



## **Estructuras de Datos I**

**Fecha entrega:** 22/02/15

**Autores:** Pedro J. Ramos

Estela Muñoz Cordón

1. Entrega el siguiente código comentado con los resultados a la derecha de cada `System.out.println()` y contesta:

- a. ¿Cómo se comparan dos cadenas?

Las cadenas se comparan con `equals` para ver si contienen lo mismo o con `==` para ver si las referencias apuntan al mismo objeto

- b. ¿Para qué se utiliza el operador de comparación `==` entre objetos?

Para ver si las referencias apuntan al mismo objeto

- c. Una cadena puede crearse mediante `new` o directamente con su literal. ¿Se crean de igual manera? ¿Se aprovecha la memoria de alguna manera?

Se pueden crear de ambas maneras pero con su literal se aprovecha mejor la memoria.

2. Crea el siguiente código. Contesta a las siguientes preguntas:

- a. ¿Cuándo la comparación de dos referencias es `true`?

Cuando apuntan al mismo objeto

- b. ¿Siempre se inicializa una referencia a `null`?

No tiene por qué ser así, sólo si no se indica a que apunta

- c. ¿Por qué `vector1` y `vector2` han de ser `static`? ¿Cuál sería otra solución?

Para que tengan un valor inicial fijo, que se pueda cambiar su referencia o valor. Se puede poner `static` sin más.

- 
3. Indica las clases Wrappers o envoltorios correspondientes a los siguientes datos primitivos:

Boolean → Boolean

Byte → Byte

Char → Character

Float → Float

Int → Integer

Long → Long

Short → Short

4. ¿Qué es el autoboxing/unboxing? Explícalo mediante tu propio código en Java “autoboxing.java”

Es una conversión automática entre los tipos primitivos y sus wrappers (objeto que te lo contiene), de lo que se encarga el compilador, así mismo, permite realizar la extracción de forma automática de un tipo primitivo de su wrapper.