEFICIO
Fila1	8/10/2020	670,00€	234,00€	436,00€
		670,00€	234,00€	436,00€

figura 3)

Rúbrica evaluación	
clase Fecha	5
clase Fila	20
clase HojaCalculo	115
clase TestHojaCalculo	60
	200
Penalización (no compila, no probado)	-0,5 (sobre 10)