

Clase 5 - estáticos

lunes, 3 de abril de 2023

17:05

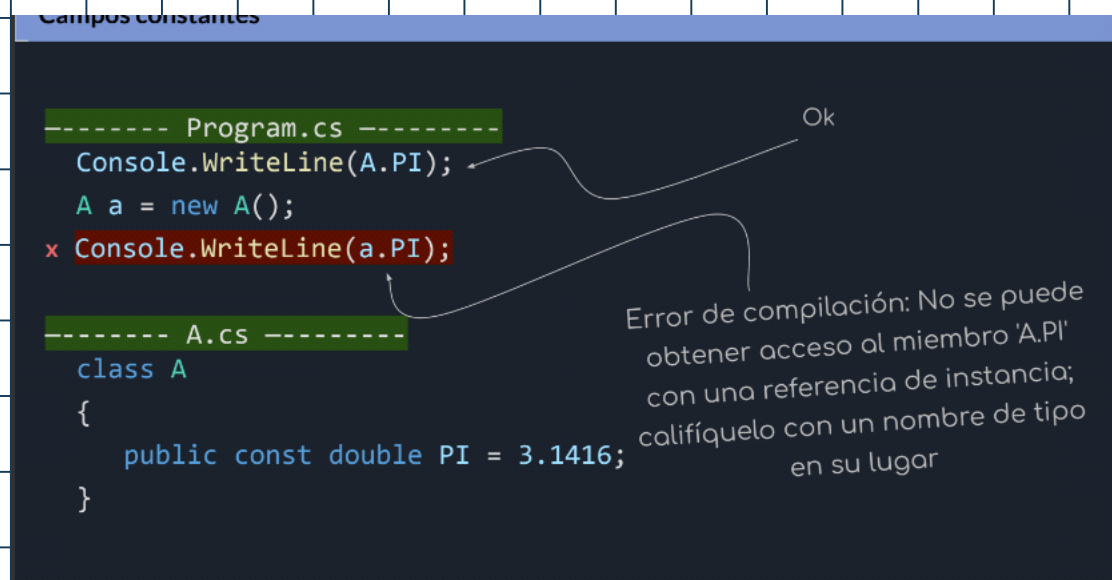
Constructores estáticos: no se usan mucho, no se pueden sobrecargar, solo pueden tener miembros estáticos.

Const. Estáticos se ejecutan cuando se crea la clase. No la instancia, la clase. Se ejecuta una sola vez en la corrida del programa, no sabemos cuando aunque probablemente antes de que llamemos a la clase por primera vez

Clases estáticas: solo tienen miembros estáticos. Agrupan utilidades y datos relacionados. Tienen cosas genéricas, no son para mi app sino que sirven para muchos proyectos

Constantes: se definen como miembros de instancia pero implícitamente son static. En c# mucho no se usan porque están limitadas pero son más rápido. Constantes no son de instancia, son estáticas.

Const las resuelve el compilador antes de ejecutar. Como la hace el compilador, no puede involucrar otras variables.



Campos readonly: no les puedo cambiar(mentira, se puede pero dentro de un constructor) valor pero se definen en ejecución así que puedo hacer más cosas. Una vez que el obj está creado no puedo cambiar la variable.

ENCAPSULAMIENTO:

Propiedades: en .net no se suelen usar getters y setters. Me permiten que desde afuera de la clase la vea como una var pero desde adentro la codifico con getter y setter

Tipo de la propiedad

Nombre de la propiedad

```

public double Lado
{
    get
    {
        <código para leer el
        valor de la propiedad>
    }
    set
    {
        <código para establecer
        el valor de la propiedad>
    }
}

```

Se debe devolver un objeto del tipo de la propiedad

Se recibe un valor en un parámetro implícito llamado `value` (mismo tipo que la propiedad)

58

Código que utiliza la clase Cuadrado

```

...
Cuadrado c = new Cuadrado();
c.Lado = 2.5;
double lado = c.Lado;
...

```

private double _lado

Clase Cuadrado

```

public double Lado
{
    set
    {
        _lado = value;
    }
    get
    {
        return _lado;
    }
}

```

- Una propiedad que implementa sólo el bloque `get` es una propiedad de sólo lectura.
- Una propiedad que implementa sólo el bloque `set` es una propiedad de sólo escritura.
- No es aconsejable el uso de propiedades de sólo escritura. Si la intención es desencadenar algún efecto secundario cuando se asigna el valor, es preferible usar un método en lugar de una propiedad.

preferible usar un método en lugar de una propiedad.

Las propiedades públicas siempre son preferibles a los campos públicos porque, al ser miembros de funciones (no de datos como los campos), pueden procesar la entrada y la salida lo que permite establecer controles sobre los valores de la propiedad.