

Tecnología y paradigmas de programación. Laboratorio 3.

Grupos 1, 2 y 3.

Teniendo en cuenta el proyecto `interfaces` dentro de la solución `inheritance`. `polymorphism` y partiendo de este Interfaz:

```
namespace Intervalos
{
    /// <summary>
    /// Devuelve el tamaño: ancho para intervalo, área para barra, ...
    /// </summary>
    public interface IMedible
    {
        double? Tamaño();
    }
}
```

Implementar las siguientes clases que derivan del mismo. Usar asertos para verificar postcondiciones e invariantes. Usar excepciones para tratar las precondiciones y manejarlas adecuadamente, ver solución `exceptions`.

Intervalo

- Dos atributos de tipo `double` anulables, los extremos del intervalo (ver `nullable` dentro de `generics`). Las propiedades que se necesiten.
- Constructor por defecto, asigna `null` a los dos atributos (intervalo vacío). Constructor a partir de dos `double` anulables.
- Método privado `Check`, para chequear el invariante de la clase. El extremo izquierdo del intervalo tiene que ser menor que el derecho, salvo si los dos son `null`, entonces representa es el intervalo vacío. No puede ocurrir que uno sea `null` y el otro no.
- `Tamaño`, devuelve un `double` anulable, la diferencia de los atributos. Si es el intervalo vacío, devuelve `null`. Ver `Compare` en `interfaces`, dentro de `inheritance.polymorphism`.
- `ToString`.
- [Equals](#) y [GetHashCode](#).
- Opcional, elegir: operador `*` o método intersección, devuelve la intersección de dos Intervalo.

Barra: Deriva de Intervalo, representa una barra, como la de un gráfico de barras.

- Un atributo, `alto`, de tipo `double`. Tiene que ser mayor o igual que cero. Las propiedades que se necesiten.
- Constructor por defecto, asigna 0 al atributo `alto`. Constructor a partir de tres reales.
- Método privado `Check`, para chequear el invariante de la clase. Se puede implementar usando `Check` de Intervalo mediante `base`.
- `ToString`.
- `Equals` y `GetHashCode`.
- `Tamaño`, devuelve el área de la barra, puede implementarse llamando a `Tamaño` de `base`.
- Opcional, elegir: operador `*` o método intersección, devuelve la intersección de dos barras, definida como la intersección de los Intervalo de cada Barra y el mínimo de los atributos `alto`.

Escribir una aplicación de consola con un método polimórfico `AnchoMayor` que reciba dos `IMedible` y devuelva una referencia al de mayor `Tamaño`. Probar los métodos y operadores implementados.

NOTA: entregar la tarea autónoma antes del siguiente laboratorio.