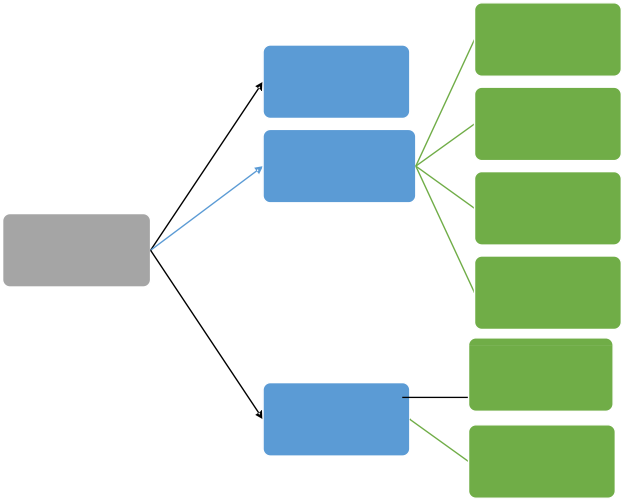
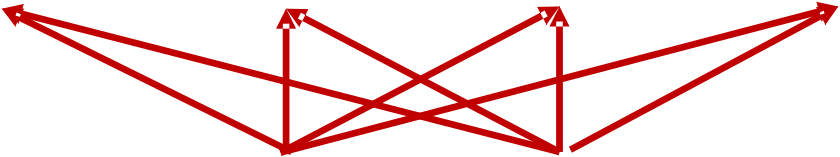
**Evaluation Only. Created with Aspose.Words. Copyright 2003-2021 Aspose Pty Ltd.**

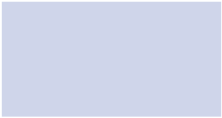
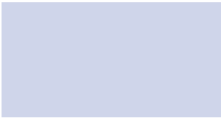


**Animal Animales Organismo Pajaro**

**void correr() void multicelular() void respirar() void volar() void viviparo() void sangreCaliente() void mover() void oviparo()**



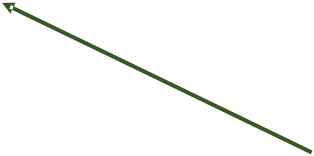
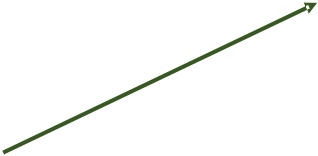
**Perro Águila**



**Interfaces**

**Animales**

**void multicelular() void sangreCaliente()**

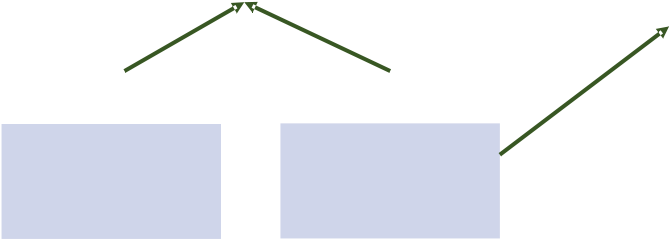
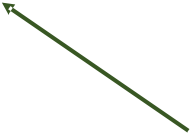


**Organismo Animal Pajaro**

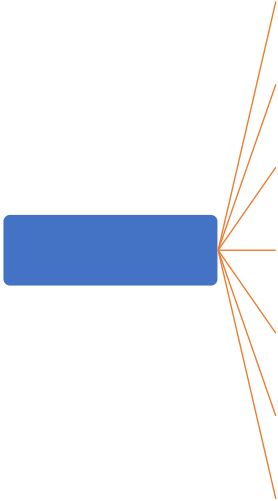
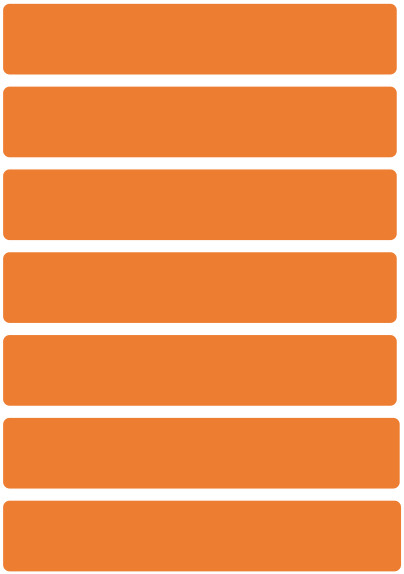
**void respirar()**

