Genéricos

Ejercicio 1

Partiendo de la siguiente definición de clase parametrizada…

```
class A<T, U> {
  private T clave;
  private U valor;
  …
}

class Program {
  public static void Main()
  {
        …
  }
}
```

Nota: Sin usar el código "autogenerado" por el IDE.

- Define un constructor que reciba los dos atributos como parámetro.
- Crea 2 propiedades que te permitirán devolver los dos atributos.
- Prueba la clase en em Main con una clave de tipo entero y un valor de tipo cadena.

Ejercicio 2

- Crea una clase estática genérica llamada **Comparador<T>** que posea a su vez dos métodos de utilidad estáticos llamados **Mayor** y **Menor**.
- Ambos reccibirán dos parámetros del tipo genérico, y devolverán true o false en el caso que el primer parámetro sea mayor que el segundo y vicenversa.
- ¿Qué problemas has encontrado?.
- La mejor forma de solucionarlo es obligando a que el parámetro genérico implemete de **IComparable<T>**.
- Crea una clase programa que te permita probar estos métodos, mandándole diferentes elementos, enteros, strings, float, etc.

Ejercicio 3

- Partiendo del ejercicio anterior. Crea una clase Persona la cual tendrá solo dos atributos que son Nombre y Edad.
- Comprueba si funcionan los métodos Mayor y Menor con ella. ¿Qué ocurre?.
- Ahora haz que la clase derive de **IComparable<Persona>** y de **ICloneable** y que invalide ToString().
- Haz lo necesario para comprobar el funcionamiento de todo lo que has añadido.

Ejercicio 4

- Crea una clase genérica **ParOrdenado<T>** con dos atributos denominados primero y segundo
- Crea el correspondiente constructor, propiedades y el método ToString
- Además, añade un indizador de solo lectura que dependiend de si el índice es 0 te devolverá el valor de atributo primero y si es 1 el valor de atributo segundo (Una excepción en cualquier otro caso).
- Crea un programa que te permita probar esta clase usando enteros y cadenas.

Ejercicio 5

- A partir de la clase del ejercicio 4 , vamos a crear una clase NumeroComplejo que derivare de **ParOrdenado<double>**.
- Como ya sabemos, los números complejos constan de dos partes una real y una imaginaria (compuesta por el número y el sufijo i), por eso vamos a utilizar la superclase ParOrdenado que ya posee los dos elementos que necesitamos.
- Tendremos un constructor al que le pasaremos un string con la forma binomial de un número complejo y se encargará de comprobarlo con una expresión regular.
- Además, también tendrás que redefinir los operadores +, - , *, ==, !=, el cast explícito y el método ToString que te devolverá el número de forma correcta con el sufijo y el signo + añadido.
- Crea los elementos necesarios para que la clase quede completa y el programa para probar su funcionamiento.

Ayuda

Un número complejo se representa en forma binomial como: $z=a+bi$

Las operaciones que se piden en el programa son…

- Suma: $(a,b)+(c,d)=(a+c,b+d)$
- Multiplicación: $(a,b)\cdot(c,d)=(a\cdot c-b\cdot d,a\cdot d-c\cdot b)$
- Igualdad: $(a,b)=(c,d)\Leftrightarrow a=c\wedge b=d$
- Resta: $(a,b)-(c,d)=(a-c,b-d)$