

Tarea 11 – Métodos con While – for - Switch

Enunciado

1. Escribe un método llamado **imprimeNumeroEnLetra**. El método tiene un parámetro llamado **numero** que es entero. El método debe imprimir "CERO", "UNO", "DOS", ... "NUEVE", "OTRO" si el parámetro **numero** es 0, 1, 2, 9 u otro para cualquier otro número incluidos los números negativos. Usa la estructura switch.
Ayuda: El método **imprimeNumeroEnLetra** debe ser public static.
2. Escribe un método llamado **esImpar** con un parámetro int y llámalo **numero**. El método debe devolver un valor booleano.
Verifica que el número sea > 0, si no es mayor a 0 el método devuelve falso.
Si el número es impar, devuelve verdadero; de lo contrario, devuelve falso.

Escribe un segundo método llamado **sumarImpares** que tenga 2 parámetros, uno int **inicio** y otro int **fin**, que representan un rango de números. El método devuelve un int.

El método debe usar un bucle for para sumar todos los números impares en ese rango, incluido el final y devolver la suma. El bucle debe llamar al método **esImpar** para comprobar si cada número que se itera dentro del bucle es impar.

El parámetro **fin** debe ser mayor o igual que el **inicio** y los parámetros de **inicio** y **fin** deben ser mayores que 0.

Si esas condiciones no se cumplen, el método devuelve -1 para indicar una entrada no válida.

Ejemplo input/output:

* sumarImpares(1, 100); → debe devolver 2500

* sumarImpares(-1, 100); → debe devolver -1

* sumarImpares(100, 100); → debe devolver 0

* sumarImpares(13, 13); → debe devolver 13 (este conjunto contiene un número, 13, y es impar)

* sumarImpares(100, -100); → debe devolver -1

* sumarImpares(100, 1000); → debe devolver 247500

Ayuda: para saber si un número es par se usa: $i \% 2 == 0$

Ambos métodos deben definirse como public static, como lo hemos estado haciendo hasta ahora en el curso.

El método **esImpar** se utiliza dentro del método **sumarImpares**.

3. Escribe un método llamado **imprimirFactores** que recibe un parámetro entero llamado **n**. Si el $n < 1$ el método imprime "Valor inválido". El método debe imprimir todos los factores del número. Un factor de un número es un entero que es divisible con resto 0 por ese número. Por ejemplo, 3 es un factor de 6, porque $6/3=2$ resto 0.

Ejemplo input/output

imprimirFactores(10) → devuelve:

1
2
5
10

imprimirFactores(6); → devuelve: 1 2 3 6 (en diferentes líneas)

imprimirFactores(-4); → devuelve: Valor inválido

4. Escribe un método llamado **esPalindromo** con un parámetro entero llamado **n**. El método devuelve un booleano. El valor devuelto será true cuando el número es un palíndromo, en otro caso devolverá false.

¿Qué es un número palíndromo? Es un número que si le damos la vuelta, es decir, que si lo leemos de atrás hacia adelante es el mismo número. Por ejemplo, 12321, 121, 1001, etc.

Ayuda. Para verificar si un número es palíndromo deberemos calcular el número en reverso y almacenarlo en una variable para al final comparar si los dos números son iguales.

Cálculo de un número reverso. Primero crea e inicializa otra variable llamada reverso = 0.

Extrae el último dígito del número n utilizando el módulo y almacénalo en otra variable: **ultimoDigito = n % 10**.

Aumenta el valor posicional de reverso en uno, para hacer esto tienes que multiplicar la variable reverso por 10: **reverso = reverso * 10;**

Suma a reverso el último dígito: **reverso +=ultimoDigito;**

Como el último dígito del número n ha sido procesado, elimínalo del número utilizando la / 10: **n = n / 10;**

Repite estos pasos hasta que el número n sea 0.

Utiliza la estructura while.

Fuera del bucle, compara el número original introducido con reverso. No te olvides de guardar el número original introducido en alguna variable ya que n se va modificando.

5. Escribe un método llamado **insertaImprimeSumaYMedia** sin parámetros y sin devolver nada. El método leerá números introducidos por el usuario por pantalla hasta que el usuario inserte algo que no es un número. Entonces el programa saldrá del while e imprimirá la suma y la media de los números introducidos con formato: SUM = XX AVG = YY, donde XX representa la suma de los datos introducidos y YY la media.

Ejemplo input:

1

2

3

4

5

a

Output

SUM = 15 AVG = 3

Input

Prueba

Output

SUM = 0 AVG = 0

Ayuda. Utiliza un while infinito. Usa break cuando el usuario inserte algo distinto a un número para salir del bucle. Si el usuario no inserta ningún número válido muestra SUM = 0 AVG = 0.

Utiliza Scanner para leer los números. Ayúdate del método hasNextInt() de Scanner para saber si ha insertado un número o no. Utiliza Math.round para redondear el cálculo de la media.

Necesitarás una variable contadora para contar el número de dígitos introducidos y que te ayude luego a hacer la media. Otra variable suma que almacene el resultado de sumar los dígitos insertados.

Entrega

- Realiza capturas con las salidas del programa en un PDF.
- Copia y pega el código Java que has implementado.