

## Ejercicio 1.-

Crea un programa Java que simule una calculadora, que permita operar con dos números.

Para ello, el programa:

- Mostrará un menú con las siguientes opciones:
  - o 1.- Sumar (A+B)
  - o 2.- Restar (A-B)
  - o 3.- Multiplicar (A\*B)
  - o 4.- Dividir (A/B)
  - o 5.- Potencia (A elevado a B) (No es válido utilizar métodos de la API Java)
  - o 0.- Salir
- Preguntará al usuario qué operación desea realizar, y:
  - o Si no es una opción válida, avisará del error, volverá a mostrar el menú y preguntará de nuevo.
  - o Si es un 0 (salir). Se despedirá del usuario y terminará.
  - o Si es cualquiera de las demás opciones, preguntará al usuario los números A y B, realizará la operación y mostrará el resultado.
- Tras cada operación realizada se volverá a mostrar el menú y se preguntará de nuevo al usuario que operación quiere realizar.

A tener en cuenta:

- Es obligatorio usar constantes (private static final ..., justo antes de la declaración del método main) para las posibles opciones elegidas por el usuario (números del 0 al 5), y usarlas a lo largo de todo el programa, evitando en la medida de lo posible el uso de números mágicos.
- Es obligatorio usar la estructura switch en algún punto del programa.
- Es obligatorio crear, al menos, los siguientes métodos:
  - o Método “mostrarMenu”, que mostrará el menú en la consola.
  - o Método “pedirOpcionUsuario”, que:
    - Recibe como parámetro un objeto Scanner.
    - Devuelve un entero con la opción elegida por el usuario.
    - Mostrará el menú (usando el método anterior).
    - Pedirá al usuario que elija una opción del menú.
    - Si el usuario introduce un valor erróneo (no es de 1 a 5 ni el cero), informará del error, y volverá a preguntar hasta que introduzca el valor correcto.
  - o Método “realizarOperacion”, que:
    - Recibe como parámetros un objeto Scanner y la opción elegida por el usuario.
    - No devuelve nada
    - Preguntará al usuario los números A y B, y realizará la operación, mostrando el resultado. Para la potencia utilizamos el método correspondiente.
    - En el caso de la división, si B es 0, en lugar de hacer la operación, avisará de que es una operación ilegal.
    - Si alguna operación puede producir números con decimales, se mostrará el resultado con dos dígitos decimales.

- Se puede asumir que en todos los casos el usuario siempre escribirá un número, no hay que comprobar si el usuario escribe letras.
- o Método “calcularPotencia”, que:
- Recibe como parámetros dos números enteros.
  - Calcula el resultado de elevar el primer número al segundo. La potencia  $a^n$  representa el producto que tiene  $n$  veces el número  $a$ . Ejemplos:

$$2^1 = 2$$

$$2^2 = 2 \cdot 2 = 4$$

$$2^3 = 2 \cdot 2 \cdot 2 = 8$$

$$2^4 = 2 \cdot 2 \cdot 2 \cdot 2 = 16$$

#### Ejemplo de ejecución (no tiene que ser exactamente igual):

```

1.- Sumar (A+B)
2.- Restar (A-B)
3.- Multiplicar (A*B)
4.- Dividir (A/B)
5.- Potencia (A elevado a B)
0.- Salir

```

1

Primer número: 3

Segundo número: 2

$3 + 2 = 5$

```

1.- Sumar (A+B)
2.- Restar (A-B)
3.- Multiplicar (A*B)
4.- Dividir (A/B)
5.- Potencia (A elevado a B)
0.- Salir

```

4

Primer número: 3

Segundo número: 0

No se puede dividir por cero

```

1.- Sumar (A+B)
2.- Restar (A-B)
3.- Multiplicar (A*B)
4.- Dividir (A/B)
5.- Potencia (A elevado a B)
0.- Salir

```

4

```
Primer número: 4
Segundo número: 3
4 / 3 = 1.33
1.- Sumar (A+B)
2.- Restar (A-B)
3.- Multiplicar (A*B)
4.- Dividir (A/B)
5.- Potencia (A elevado a B)
0.- Salir
5
Primer número: 2
Segundo número: 4
2 elevado a 4 = 16
1.- Sumar (A+B)
2.- Restar (A-B)
3.- Multiplicar (A*B)
4.- Dividir (A/B)
5.- Potencia (A elevado a B)
0.- Salir
0
Gracias por usar la calculadora. Hasta la próxima.
```

### Ejercicio 2.-

Desarrolla un programa en Java que pida al usuario un número entero positivo que represente la cantidad de términos a mostrar de la sucesión de Fibonacci. El programa deberá validar que el número sea mayor que cero.

La sucesión de Fibonacci comienza con los números 0 y 1. A partir de estos, cada término es la suma de los dos anteriores. (0, 1, 1, 2, 3, 5, 8, ...). El programa deberá imprimir por pantalla los **n** primeros términos de la sucesión, separados por comas.

### Ejemplo de ejecución:

```
Introduce la cantidad de términos de la sucesión de Fibonacci a
mostrar: -2
La cantidad debe ser un número positivo.
Introduce la cantidad de términos de la sucesión de Fibonacci a
mostrar: 8
Los primeros 8 términos son: 0, 1, 1, 2, 3, 5, 8, 13
```

### Ejercicio 3.-

Escribe un programa en Java que solicite al usuario dos números enteros, uno menor y otro mayor. El programa deberá validar que el primer número introducido es estrictamente menor que el segundo. Si no es así, deberá volver a pedirlos hasta que la condición se cumpla.

Una vez introducidos los números correctamente, el programa deberá calcular y mostrar por pantalla:

1. La **suma** y la **media aritmética** de todos los **números pares** que se encuentren en el intervalo comprendido entre el número menor (no incluido) y el número mayor (incluido).
2. La **suma** y la **media aritmética** de todos los **números múltiplos de 3** en ese mismo intervalo (menor no incluido, mayor incluido).

La media debe mostrarse con dos decimales.

#### **Ejemplo de ejecución:**

Introduce el número menor: 10

Introduce el número mayor: 5

Números incorrectos. El menor tiene que ser realmente menor que el mayor.

Introduce el número menor: 1

Introduce el número mayor: 10

La suma de los números pares entre 1 (no incluido) y 10 (incluido) es 30

La media de estos números es 6.00

La suma de los números múltiplos de tres entre 1 (no incluido) y 10 (incluido) es 18

La media de estos números es 6.00