

TEMA 6. DESARROLLO DE CLASES. PRÁCTICA 4

Realizar los siguientes ejercicios utilizando el lenguaje Java y el IDE Apache NetBeans.

Entregar en la tarea correspondiente del curso del AulaVirtual, un fichero T06p04.zip con el contenido del proyecto completo.

Antes de comprimir la carpeta realizar un CLEAN del proyecto NetBeans.

1.- Construir un proyecto Java con *NetBeans* llamado “T06p04” con una clase principal “T06p04.java”.

Utilizar como directorio de trabajo “C:\Java” o “D:\Java”.

Eliminar comentarios innecesarios de las clases generadas.

2.- Agregar un paquete al proyecto llamado “modelo” (paralelo al ya creado “t06p04”).

3.- Agregar una clase llamada “Coordenada.java” dentro del paquete “modelo”.

- Esta clase dispone de dos atributos privados enteros *x* e *y*.
- Un atributo público estático entero *numCoordenadas*.

4.- Agregar a la clase “Coordenada” métodos *getters* para los atributos privados.

5.- Agregar a la clase “Coordenada” un método público “setCoordenada” que reciba dos enteros como parámetros y se los asigne a los atributos privados de la clase. Además debe incrementar el valor del atributo estático “numCoordenadas”.

6.- Agregar a la clase “Coordenada” un método público “getCoordenada” que devuelva un *String* con los valores de la coordenada entre paréntesis y separados por una coma, ej.: (1,2).

7.- Agregar a la clase “Coordenada” un método público para poner a 0 el valor del atributo “numCoordenadas” llamado “resetNumCoordenadas” (¿estático?).

8.- Probar la clase anterior desde el programa principal:

- Crear 3 coordenadas.
- Asignarles los valores (1,1), (2,2) y (3,3) respectivamente utilizando el método “setCoordenada”.
- Mostrar los valores de las coordenadas por pantalla utilizando el método “getCoordenada”.
- Mostrar el valor del atributo estático “numCoordenadas”.
- Resetear el valor del atributo estático utilizando el método construido anteriormente.
- Mostrar el valor del atributo estático “numCoordenadas”, otra vez.

9.- Agregar dos métodos al programa principal para obtener el *eje X* de una coordenada llamados “obtenerEjeX”:

- El primero recibe como parámetros una coordenada y un entero (paso por valor).
- El segundo recibe como parámetros una coordenada y un array de un entero (paso por referencia).
- Los métodos deben asignar el *eje X* de la coordenada del primer parámetro al segundo parámetro.

10.- Probar los métodos anteriores en el programa principal inicializando dos variables, un entero y un array de un entero y mostrando por pantalla sus valores después de llamar a los métodos. Observar cómo funciona el paso por valor y el paso por referencia.

11.- Agregar dos constructores a la clase “Coordenada”:

- Un primer constructor básico que inicialice los atributos *x* e *y* a 0.
- Un segundo constructor que reciba por parámetro los valores para los atributos *x* e *y*. Además debe incrementar el valor del atributo estático “numCoordenadas”.

12.- Crear 3 nuevas coordenadas utilizando el constructor sobrecargado anterior en vez del método “setCoordenada”, y mostrar sus valores por pantalla.

```
run:

Coordenada1: (1,1)
Coordenada2: (2,2)
Coordenada3: (3,3)
NumCoordenadas: 3
NumCoordenadas: 0

EjeX de la Coordenada1 (por valor): 0
EjeXX de la Coordenada2 (por referencia): 1

Coordenada4: (4,4)
Coordenada5: (5,5)
Coordenada6: (6,6)
NumCoordenadas: 3

BUILD SUCCESSFUL (total time: 0 seconds)
```