

UT 03 – Utilización de objetos – Ejercicios Strings

Ejercicio 09

Crea un método `insertarGuionesCadena3`, que haga exactamente lo mismo que los de los ejercicios 01 y 02, pero usando `StringBuilder`. Aprovecha los métodos que ofrece `StringBuilder` para manipular cadenas.

Recordatorio ejercicio 02:

Crea un método `insertarGuionesCadena` que:

- *Recibe una cadena como parámetro*
- *Devuelve una nueva cadena con los mismos caracteres de la cadena uno a uno, pero separándolos con un guión.*
- *Si la cadena es null, devuelve null.*

Ejemplo:

`insertarGuionesCadena3("101 Dálmatas")` devolvería "1-0-1- -D-á-l-m-a-t-a-s"

Ejercicio 10

Crea un método `invertirCadena3`, que haga lo mismo que los métodos de los ejercicios 03 y 04, pero usando `StringBuilder`. Aprovecha los métodos que ofrece `StringBuilder` para manipular cadenas.

Recordatorio ejercicio 04:

Crea un método `invertirCadena` que:

- *Recibe una cadena como parámetro*
- *Devuelve una cadena que es igual que la recibida, pero escrita a la inversa.*
- *Si la cadena es null, devuelve null.*

Ejemplo:

`invertirCadena("murciélago")` devolvería "ogaléicrum".

Ejercicio 11

Crea un método `esPalíndromo2` que haga lo mismo que el método del ejercicio 05, pero usando `StringBuilder`.

Recordatorio ejercicio 07:

Realiza un método `esPalindromo` que

- *Recibe una cadena de caracteres*
- *Devuelve true si la cadena es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).*
- *Devuelve false si la cadena es null o si no es un palíndromo.*

Ejemplo:

- *esPalindromo("cactus") devuelve false*
- *esPalindromo(null) devuelve false*
- *esPalindromo("arenera") devuelve true*
- *esPalindromo("yo hago yoga hoy") devuelve true*

Ejercicio 12

Crea un método espejo2 que haga lo mismo que el método del ejercicio 06, pero usando StringBuilder.

Recordatorio ejercicio 08:

Realiza un método espejo que:

- *Recibe una cadena de caracteres*
- *Devuelve la cadena como si estuviera reflejada en un espejo. Esto es, la misma cadena a la que se añade la misma cadena invertida, pero compartiendo una letra, que será la última de la cadena original y la primera de la cadena invertida que se añade. Ver ejemplos.*
- *Si la cadena es null, devuelve null*
- *Este método crea palíndromos, y si usamos el método esPalindromo sobre el resultado de este método, siempre devolverá true.*

Ejemplos:

- *espejo("año") devuelve añoña*
- *espejo("bicicleta") devuelve "bicicletatclcib".*
- *esPalindromo(espejo(<cualquier cadena>)) siempre devolverá true, salvo si la cadena es null, en cuyo caso devolverá false.*

Ejercicio 13

Crea un método aniadirSeparadores que:

- *Recibe un número entero. Ejemplo: 34445623.*
- *Lo devuelve como un String formateado con puntos para separar las unidades de millar, las unidades de millón, etc. En el caso del ejemplo devolvería 34.445.623*

Para hacerlo:

- *Convierte el número a String.*
- *Usa un StringBuilder para insertar los "." en las posiciones adecuadas, usando el método insert.*

Ejercicio 14

Crea un método buscaYPasaMayusculas que:

- *Recibe dos cadenas de caracteres*

- Devuelve la primera cadena, pero transformando en mayúsculas la parte que coincide con la segunda cadena introducida. Por ejemplo, si se introducen las cadenas “Este es mi amigo Juan” y “amigo”, devolverá “Este es mi AMIGO Juan”.
- Usa StringBuilder.
- Si la primera cadena es null, devolverá null
- Si la segunda cadena de caracteres es null, devuelve la primera cadena sin modificar.

Ejercicio 15

Crea un método `eliminaCaracteresRepetidos` que:

- Recibe un parámetro de tipo String
- Devuelve la cadena de texto, pero tras haber eliminado todos los caracteres repetidos. Case-sensitive, es decir, considerando que una letra mayúscula (A) es diferente a la letra minúscula (a).
- Por ejemplo, si se recibe “permanentemente”. Devolvería “permant”
- Usa StringBuilder.