

EJERCICIOS RESUELTOS Y PROPUESTOS

PARAMS, IN, REF, OUT

EJERCICIOS RESUELTOS

Ejercicio 1: Calcular el área y el perímetro de un rectángulo con out

Crea un método llamado `CalcularAreaYPerimetro` que acepte la longitud y el ancho de un rectángulo. Usa `out` para devolver tanto el área como el perímetro del rectángulo.

Instrucciones:

1. Solicita al usuario ingresar la longitud y el ancho del rectángulo.
2. Usa el método `CalcularAreaYPerimetro` para calcular y devolver el área y el perímetro.
3. Imprime los resultados.

RESOLUCION:

```
6
7 namespace ConsoleApp1
8 {
9     // 0 referencias
10    internal class Restangulo
11    {
12        // 0 referencias
13        // Método para calcular área y perímetro usando parámetros out
14        public void CalcularAreaYPerimetro(double longitud, double ancho, out double area, out double perimetro)
15        {
16            area = longitud * ancho; // Cálculo del área
17            perimetro = 2 * (longitud + ancho); // Cálculo del perímetro
18        }
19    }
```

CLASE PRINCIPAL

```
namespace ConsoleApp1
{
    // 0 referencias
    class Program
    {
        // 0 referencias
        static void Main(string[] args)
        {
            // Crear una instancia de la clase Rectangulo
            Restangulo rectangulo = new Restangulo();

            // Pedir datos al usuario
            Console.WriteLine("Ingrese la longitud del rectángulo:");
            double longitud = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("Ingrese el ancho del rectángulo:");
            double ancho = Convert.ToDouble(Console.ReadLine());

            // Variables para almacenar los resultados
            double area, perimetro;

            // Llamar al método para calcular área y perímetro
            rectangulo.CalcularAreaYPerimetro(longitud, ancho, out area, out perimetro);

            // Imprimir los resultados
            Console.WriteLine($"El área del rectángulo es: {area}");
            Console.WriteLine($"El perímetro del rectángulo es: {perimetro}");
        }
    }
}
```

RESULTADO ESPERADO

```
C:\Windows\system32\cmd.e: X + v

Ingrese la longitud del rectángulo:
12
Ingrese el ancho del rectángulo:
45
El área del rectángulo es: 540
El perímetro del rectángulo es: 114
Presione una tecla para continuar . . .
```

Ejercicio 2: Devolver el primer y último elemento de un arreglo con out

Crea un método llamado `ObtenerPrimeroYUltimo` que acepte un arreglo de objetos (`object[]`) y use `out` para devolver el primer y último elemento del arreglo.

Instrucciones:

1. Declara un arreglo de objetos con diferentes tipos de datos.
2. Usa el método `ObtenerPrimeroYUltimo` para obtener el primer y el último elemento del arreglo.

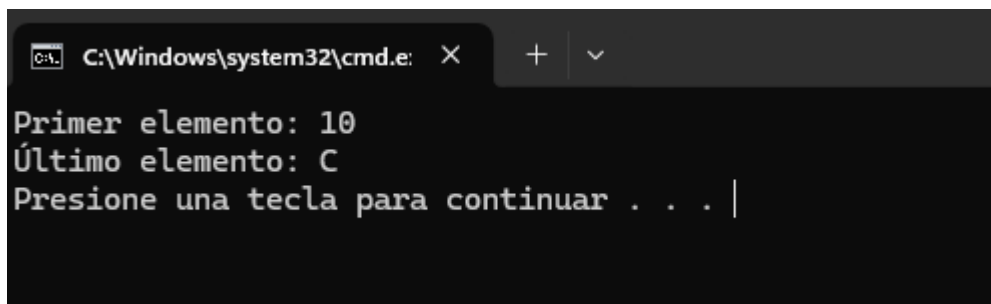
RESOLUCION

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace ConsoleApp1
8  {
9      2 referencias
10     internal class ArregloUtilidades
11     {
12         // Método que devuelve el primer y último elemento de un arreglo utilizando out
13         1 referencia
14         public void ObtenerPrimeroYUltimo(object[] arreglo, out object primero, out object ultimo)
15         {
16             if (arreglo.Length > 0)
17             {
18                 primero = arreglo[0]; // Primer elemento del arreglo
19                 ultimo = arreglo[arreglo.Length - 1]; // Último elemento del arreglo
20             }
21             else
22             {
23                 // Si el arreglo está vacío, asignamos null
24                 primero = null;
25                 ultimo = null;
26             }
27         }
28     }
29 }
```

PROGRAMA PRINCIPAL

```
1  using System;
2
3  namespace ConsoleApp1
4  {
5      {
6          0 referencias
7          class Program
8          {
9              0 referencias
10             static void Main(string[] args)
11             {
12                 // Declarar un arreglo de objetos con diferentes tipos de datos
13                 object[] arreglo = { 10, "hola", true, 3.14, 'A', 'C' };
14
15                 // Instancia de la clase ArregloUtilidades
16                 ArregloUtilidades utilidades = new ArregloUtilidades();
17
18                 // Variables para almacenar el primer y último elemento
19                 object primero, ultimo;
20
21                 // Llamada al método para obtener el primer y último elemento
22                 utilidades.ObtenerPrimeroYUltimo(arreglo, out primero, out ultimo);
23
24                 // Imprimir los elementos obtenidos
25                 Console.WriteLine($"Primer elemento: {primero}");
26                 Console.WriteLine($"Último elemento: {ultimo}");
27             }
28         }
29     }
```

RESULTADO ESPERADO



```
C:\Windows\system32\cmd.exe
Primer elemento: 10
Último elemento: C
Presione una tecla para continuar . . . |
```

EJERCICIOS PROPUESTOS

1. Cálculo de promedio con params y validación

Desarrolla un método llamado `CalcularPromedio` que acepte un número variable de calificaciones como parámetros utilizando `params`. Si no se pasa ninguna calificación, el método debe retornar 0. Si se pasan calificaciones, debe calcular y retornar el promedio.

2. Filtrar y contar elementos in

Desarrolla un método llamado ContarCadenas que acepte un valor in como referencia de un número mínimo de caracteres y un params object[] que contenga cadenas. El método debe contar cuántas cadenas tienen más caracteres que el valor in.

3. Cálculo de área de figuras geométricas

Desarrolla un método llamado CalcularArea que acepte un número variable de figuras geométricas (círculos, rectángulos, etc.) usando params object[]. El método debe calcular el área de cada figura y mostrarla.

4. Suma y multiplicación de dos números usando ref y out

Crea un método llamado CalcularOperaciones que acepte dos números por referencia usando ref y que también devuelva la suma y la multiplicación de esos números usando out.

5. Determinar si un número es par o impar usando out

Crea un método llamado EsPar que acepte un número entero y use out para devolver un valor booleano que indique si el número es par o impar.