TEMAS:

excepciones, test, generics, interfaces, archivos, bd, delegados, hilos, extensión

1.- Crear una clase genérica que contenga un atributo de tipo List<T> y otro de tipo U, ambos protected.

Realizar sobre la clase: propiedades (l/e), constructor por defecto + sobrecarga (1 parametro U), métodos Add y Remove y polimorfismo sobre ToString.

2.- Agregar interface para: Serializar y deserializar en xml con bool Serializar(string) y bool Deserializar(string, out Clase<T,U>) en la clase del punto anterior.

3.- Generar excepción propia para administrar posibles errores en el punto anterior.

4.- Crear la bd bd-modelo-sp y ejecutar el script, luego realizar un crud en la entidad correspondiente.

USE [bd-modelo-sp]

GO

CREATE TABLE [dbo].[usuario](

[id] [int] IDENTITY(1,1) NOT NULL,

[nombre] [varchar](100) NOT NULL,

[edad] [int] NOT NULL

) ON [PRIMARY]

GO

5.- Agregar interface para: Serializar y deserializar en json con bool SerializarJson(string) y bool DeserializarJson(string, out Clase), a la clase del punto anterior.

6.- Realizar una clase que posea un evento, el cuál será disparado cuando se invoque al ToString. Diseñar dicho evento según las convenciones.

7.- Diseñar un manejador de eventos que capture el evento anterior y guarde en un archivo de texto (\*.log) la fecha, con hora, minutos y segundos y, en un renglón distinto, el contenido del ToString.

8.- Generar un método de extensión (ObtenerTodos) para la clase del punto 4, que retorne un List<Clase>

9.- Diseñar un método que, en segundo plano, recupere toda la información del archivo (\*.log) del punto 7 y lo muestre.

10.- Realizar, al menos dos Test Method, para testear la clase del punto 1.