

PROGRAMACIÓN JAVA. EJERCICIOS CON CADENAS DE CARACTERES.

Clase String.

Estos ejercicios desde el 1 hasta el 6 se deben resolver usando la clase String. No se puede utilizar las clases StringBuilder, StringTokenizer ni el método split.

1. Escribe un programa que lea un texto y lo guarde en un String. Elimina la última palabra del texto introducido y muestra el texto resultante. Las palabras del texto se introducirán separadas por espacios en blanco.
2. Escribe el mismo programa del ejercicio anterior pero esta vez utilizando un método que reciba el texto original y devuelva el texto modificado.
3. Escribe un programa que lea un texto y muestre cuantas veces aparece un determinado carácter en el texto. El carácter se pide por teclado.
4. Método que recibe un texto y devuelve el número de palabras que contiene. Supondremos las palabras separadas por un espacio en blanco.
5. Escribe un método que reciba un String conteniendo un texto y otro String conteniendo caracteres y devuelva cuantas aparece esa secuencia de caracteres en el texto. Por ejemplo, si la frase es:
Durante cuatro días y cuatro noches, cualquier cuatrero con cuatro ...
Y los caracteres a buscar son "uatr"
El método devolverá: 4
6. Escribe un método que reciba un String y un carácter y devuelve cuántas veces aparece el carácter en el String y en qué posiciones se encuentra. Utiliza un ArrayList para guardar las posiciones.

Clases String, StringBuilder, StringTokenizer. Método split. Método replaceAll.

Estos ejercicios desde el 7 hasta el final se pueden resolver utilizando String, StringTokenizer y StringBuilder. En aquellos ejercicios que se puedan resolver mediante StringTokenizer o mediante Split, resuélvelo de las dos formas. Algunos ejercicios se pueden resolver también con el método replaceAll, si es el caso, resuélvelo también de esta forma.

7. Escribe un **método** que elimine los espacios en blanco sobrantes de una cadena de texto. Los espacios sobrantes pueden estar al principio, al final o entre las palabras del texto.
8. Escribe un **método** que quite la última palabra de un String. Supondremos que las palabras contenidas en el String están separadas por un espacio en blanco.
9. Escribe un programa que lea una serie de números enteros separados por espacios en blanco (todos los números en la misma línea y separados por un espacio en blanco) y los muestre ordenados de menor a mayor.

10. Dada la siguiente cadena:

```
"23-239 12-120:23=5203 28340:23\n238"
```

Escribe un programa que muestre:

Hay 9 números:

- 1) 23 Quedan 8
- 2) 239 Quedan 7
- 3) 12 Quedan 6
- 4) 120 Quedan 5
- 5) 23 Quedan 4
- 6) 5203 Quedan 3
- 7) 28340 Quedan 2
- 8) 23 Quedan 1
- 9) 238 Quedan 0

11. Escribe un **método** que reciba como entrada un string que representa un número y devuelva el mismo número (String) dividido por puntos cada tres dígitos. Por ejemplo, si el método recibe el String "12345678" devuelve "12.345.678"

12. Se dispone de un array de cadenas de caracteres. Cada cadena contiene el nombre completo de una persona con el formato:

apellido1<blanco>apellido2,<blanco>nombre

Escribe un programa que a partir de este array cree tres nuevos ArrayList que contengan el primer apellido, el segundo apellido y el nombre respectivamente.

13. A partir de un String que contiene parejas con el formato Deportista=Deporte; por ejemplo:

```
"Messi=Fútbol;Gasol=Baloncesto;Nadal=Tenis;Jordan=Baloncesto;"
```

Escribe un programa que muestre por pantalla:

Hay 4 deportistas:

Deportista	Deporte
Messi	Fútbol
Gasol	Baloncesto
Nadal	Tenis
Jordan	Baloncesto

14. Escribe un **método** que recibe como entrada un String que contiene una serie de NIFs separados por espacios en blanco. A su vez la letra del NIF va separada del número por un guión.
- Por ejemplo: "43434321-A 65473820-T 33238765-D 32994102-X 74771234-F"
- El método debe devolver un String sin los guiones.
- Para el String del ejemplo devolverá:
- "43434321A 65473820T 33238765D 32994102X 74771234F"
15. Escribe un **método** que recibe un String que contiene números ≥ 0 separados por comas y espacios en blanco y devuelve un String con los números separados solamente por guiones. Además, si se han añadido al String menos de cuatro guiones, se añadirá al principio y al final del String el caracter '*'.
Ejemplo 1: si el método recibe el String: 12 345,44,2 3,739
devolverá el String: 12-345-44-2-3-739
Ejemplo 2: si el método recibe el String: 12345,442 3,732
devolverá el String: *12345-442-3-732*
16. Escribe un **método** que reciba como entrada un String que contiene caracteres separados por espacios en blanco, comas y guiones y devuelva un String con los guiones y las comas sustituidos por espacios.
- Por ejemplo, si el método recibe el String "ab c,de-f-g h" devolverá el String "ab c de f g h".