

U5.- Strings

1. Escribe un programa que solicite tres cadenas de caracteres, una llamada nombre, otra llamada primerApellido y segundoApellido. El programa debe **crear** una cadena con el nombre completo y después mostrar por pantalla:
 - el nombre completo con todos los caracteres en minúsculas y luego en mayúsculas y su longitud
 - los dos primeros caracteres de la cadena concatenada (solo en el caso de que la longitud de esta sea de dos o más caracteres)
 - los dos últimos caracteres de la cadena concatenada (solo en el caso de que la longitud de esta sea de dos o más caracteres)
 - el número de ocurrencias en la cadena del último carácter
 - la cadena con todas las ocurrencias del primer carácter en mayúsculas
 - la cadena con tres asteriscos por delante y por detrás
 - la cadena invertida (Sólo la muestra, no la guarda invertida)

Ej.:

- Nombre →pepe
- primerApellido →lopez
- segundoApellido →martinez
- nombreCompleto →pepelopezmartinez

```
Nombre:pepe
Apellido1:lopez
Apellido2:
martinez
pepelopezmartinez--PEPELOPEZMARTINEZ---17
pe
ez
pepelopezmartinez
La z aparece 2 veces
PePeloPezmartinez
***elopezmarti***
zenitramzepolepep
zenitramzepolepep
-----
```

2. Escribe una clase con los métodos que se piden a continuación y un método main en el que se pruebe su funcionamiento:
 1. Escribe un método que, dada una cadena de caracteres, devuelve la mitad inicial de la cadena. El método principal llamará a este método con las cadenas “Hola que tal” y “Adiós”, para probarlo
 2. Escribe un método que, dada una cadena de caracteres, sustituya todas las ocurrencias del texto “es” por “no es”.

Si la cadena de caracteres es “esto1234es5678bueno900” el método debe devolver “no esto1234no es5678bueno900”

3. Escribe un método que dado un texto y una cadena de caracteres, cuente cuántas veces aparece la misma en dicho texto.

Por ejemplo, para la cadena “En un lugar de la Mancha, de cuyo nombre no puedo acordarme”, si se cuenta cuántas veces aparece “o”, deberá indicar que son 5.

EXPRESIONES REGULARES

Prácticamente todos los lenguajes de programación manejan el concepto de expresiones regulares. Las expresiones regulares son una forma de buscar "patrones" dentro de una cadena de texto. Por ejemplo, nos facilitan saber si una cadena tiene el formato de fecha estilo "dd/mm/yyyy", o si tiene un formato de email válido estilo "nombre@dominio.com", etc

En casi todos los lenguajes de programación las expresiones regulares son muy similares (usan sintaxis muy similares)

Más información sobre expresiones regulares en Java:

<http://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html#sum>
[http://chuwiki.chuidiang.org/index.php?title=Expresiones Regulares en Java](http://chuwiki.chuidiang.org/index.php?title=Expresiones_Regulares_en_Java)

Página interesante para probar expresiones regulares (aunque no soporta Java, sirve para ver expresiones que son comunes a otros lenguajes): <https://regexr.com/>

Expresiones comunes:

[abc]	a, b, or c (simple class)
[^abc]	Any character except a, b, or c (negation)
[a-zA-Z]	a through z or A through Z, inclusive (range)
[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)
[a-z&&[def]]	d, e, or f (intersection)
[a-z&&[^bc]]	a through z, except for b and c: [ad-z] (subtraction)
[a-z&&[^m-p]]	a through z, and not m through p: [a-lq-z] (subtraction)
Predefined character classes	
.	Any character (may or may not match line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

1. Escribe un método que, dado un String, devuelve otro objeto String en el que se cambian todas las vocales minúsculas del original por la letra 'a'.

¿cuál sería la expresión regular para vocal minúscula?

2. Escribe un método que sustituya todos los grupos de dígitos por un único carácter asterisco, es decir, si la cadena de caracteres es "esto1234es5678bueno900" el método debe devolver "esto*es*bueno*".

Nota: La expresión regular \d+ significa un dígito o más.

Puedo usar replaceAll("\\d+", "*"); //aparecen dos \\ porque el primero es el carácter de 'escape', es decir para que el segundo se tome tal cual y no como un carácter especial de java

Repaso

- Método que devuelve el número de palabras de la cadena que recibe como parámetro.
- Método que devuelve las siglas de la cadena que recibe como parámetro (Inicial de cada palabra en mayúsculas)

Ej.: Producto Bruto Interno --> PIB