



Ejercicios

Clases y métodos estáticos

Clases y métodos estáticos

Antes de empezar

- Deberá asignarle a la propiedad 'Title' de la clase Console, el número de ejercicio.

Por ejemplo:

Console.Title = "Ejercicio Nro ##" donde ## será el número del ejercicio.

- Del mismo modo se deberán nombrar los proyectos de consola.

Por ejemplo:

Ejercicio_##.

- Para visualizar los valores decimales de los ejercicios, deberá dar el siguiente formato de salida al método Write/WriteLine: "#,###.00".

I01. Validador de rangos

Realizar una clase llamada Validador que posea un método estático llamado Validar con la siguiente firma:

```
bool 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.

*Nota: Utilizar variables escalares, **NO** utilizar vectores/arrays.*

I02. ¿Desea continuar?

Realizar un programa que sume números enteros hasta que el usuario lo determine por medio de un mensaje "¿Desea continuar? (S/N)".

Crear la clase Validador y el método estático ValidarRespuesta, que validará el ingreso de respuestas.

El método devolverá un valor de tipo booleano, *TRUE* si se ingresó una 'S' y *FALSE* si se ingresó cualquier otro valor.

I03. Conversor binario

Desarrollar una clase llamada `Conversor` que posea dos métodos de clase (estáticos):

El método `ConvertirDecimalABinario` que convierte un número entero del sistema decimal al sistema binario.

```
public String ConvertirDecimalABinario(int numeroEntero) {}
```

El método `ConvertirBinarioADecimal` que convierte un número entero del sistema binario al sistema decimal.

```
public int ConvertirBinarioADecimal(int numeroEntero) {}
```

*Nota: **NO** utilizar los atajos del lenguaje, hacerlo mediante estructuras de iteración y selección.*

I04. 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 operando, el segundo operando y la operación matemática. El método devolverá el resultado de la operación.

`Validar` (privado): Recibirá como parámetro el segundo operando. Este método se debe utilizar sólo cuando la operación elegida sea la DIVISIÓN. Este método devolverá *true* si el operando es distinto de cero.

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.

I05. 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.

Se deberá utilizar la clase `StringBuilder` combinada con **strings interpolados** para armar el resultado.

Mostrar en la consola el resultado.

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

Tabla de multiplicar del número 2:

$2 \times 1 = 2$
 $2 \times 2 = 4$
 $2 \times 3 = 6$
 $2 \times 4 = 8$
 $2 \times 5 = 10$
 $2 \times 6 = 12$
 $2 \times 7 = 14$
 $2 \times 8 = 16$
 $2 \times 9 = 18$
 $2 \times 10 = 20$

I06. Calculadora de áreas

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

```
public double CalcularAreaCuadrado(double longitudLado) {}

public double CalcularAreaTriangulo(double base, double altura) {}

public double CalcularAreaCirculo(double radio) {}
```

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

I07. 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.

I08. 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 **DateTime** para resolver el ejercicio.