Clases abstractas:

* Pueden contener miembros virtuales y abstractos.
* Pueden contener miembros de instancia.

Si se define un método abstracto:

* La clase derivada debe utilizar la palabra reservada *override* para implementar el método abstracto.
* Por definición los métodos abstractos son virtuales, pero no pueden estar marcados con *virtual* de forma explícita.

Una clase sellada que hereda de una clase abstracta:

* Sólo debe implementar los métodos abstractos.

Concepto de Polimorfismo:

* Tiende a existir en la relaciones de herencia.
* Es la propiedad que tienen los objetos de invocar genéricamente un comportamiento.
* La invocación es resuelta en tiempo de ejecución.

Programación orientada a objetos (POO):

* Es una manera de construir sofware basada en un nuevo paradigma.
* Propone resolver problemas de la realidad a través de identificar objetos y relaciones de colaboración entre ellos.
* El Objeto y el Mensaje son sus elementos fundamentales.

Un método virtual NO puede crearse cuando:

* La clase está definida como *sealed*.

Un constructor no puede ser declarado como *virtual*.

Una propiedad definida como virtual en una clase:

* Debe tener implementación en la clase que se definió.

Interfaces:

* Por definición todos los miembros son abstractos, pero no pueden llevar la palabra *abstract*.
* Sin atributos.
* Todos los miembros son públicos, sin decir *public*.
* Puede tener propiedades sin implementación.
* Implícito: public void guardar(){} / Explícito: void IDatos.guardar(){}.

Métodos de extensión:

* Son métodos estáticos que se usan como métodos de instancia en las clases donde se agregan.
* Se definen como estáticos en clase estáticas.

Palabra *new*:

* Lo único que hace es adquirir memoria binaria sin inicializar.
* No participa de ninguna manera en la inicialización de la memoria.

Firma del método:

* Nombre método, tipo y cantidad de parámetros, modificador de parámetro (out o ref).
* NO: nombre de parámetros, tipo de retorno del método.

Arrays:

* Implementa IEnumerable, se puede usar *foreach*.
* Son todos dinámicos.
* Int[] unidimensional. Int[ , ] multidimensional.

Generics:

* Aumenta seguridad de tipo de datos.
* Aumenta la reutilización de código.
* Aumenta la performance (boxing/unboxing).

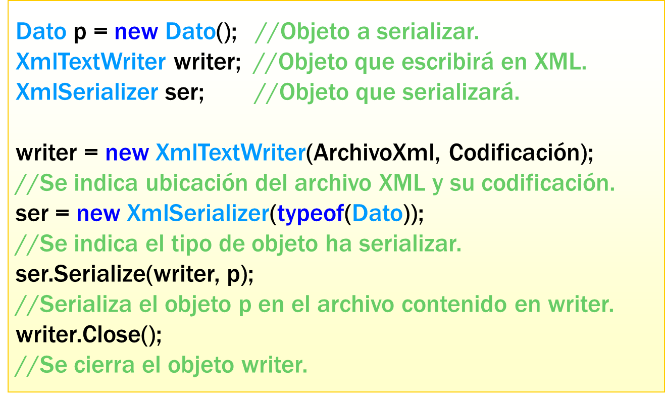
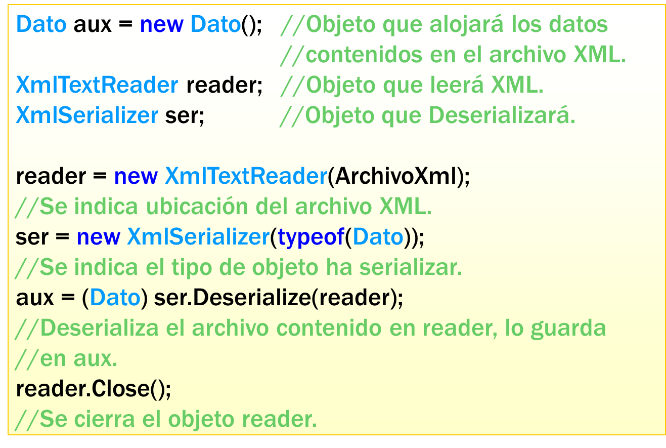
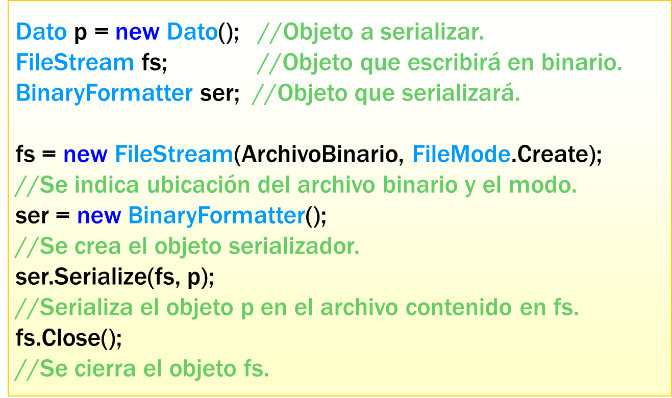
Excepciones:

* Clase exception. Mensaje significativo.
* Catch general debe ser el último bloque catch en el programa.

Archivos de texto:

* StreamWriter (string path, bool append). WriteLine(string value).
* StreamReader (string path). ReadToEnd().

Serialización:

* Es el proceso de convertir un objeto en memoria en una secuencia lineal de bytes.
* XML:
  + Atributos públicos y propiedades de lectoescritura.
  + Tienen que tener constructor por defecto.
  + 
  + 
* Binaria:
  + Puede serializar atributos públicos y privados.
  + Tiene que tener constructor por defecto.
  + 
  + 