

1. Ejercicio #1 ()

- ¿Por qué esto es así? ¿Qué diferencia representa con otros lenguajes como C#?

Respuesta: En Java, no es posible comparar dos objetos de tipo String con el operador '==' o '!=', porque no está comparando el contenido, sino que compara referencias o posiciones en memoria, es decir que el bucle no termina a menos que se compare el mismo objeto String. Lo correcto para comparar los contenidos sería utilizar el método 'equals()'.

En C#, está permitido comparar Strings con '=='.

2. Ejercicio #2 ()

Respuesta: En el primer caso la comparación resulta "True" y en la segunda resulta "False". Para entender esto es preciso explicar lo que es el String pool en Java.

En Java, cada vez que se inicializa un String de la forma "String s = 'algo'", es almacenado en el String pool, que es un espacio en memoria donde se guardan los String 'literales'. Cuando se vuelve a declarar otro String de la misma manera, si tiene el mismo contenido, Java vuelve a utilizar la instancia del String anterior que se guardaba en el String pool, por lo que ambos apuntan a una misma instancia, es decir que referencian lo mismo. Por esta razón, al compararse ambos String con el operador '==', s1 y s2 referencian el mismo objeto y devuelve True.

En cambio, cuando se inicializa un nuevo String con la palabra 'new', se está creando un nuevo objeto String fuera del String pool. Es así que si s1 y s2 tienen el mismo contenido pero referencian dos objetos distintos.

Referencias:

<https://docs.oracle.com/javase/specs/jls/se17/html/jls-3.html#jls-3.10.5>

<https://muratakkkan.medium.com/understanding-and-using-the-java-string-pool-in-java-d60d3176716>

Propone una versión correcta del código del Ejercicio #1

Respuesta:

```
String s = "1";  
while (!s.equals("1000")) {  
    s += "0";  
}
```