

PROGRAMACIÓN JAVA. EJERCICIOS TEMA 4. MÉTODOS

1. Escribe un método que reciba un número entero positivo y calcule si es un número perfecto. El método devolverá true si el número es perfecto y false si no lo es.
2. Escribe un método que reciba dos números enteros A y B. El método debe calcular y devolver el producto de A * B sin utilizar el operador *. El cálculo del producto se realizará mediante sumas.
Por ejemplo, si el método recibe los valores A = 3 B = 4 debe calcular 3 * 4 realizando las sumas 3 + 3 + 3 + 3. Para realizar el cálculo y obtener el resultado correcto debemos tener en cuenta que A y B pueden ser positivos o negativos.
3. Escribe un método que reciba un número entero y devuelva el número de cifras que tiene.
4. Escribe un método que reciba un número N entero positivo y un número entero k entre 1 y 9. El método debe devolver un número entero formado por los k dígitos del número N situados más a la derecha. Ejemplo para N = 12345 y k = 2 el método devolverá 45. Si el número N tiene menos cifras que las que se quieren obtener, el método devolverá -1. Por ejemplo para N = 123 y k = 4 el método devolverá -1 porque el número tiene menos de 4 cifras. Utiliza el método creado en el ejercicio anterior para saber las cifras que tiene el número N.
5. Escribe un método que reciba un número entero ≥ 0 y calcule y devuelva su factorial.
6. Escribe un programa que lea un número entero N mayor o igual a cero y muestre el factorial de todos los números desde 0 hasta N. Utiliza el método factorial creado en el ejercicio anterior.
7. Un número combinatorio es el valor numérico de las combinaciones ordinarias (sin repetición) de un conjunto de n elementos tomados en grupos de r, siendo n y r dos números enteros y positivos tales que $n \geq r$. Matemáticamente, un número combinatorio se calcula así:

$$C_{n,r} = \frac{n!}{r! (n - r)!}$$

Escribe un método que reciba dos números enteros n y r y devuelva el número combinatorio correspondiente. Utiliza el método factorial creado en el ejercicio 5 para calcularlo.

8. Escribe un método que calcule de forma aproximada el seno de un ángulo x, expresado en radianes, utilizando el desarrollo de Taylor:

$$\text{sen}(x) = x - (x^3 / 3!) + (x^5 / 5!) - (x^7 / 7!) + (x^9 / 9!) - (x^{11} / 11!) \dots\dots\dots$$

El método recibe el ángulo x en radianes y un número N que indica el número de términos que se van a emplear para calcular la aproximación. A mayor valor de N más se aproximará el resultado obtenido al valor real.

Por ejemplo:

El seno de 10 radianes es: -0,54402111088936981340474766185138

Nuestro método debe obtener un resultado cercano a ese valor.

Ejemplo de ejecución:

```
Introduce ángulo en radianes: 10
Introduce número de términos para realizar el cálculo: 20
sen(10.0) = -0.5440211137210484
```

Ejemplo de ejecución:

```
Introduce ángulo en radianes: 10
Introduce número de términos para realizar el cálculo: 40
sen(10.0) = -0.54402111088927
```

9. Escribe un método que compruebe si un carácter que se le pasa como parámetro es una vocal o no. El método devolverá true si es una vocal o false si no lo es.
10. El número e, base de los logaritmos neperianos, es un número irracional y su valor exacto no puede ser expresado con un número finito de cifras decimales o con decimales periódicos. Su valor aproximado con 20 cifras decimales es 2.71828182845904523536...
Escribe un método que calcule el valor aproximado del número e utilizando la siguiente fórmula:
$$e = 1/0! + 1/1! + 1/2! + 1/3! + \dots + 1/N!$$

El método recibe el valor de N y devuelve el cálculo aproximado de e. A mayor valor de N más se aproximará el resultado obtenido al real.

11. Escribe un método que reciba un número entero y muestre todas las tablas de multiplicar desde la tabla del 2 hasta la tabla del número recibido.
12. Escribe un método que reciba un número entero positivo y devuelva true si la primera cifra del número es igual a la última. El método devolverá false en caso contrario. Si el número tiene solo una cifra el método devolverá true.
13. Escribe un método que reciba dos números A y B de tipo double. El método debe devolver un String con este formato: "A = *valor de A con tres cifras decimales* B = *valor de B con tres cifras decimales*".

Por ejemplo, si A = 2.44567 y B = 1.4423056

El método devolverá este String: "A = 2,446 B = 1,442"

14. Escribe un método que reciba un número entero positivo N y otro número entero p y devuelva la cifra de N que se encuentra en la posición p. En el método main se introducirá por teclado el valor del número y la posición de la cifra que queremos obtener y se comprobarán que sean válidos. El valor de la posición debe ser un número entero positivo menor o igual que el número total de cifras que tenga el número del que queremos obtener la cifra. Si el valor de la posición no es correcto se vuelve a pedir. Se considerará que la cifra de la derecha ocupa la posición 1.

Aunque en el método main se compruebe que el valor del número y el de la posición sean valores correctos, dentro del método también se comprobará. En este caso si los valores que se reciben no son válidos el método devolverá -1.

Debemos tener en cuenta que un método se puede usar en cualquier programa y tenemos que asegurarnos en la medida de lo posible que no se produzca un error por trabajar con valores no válidos. En este ejercicio estamos comprobando que el valor del número y la posición sean válidos antes de enviárselos al método por lo que en teoría no sería necesario comprobarlo de nuevo dentro del método. Pero este método lo podemos usar en otro programa en el que los datos no se comprueben antes y al método le llegue cualquier valor para la posición o el número y en ese caso el método podría fallar.