**void correr() void mover() void volar() void viviparo() void oviparo()**

**Perro Águila**

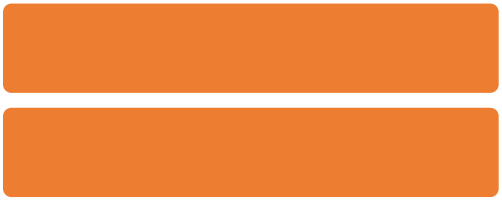


**Interfaces**



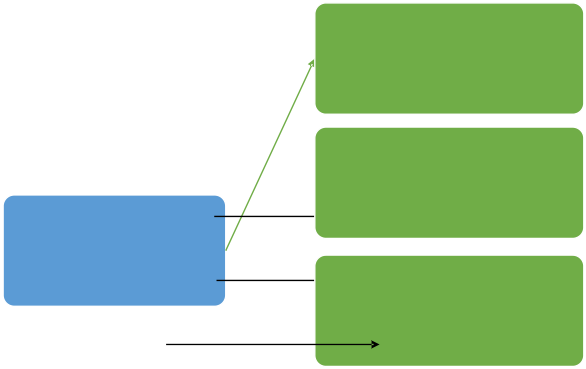
**Interfaces**

|  |
| --- |
|  |
|  |

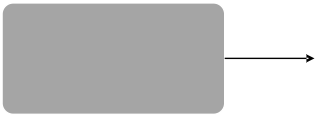


|  |
| --- |
|  |
|  |

**Tiempo**



**de Ejecución**



**Ejemplo de Delegado**

**public delegate** int MiDelegado **(**string **s);**

**Declaración de Delegado**

**delegate <valor retorno> <nombre delegado> <lista parámetros>**

**Ejemplo Inicialización Delegado**

**public delegate void imprimirCadena(string s);**

**...**

**imprimirCadena ic1 = new imprimirCadena(MetodoEscribirPantalla); imprimirCadena ic2 = new imprimirCadena(MetodoEscribirArchivo);**

**Métodos Anónimos**

**Ejemplo de Método Anónimos**

**delegate void Numeros(int n);**

**...**

**Numeros num = delegate(int x) {**

**Console.WriteLine("Método Anónimo: {0}", x); };**

**Declaración de Delegado**

**delegate <valor retorno> <nombre delegado> (<lista parámetros>); …**

**<nombre delegado> = delegate(<lista parámetros>)**

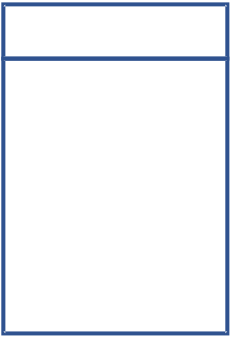
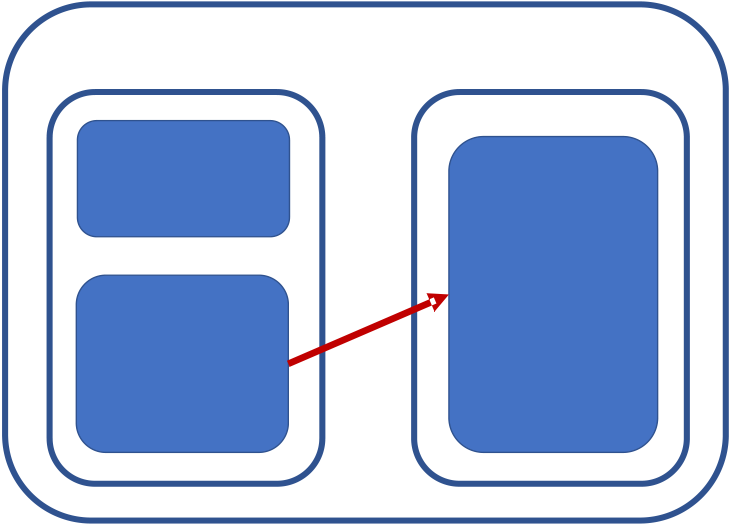
**{**

