

Ejercicios clases y métodos estáticos

[B.01] - Validador de rangos

Realizar una clase llamada **Validadora** que posea un método estático llamado **validar** con la siguiente firma:

boolean validar(int valor, int min, int max)

- **valor**: dato a validar.
- **min**: mínimo valor incluido.
- **max**: máximo valor incluido.

Pedir al usuario que ingrese 10 números enteros. Validar con el método desarrollado anteriormente que estén dentro del rango -100 y 100.

Terminado el ingreso mostrar el *valor mínimo ingresado*, *valor máximo ingresado* y el *promedio*.

IMPORTANTE

Utilizar variables escalares.

[B.02] - Validadora

Realizar un programa que sume números enteros hasta que el usuario lo determine, por medio de un mensaje “¿Continúa? (S/N)”.

En el método estático **validarS_N()** de la clase **ValidarRespuesta**, se validará el ingreso de opciones.

El método NO recibe parámetros y devuelve un valor de tipo booleano, **true** si se ingresó una “S” y **false** si se ingresó una “N”.

El método deberá validar si otro caracter fue ingresado mostrando un mensaje de error y pidiendo el reingreso del mismo.

[B.03] - La calculadora

Realizar un programa que permita realizar operaciones matemáticas simples (suma, resta, multiplicación y división).

Crear una clase llamada **Calculadora** que posea dos métodos estáticos (de clase):

calcular (público): Recibirá tres parámetros, el primer parámetro será el primer operando, el segundo parámetro será el segundo operando y el tercer y último parámetro será la operación matemática a realizar. El método devolverá el resultado de dicha operación.

validar (privado): Recibirá como parámetro al segundo operando de la función **calcular**. Este método se debe utilizar sólo cuando la operación elegida sea la DIVISIÓN. Este método devolverá **true** si el operando es distinto de cero, **false** caso contrario.

Se le debe pedir al usuario que ingrese dos números y la operación que desea realizar (ingresando el caracter **+**, **-**, *****, o **/**).

El usuario decidirá cuándo finalizar el programa.

[B.04] - Conversor numérico

Desarrollar una clase llamada **Conversora**, que posea dos métodos de clase (estáticos):

- **String decimalBinario(double)**. Convierte un número de decimal a binario.
- **double binarioDecimal(String)**. Convierte un número binario a decimal.

[B.05] - Aprendete las tablas

Crear una aplicación de consola que permita al usuario ingresar un número entero.

Desarrollar un método estático que reciba un número y devuelva la tabla de multiplicación de ese número en formato **String**.

Mostrar en la consola el resultado.

Por ejemplo, si se ingresa el número 2 la salida deberá ser:

Tabla de multiplicar del número 2:

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18

*Bonus: Utilizar la clase **StringBuilder** combinada con **Strings** interpolados para armar el resultado.*

[B.06] - Calcular un factorial

El [factorial](#) de un número es una operación que consiste en multiplicar un número “n” por todos los números enteros positivos que estén debajo de él, por ejemplo el factorial de 3 es el resultado de multiplicar 3 por 2 por 1.

En una aplicación de consola, desarrollar un método estático que calcule el factorial de un número dado.

[B.07] - Calculadora de áreas

Realizar una clase llamada `CalculadoraDeArea` que posea tres métodos de clase (estáticos) que realicen el cálculo del área que corresponda:

- public static double `calcularAreaCuadrado`(double longitudLado)
- public static double `calcularAreaTriangulo`(double base, double altura)
- public static double `calcularAreaCirculo`(double radio)

El ingreso de los datos como la visualización se deberán realizar desde el método `main()`.

[B.08] - Pitágoras estaría orgulloso

Crear una aplicación de consola que pida al usuario ingresar la base y la altura de un triángulo en centímetros.

El programa deberá calcular la longitud de la hipotenusa aplicando el [teorema de pitágoras](#) y usar los métodos `pow` y `sqrt` de la clase `Math` para realizar los cálculos.

Mostrar el resultado en la consola.

[B.09] - El tiempo pasa...

Crear un método estático que reciba una fecha y calcule el número de días que pasaron desde esa fecha hasta la fecha actual. Tener en cuenta los años bisiestos.

Pedir por consola la *fecha de nacimiento* de una persona (día, mes y año) y calcule el *número de días vividos* por esa persona hasta la fecha actual utilizando el método desarrollado anteriormente.

Ayudarse con las funcionalidades del tipo `LocalDateTime` para resolver el ejercicio.