

EXAMEN TAP

curso 2016/17 (27/1/17)

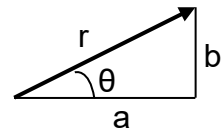
FCT/ZTF – UPV/EHU

Los tres ejercicios se resolverán dentro de un mismo proyecto, cada uno en un paquete denominado: edu.upv.fct.examenTap1617.<su DNI>.ejN, donde N será el número del ejercicio.

1- ESCRITURA DE UNA CLASE (3.75 puntos -relación de pesos 3:2:3-)

Este ejercicio supone aplicar conocimientos específicos, tomar algunas decisiones y respetar unas cuantas “obligaciones” y “recomendaciones” que ha de conocer. Se valorará puntualmente y por separado cada uno de estos aspectos. **Hacer que la clase sea “robusta” (gestione bien posibles errores aritméticos, infinitos, NaNs, cuestiones de precisión, etc.) es algo que puede llevar horas. No se trata de eso, basta con plantear bien toda la estructura de la clase y el código genérico de cada método conforme a las operaciones aritméticas mostradas.**

Defina la clase “Complex” para representar números complejos, de modo que:



- tenga un constructor privado, y dos métodos para instanciar objetos:

newFromCartesian(double real, double imaginary)

newFromPolar(double magnitude, double argument)

[$z = a+bi = r e^{i\theta}$, a:parte real, b:parte imaginaria, r:magnitud (módulo), θ : argumento (ángulo)]

- los números complejos no deben ser alterables una vez construidos, y los campos internos que los representen no deberán ser accesibles desde ninguna otra clase.

- implemente los interfaces que considere adecuados a la clase. En particular implemente el interfaz **Comparable** tomando como criterio de relación de orden entre dos complejos el que supone comparar primero sus magnitudes, y en caso de igualdad comparar sus argumentos.

- disponga de métodos públicos (estáticos o no, según sea adecuado en cada caso) para obtener ...

· la parte real	real()	a	$r \cos(\theta)$
· la parte imaginaria	imaginary()	b	$r \sin(\theta)$
· la magnitud	magnitude()	$\sqrt{a^2+b^2}$	r
· el argumento	argument()	$\arctg^{(*)}(b/a)$	θ
· el conjugado	conjugate()	$a-bi$	$r_{-\theta}$
· el inverso	inverse()	$[a-bi]/(a^2+b^2)$	$1/r_{-\theta}$
· la suma de dos complejos	add(Complex c1, Complex c2)	$(a_1+a_2)+(b_1+b_2)i$???
· el producto por un real	multiply(double d)	$d a + d b i$	$d r_{\theta}$
· el producto de dos complejos	multiply(Complex c1, Complex c2)	$[(a_1 a_2 - b_1 b_2) + (a_1 b_2 + a_2 b_1) i]$	$(r_1 r_2)_{\theta_1+\theta_2}$
· la división de dos complejos	divide(Complex c1, Complex c2)	$[(a_1 a_2 + b_1 b_2) + (b_1 a_2 - b_2 a_1) i] / (a^2 + b^2)$	$(r_1 / r_2)_{\theta_1 - \theta_2}$
· la potencia d-ésima de un complejo	pow(Complex c, double d)	???	$r_{d\theta}^d$
· la raíz cuadrada de un complejo	sqrt(Complex c)	???	$r_{\theta/2}^{1/2}$

(*)no es trivial, pero en Math existe un método que hace exactamente esto.

- reescriba los métodos de **Object** que considere oportunos. En concreto reescriba el “toString()” para que presente el complejo con el siguiente formato: “(parte_real +/- parte_imaginaria i) (modulo; argumento)”

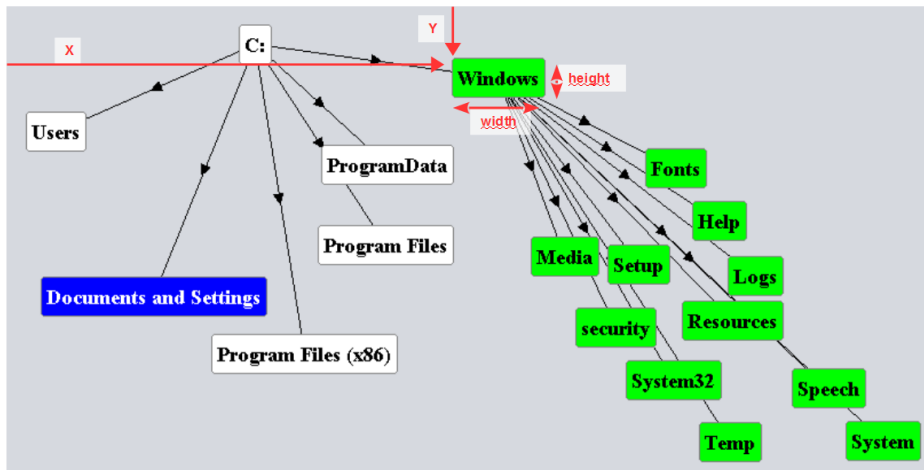
p.ej.: (32,0000-32,0000 i) (45,2548; -0,7854)