**Método Anónimo**

**};**

**MEMORIA**

**Stack Heap Libros**



**+ Título**

**+ Autor + Categoría**

**+ id**

**Definición de Estructura**

**Palabra clave: struct**

**struct** Libros **{**

public string titulo;

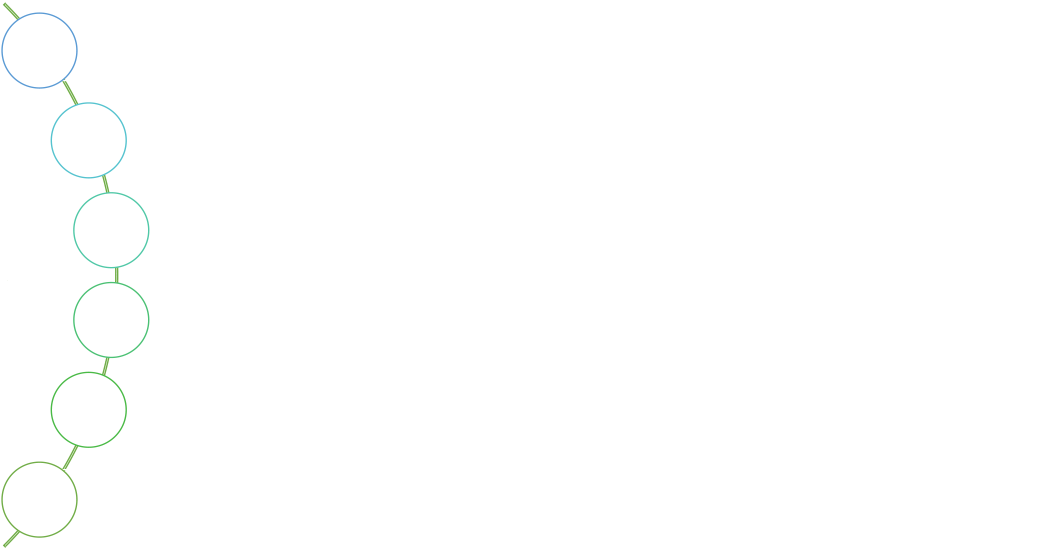
public string autor;

public string categoria; public int libro\_id;

**};**

**Estructuras - Características**

Pueden tener métodos, campos, indexadores, propiedades, métodos de operador y eventos



Pueden tener constructores definidos, pero no destructores

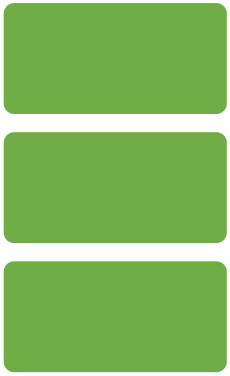
No pueden Heredar otras Estructuras o Clases

Pueden Implementar Interfaces

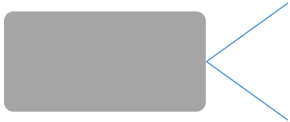
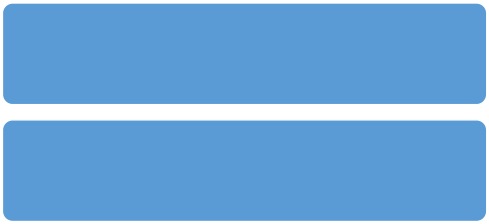
No Abstract, virtual o protected

