Clase #2

Programación orientada a objetos

* Herencia
* Encapsulamiento
* Abstracción
* Polimorfismo

**Herencia:** Heredar características de una clase padre y así sucesivamente llegar a una clase Object.

|  |
| --- |
| Object |
|  |

Clase Padre

Clase abstracta  
4 (todas excepto las de abajo)

|  |
| --- |
| Instrumento |
| * Color: String * Tamaño: Double * Código: Integer * Precio: Double * Tipo de material: String |

Clase hija Clase hija

|  |
| --- |
| Viento |
| * Tipo de boquilla: String |

|  |
| --- |
| Percusión |
| * Tipo de baqueta |

|  |
| --- |
| Trompeta |
| \* # botones:  Integer |

|  |
| --- |
| Flauta |
| \* # agujeros:  Integer |

|  |
| --- |
| Cimbal |
| \* # cuerdas:  Integer |

|  |
| --- |
| Tambor |
| \* Tipo de parche:  String |

**Abstracción:** Observar o analizar las características en común entre una o más clases.

**Encapsulamiento:** Es una característica que nos permite el control de nuestras propiedades. (Tipo s de datos) (Get / Set)

Set = Asignar (Asignar un valor a una propiedad), en términos de programación se conoce como “modificar”

Get = Obtener (Es obtener o retornar el valor que s ele fue asignado a la propiedad en el Set), En términos de programación se conoce como “Accesor”

**Polimorfismo:** Las clases hijas pueden actuar de la forma que se necesite dependiendo las clases de las que provengan.

**NameSpace:** Clasifican las clases para poderlas utilizarlas más delante de forma clasificada.

NOTA: Todas las propiedades son privadas y las propiedades son públicas.

Instancia: Reservación de memoria de un objeto