[link para “copy-paste” en la sección “Evaluación” de la web del curso]

2- ESCRITURA DE UN ALGORITMO (2.50 puntos -relación de pesos 3:2:3-)

Una aplicación muestra en un interfaz gráfico los directorios del ordenador como se ve en la figura que se encuentra más adelante. La estructura de datos que contiene la información es un conjunto (Set) de objetos “Folder”, que para lo que nos interesa tiene estos campos

```
class Folder{
    int x,y;           //posición del vértice superior izquierdo del rectángulo.
    int width,height; //ancho y alto del rectángulo
    String dir;       //el path del directorio p.ej.: "c:/Windows/Media"
    //aquí el resto de la definición de la clase
}
```



Para hacer posible el desplazamiento de toda una rama que “cuelga” de un determinado directorio arrastrándolo con el ratón, escriba el cuerpo de la siguiente rutina:

```
void desplaza(Set<Folder> folders, int x, int y, int desplazaX, int desplazaY){...}
```

teniendo en cuenta que

- el primer parámetro es el conjunto de todas las carpetas representadas en el Interfaz,
- los parámetros “x” e “y” indican la posición donde se “pincha” con el ratón y por tanto servirá para saber qué carpeta se selecciona
- se deberán incrementar las posiciones de todas las carpetas del subárbol de la “pinchada” (ella incluida) en la cantidad “desplazaX” y “desplazaY”. (aclaración: se deben mover todas las carpetas para las que el valor de la variable “dir” comience con el valor de la variable “dir” de la carpeta “pinchada”)

3- PLANTEAMIENTO DE UNA SOLUCIÓN ESTRUCTURADA (3.75 puntos -relación de pesos 3:2:3-)

Escriba toda la estructura que considere necesaria (clases/interfaces/enumeraciones/, campos, métodos, etc. **excepto el cuerpo de los métodos**) para llevar a cabo una aplicación para un **Banco de Inversiones**, que permita la gestión de **carteras de acciones** de sus **clientes** con los siguientes requerimientos:

- Se dispondrá de DNI, nombre completo y número de teléfono de los clientes
- Un cliente podrá disponer de varias carteras de acciones.
- Las carteras podrán ser de dos tipos “Estándar” y “Black”, ambas contendrán una colección de **movimientos de compraventa** de acciones, y un saldo (efectivo disponible que disminuye con la compra de acciones y aumenta con su venta) . Las Estándar darán la posibilidad de estimar el coste tributario acumulado de todos los movimientos realizados desde el 1 de enero. Las Black dispondrán del DNI del gestor de la empresa offshore que se encargará de la operativa.
- Los movimientos de compraventa se caracterizarán por el nombre de la empresa (el “ticker”, p.ej.: “GOOG”), el indicador de si es una compra o una venta, la fecha de la operación, la cantidad de acciones y el precio unitario.

Los casos de uso típicos de la aplicación para el Banco de Inversiones serán:

- dar de alta/baja a un cliente.
- Abrir/cerrar una cuenta de un cliente (podrá abrirse con un saldo dado o a cero).
- Ingresar/retirar efectivo y consultar el saldo.
- indicar la compra/venta de acciones,
- consultar el resultado (ganancia o perdida) de las operaciones realizadas entre dos fechas.
- consultar el coste tributario acumulado desde el 1 de enero. (sólo carteras Estándar)
- obtener el listado de acciones en cartera o el número de acciones de una determinada empresa,
- << puede añadir lo que le parezca potencialmente util >>

(nota: para conseguir compilar sin errores, los métodos con valor de retorno necesitarán un “return”, que puede escribirse con una constante. p.ej.: return null, o return 0.0)