* ¿Qué pasaría si ahora queremos agregar un pato de madera?

|  |
| --- |
|  |

* Nuestro diagrama de clases quedaría así:

|  |  |
| --- | --- |
|  | * Los patos de madera no emiten sonidos. * Los patos de manera no vuelan. * ¿Cómo resolvemos esto? * Sobre escribimos los dos métodos: **sonido** y **volar** * Dentro de cada método no escribimos ninguna línea, dejamos vacío el método. |

* El código de la clase **Pato**:

|  |
| --- |
|  |
|  |

* El código de la clase **PatoDeMadera** queda:

|  |  |
| --- | --- |
|  | * Los patos de madera no emiten sonidos. * Los patos de manera no vuelan. * ¿Cómo resolvemos esto? * Sobre escribimos los dos métodos: **sonido** y **volar** * Dentro de cada método no escribimos ninguna línea, dejamos vacío el método. |

* El código de la clase principal **Simulador** queda:

|  |
| --- |
|  |

* Llamamos al método **mostrar en pantalla** y se muestra ## PATO DE MADERA ##
* Llamamos al método **nadar** y se muestra :: El pato esta nadando ::
* Aunque llamamos al método **sonido** y **volar**, dichos métodos no realizan ninguna acción. No muestran nada.
* ¿Cuáles son las desventajas de utilizar herencia?
* **El código se duplica.**
* El método **sonido** se duplica en las clases **Pato**, **PatoDeGoma** y **PatoDeMadera**

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

* El método **volar** se duplica en **Pato**, **PatoDeGoma** y **PatoDeMadera**

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

* **Los cambios en la super clase afectan a las sub clases.**
* Cuando agregamos el método volar a la super clase Pato, automáticamente ese método puede ser utilizado por sus subclases.
* Algunas sub clases no deberían tener esta característica. Por ejemplo: un pato de goma y un pato de madera no vuelan.