No es Obligatorio el uso del Operador New

|  |
| --- |
|  |
|  |

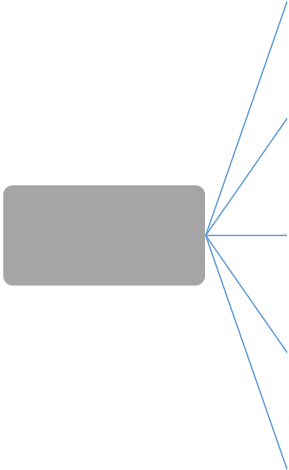
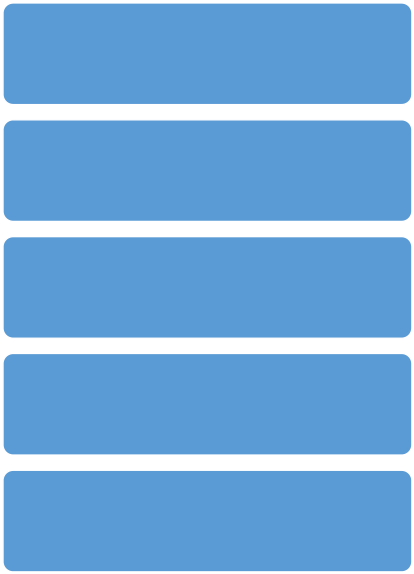


**Eventos – Publisher - Suscriptor**

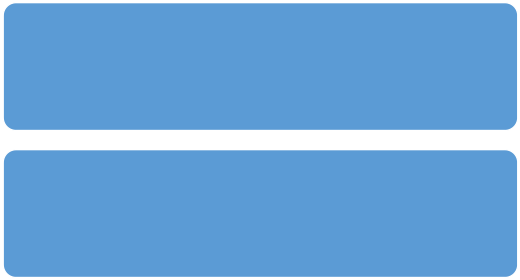


|  |
| --- |
|  |
|  |
|  |

**Eventos - Características**

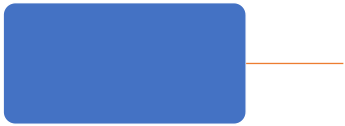
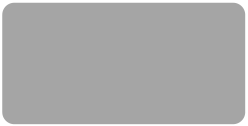
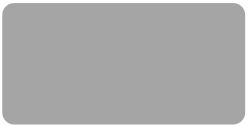
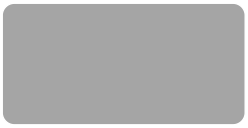


|  |
| --- |
|  |
|  |



|  |
| --- |
|  |
|  |

**Espacios de Nombres**



**Declaración del Espacio de Nombres**

**namespace** nombre\_espacio\_de\_nombres **{ namespace** espacioNombres1 **{**

**class** clase **{**

**// código fuente public void** funcion**() {**

**Console.WriteLine(“**Prueba Espacio de Nombres**"); } }**

**}**

**}**

**Invocar Espacio de Nombres**

**nombre\_espacio\_de\_nombres.nombre\_elemento\_espacio\_de\_nombres espacioNombres1.clase**

**Directiva Using**

**using** System**; …**

**…**

**Console.WriteLine(“**Hola Mundo**”);**

**Sin Directiva Using**

**System.Console.WriteLine(“**Hola Mundo**”);**

**Declaración Clase Abstracta**

**public abstract class Hablar**

**{**

**// código fuente**

**{**

**Declarando Método Clase Abstracta**

**public abstract void charlar();**

**Derivando la Clase Abstracta**

**public class Conversacion : Hablar**

**{**

**// código fuente**

**{**

**This document was truncated here because it was created in the Evaluation Mode.**