



Ejercicios

Polimorfismo y clases abstractas

Ejercicios de polimorfismo y clases abstractas

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".

101. Sobre-sobrescribiendo esas advertencias

Crear un proyecto de biblioteca de clases contenga la clase Sobrescrito.

1. Sobrescribir el método ToString para que retorne *"¡Este es mi método ToString sobrescrito!"*.
2. Sobrescribir el método Equals para que retorne *true* si son del mismo tipo (objetos de la misma clase), false caso contrario.
3. Sobrescribir el método GetHashCode para que retorne el número *1142510181*.
4. Agregar en el método Main el siguiente código:

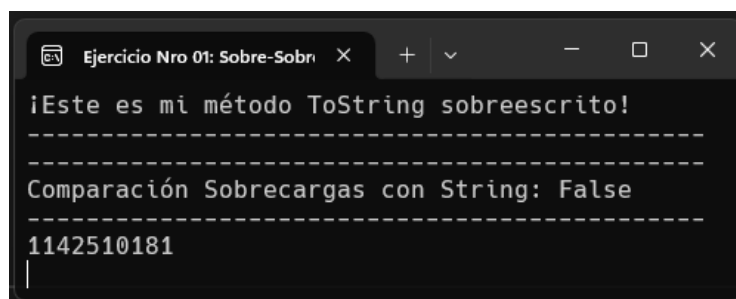
```
Console.Title = "Ejercicio Nro 01: Sobre-Sobrescrito";
Sobrescrito objetoSobrescrito = new Sobrescrito();

Console.WriteLine(objetoSobrescrito.ToString());

string objeto = "¡Este es mi método ToString sobrescrito!";

Console.WriteLine("-----");
Console.WriteLine("Comparación Sobrecargas con String: ");
Console.WriteLine(objetoSobrescrito.Equals(objeto));
Console.WriteLine("-----");

Console.WriteLine(objetoSobrescrito.GetHashCode());
Console.ReadKey();
```



```
Ejercicio Nro 01: Sobre-Sobri
¡Este es mi método ToString sobreescrito!
-----
Comparación Sobrecargas con String: False
-----
1142510181
```

Luego agrega las siguientes condiciones:

1. Agregar a la clase Sobrescrito un atributo `miAtributo` del tipo `string`, con visibilidad `protected`.
2. Generar un constructor de instancia que inicialice `miAtributo` con el valor *"Probar abstractos"*.
3. Agregará propiedad abstracta `MiPropiedad` de sólo lectura. Una vez implementada, retornará el valor de `miAtributo`.
4. Crear un método abstracto `MiMetodo` que retorne un `string`. Una vez implementada, retornará el valor de `MiPropiedad`.
5. Agregar una clase llamada `SobreSobrescrito` que herede de `Sobrescrito`. Implementar el código necesario para que todo funcione correctamente.
6. Modificar el método `Main` para probar las modificaciones

102. Calculadora de formas

1. Crear una biblioteca de clases que contenga la siguiente jerarquía de herencia:
 - a. La clase `Figura` será base de las demás y no deberá poder instanciarse.
 - b. Las clases `Rectangulo` y `Circulo` heredarán de `Figura`.
 - c. La clase `Cuadrado` heredará de `Rectangulo`.
 - d. Las clases `Circulo` y `Cuadrado` no se deben poder heredar (ser base).
2. Agregar el método `Dibujar` que tendrá una implementación por defecto en la clase base y retornará el texto *"Dibujando forma..."*. Las clases derivadas directas invalidarán dicha implementación y la cambiarán por *"Dibujando círculo"*, *"Dibujando rectángulo"*, según corresponda.
3. Agregar el método `CalcularSuperficie` que deberá ser implementado de manera obligatoria por las clases derivadas. Debe retornar un *double* con el resultado del cálculo de superficie (área) para la figura correspondiente.
4. Agregar el método `CalcularPerimetro` que deberá ser implementado de manera obligatoria por las clases derivadas. Debe retornar un *double* con el resultado del cálculo del perímetro para la figura correspondiente.
5. Crear un proyecto de consola y en el método `Main`:
 - a. Crear una lista de figuras.
 - b. Agregar a la lista una figura de cada tipo.
 - c. Recorrer la lista mostrando:
 - i. El resultado del método `Dibujar`.
 - ii. El área y el perímetro de cada figura con una precisión de 2 decimales.
 - iii. El tipo del objeto (método `GetType`).

```
===== FIGURA 01 =====
Tipo: Biblioteca.Circulo
Dibujando Círculo...
Área: 12,57
Perímetro: 12,57
=====
```

```
===== FIGURA 02 =====
Tipo: Biblioteca.Cuadrado
Dibujando Cuadrado...
Área: 9,00
```

Perímetro: 12,00

=====

===== FIGURA 03 =====

Tipo: Biblioteca.Rectangulo

Dibujando Rectángulo...

Área: 32,00

Perímetro: 24,00

=====

