

Problema 4: Practicar con strings, instrucciones condicionales y bucles

Datos personales

Apellidos:

Nombre:

1 Strings

Objetivos

Practicar con la manipulación de strings

Descripción

Escribir las instrucciones Java necesarias para realizar las siguientes operaciones con los Strings `s1` y `s2`. Se supone que tienen cada uno seis caracteres o más.

- Mostrar en pantalla las longitudes de ambos Strings
- Mostrar en pantalla los tres primeros caracteres de `s2`, uno por línea
- Crear una *nueva* variable de tipo String que se componga de los tres primeros caracteres de `s1`, seguidos por los caracteres de `s1` posteriores al quinto
- Obtener una variable booleana que indique si el String `s1` contiene algún espacio en blanco, o no
- Obtener una variable booleana que indique si los contenidos de los cuatro primeros caracteres de los Strings `s1` y `s2` son iguales o no

Nota: para comparar los contenidos de Strings, obtener fragmentos de un String o buscar caracteres en un String existen métodos que se pueden encontrar en la documentación de la clase String de Java 8 (accesible por Internet). Son los métodos `equals`, `substring` e `indexOf`, respectivamente.

Se pide

Las instrucciones necesarias para realizar las operaciones indicadas. Basta con las instrucciones, sin necesidad de encapsularlas en un método.

Respuesta:

<poner aquí las instrucciones>

2 Instrucción condicional simple

Objetivos

Practicar con la instrucción condicional simple (if)

Descripción

Se desea escribir un método Java para determinar si un código postal que se pasa como parámetro en forma de String de 5 caracteres es de Cantabria o no. Los códigos de Cantabria empiezan con las cifras “39” y siguen con tres cifras. Para saber si un carácter es una cifra se puede usar el método isDigit() de la clase Character. El método retornará el texto “Cantabria” si el código postal es de Cantabria, o el texto “Otro” en caso contrario.

La cabecera del método será:

```
public static String zonaPostal (String codigoPostal)
```

Respuesta:

<poner aquí el método>

3 Instrucción condicional múltiple

Objetivos

Practicar con la instrucción condicional múltiple (switch)

Descripción

Utilizando la instrucción switch escribir un método Java al que se le pasa un carácter y un número real, y que devuelve el número real multiplicado por el factor de escala obtenido a partir del carácter, según los valores de la tabla:

Carácter	Factor de escala
'n'	0,000000001
'u'	0,000001
'm'	0,001
'k'	1000,0
'M'	1000000,0
'G'	1000000000,0

Si se pasa otro carácter que no esté en la tabla, retornar Double.NaN para indicar el error.

La cabecera del método será:

```
public static double convierte (char escala, double valor)
```

Respuesta:

<poner aquí el método>

4 Algoritmo iterativo

Objetivos

Practicar con un algoritmo iterativo y el pseudocódigo

Descripción

Se desea escribir un método que implementa un *algoritmo iterativo* que calcula y retorna la suma del siguiente desarrollo en serie de la función $f(x)=(1+x)e^x$ para valores de x no muy grandes (por encima de $x=10$ se necesitan bastantes iteraciones para obtener una aproximación aceptable):

$$f(x) \approx \sum_{i=0}^n \frac{i+1}{i!} x^i$$

El método recibe como parámetros los valores x y n . Para hacer más eficiente este algoritmo, observar que los valores del factorial $(i+1)!$ y de la potencia x^{i+1} se pueden obtener en la iteración número i para la iteración siguiente multiplicando el factorial anterior por $i+1$ y la potencia anterior por x , de acuerdo a estas expresiones:

$$\begin{aligned}(i+1)! &= i! \cdot (i+1) \\ x^{i+1} &= x^i \cdot x\end{aligned}$$

De esta manera nunca hace falta calcular factoriales ni potencias; tan solo productos y sumas.

Respuesta:

<poner aquí el pseudocódigo del método>