**Laboratorio II - Tema 1**

**Trabajo práctico I - Programación Orientada a Objetos: Polimorfismo por Herencia**

**Consigna:** analizar la siguiente porción de un Diagrama UML donde se observa una jerarquía de clases relacionadas por Herencia (continúa luego del diagrama).

|  |
| --- |
| **Vehículo** |
| //Miembros atributos  Atributo 1  Atributo 2  Atributo 3 |
| //Miembros métodos  Constructor 1() //por defecto  Constructor 2() //sobrecargado  setAtribto1(argumento): tipo de dato  getAtrinuto1(): tipo de dato  setAtribto2(argumento): tipo de dato  getAtrinuto2(): tipo de dato  setAtribto3(argumento): tipo de dato  getAtributo3(): tipo de dato  método que se redefinirá en las clases herederas () |

|  |
| --- |
| **Automóvil** |
| //Miembros atributos  Atributo 1  Atributo 2 |
| //Miembros métodos  Constructor 1() //por defecto  Constructor 2() // llama al constructor de la clase padre  setAtribto1(argumento): tipo de dato  getAtrinuto1(): tipo de dato  setAtribto2(argumento): tipo de dato  getAtrinuto2(): tipo de dato  método heredado que se redefine en esta clase() |

**Actividades a realizar:**

* **Abstracción / generalización:** crear un diagrama dentro de lucidchart con la estructura propuesta y completar los atributos y los métodos correspondientes en cada clase.
* **Encapsulamiento/ocultamiento:** asignar los modificadores de acceso o de visibilidad a los atributos y los métodos.

+ public - private #protected

* Codificar en C# las 2 clases en un proyecto de consola dentro del IDE de Visual Studio.
* Crear en la clase principal dos objetos de la clase heredera con sus respectivos datos y ejecutar el programa.

**Condiciones de presentación:**

El trabajo se presentará en el aula de Classroom (1 trabajo por grupo).

El proyecto de Visual C# debe subirse comprimido.

En la clase main del proyecto escribir como comentario los integrantes del grupo y el número, incluyendo además el enlace del modelo creado en lucidchart.