* Últimamente la compañía ha estado bajo presión de los competidores.
* Los ejecutivos de la compañía piensan que es tiempo de una gran innovación.
* Necesitan algo realmente impresionante para mostrar a los accionistas en su próxima reunión.
* Los ejecutivos deciden que el simulador debe de lograr que los patos puedan **volar** y con esto tener ventaja sobre los competidores.
* Joe les dice que no hay ningún problema, después de todo es un buen programador, ¿Qué tan difícil puede ser?
* **Joe:** solo necesito agregar el método volar a la super clase Pato y después todos las sub clases heredaran el método **volar()**.

|  |
| --- |
|  |

* Agreguemos un tipo de pato más, agreguemos un pato de goma.

|  |
| --- |
|  |

* El código de la clase **Pato** queda

|  |
| --- |
|  |

* El código de **PatoReal**, **PatoPelirojo** y **PatoDeGoma** quedan:

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

* El código del **Simulador**:

|  |
| --- |
|  |

* Todo parece funcionar bien, pero hay 2 pequeños detalles.
* **Los patos de goma no hacen quack** (los patos de goma emiten chirridos)
* **Los patos de goma no vuelan.**
* Los accionistas probaron el simulador y había patos de goma volando por la pantalla.
* ¿Qué paso?
* **1ro: a Joe se le paso que no todos los patos emiten el mismo sonido**
* **2do: a Joe se le paso que no todos los patos pueden volar.**
* Solucionemos el primer problema.
* Los patos de goma no hacen quack, los patos de goma emiten chirridos.
* ¿Cómo solucionamos este problema?
* Sobre escribimos el método **sonido()** y cambiamos el mensaje con el tipo de sonido que hacen los patos de goma.

|  |
| --- |
|  |

* No cambiamos nada en el código del Simulador, solo lo ejecutamos nuevamente.

|  |
| --- |
|  |

* Problema solucionado. Como sobre escribimos el método **sonido()** en la clase PatoDeGoma para que emita un chirrido (“el pato ha chirriado”), ahora se invoca a este método y ya no al método **sonido()** (“el pato ha hecho quack”) de la super clase Pato
* Ahora solucionemos el segundo problema.
* Los patos de goma no vuelan.
* ¿Cómo resolvemos este problema? ¿Qué podemos hacer para que los patos de goma no vuelen?
* Joe piensa en sobre escribir el método **volar()** en la clase **PatoDeGoma.**  Y dentro del mpetodo no escribir nada, dejarlo vacío.

|  |
| --- |
|  |

* El código del **Simulador** no cambia, queda como esta, solo lo ejecutamos.

|  |
| --- |
|  |

* Problema solucionado.
* Aunque llamemos al método volar, no se imprime nada, porque sobre escribimos el método **volar()** en la clase **PatoDeGoma** y dicho método está vacío.
* ¿Qué problemas se presentaron en esta versión?
* El primer detalle que tuvimos fue con el sonido. Los patos de goma no hacen quack.
* Solucionamos este problema sobre escribiendo el método **sonido()** en la clase **PatoDeGoma.** Y dentro del método cambiamos el tipo de sonido que emite el pago de goma, en este caso un chirrido.
* El segundo problema fue que los patos de goma no vuelan.
* Solucionamos este problema sobre escribiendo el método **volar()** en la clase **PatoDeGoma.** Y dentro del método no escribimos ninguna línea de código, lo dejamos vacío.
* El primer problema surgió a la hora de que el pato de goma llama al método sonido, al llamar a esté método se indica que el pato de goma hizo quack, cosa que no es verdad.
* Solucionamos esto sobre escribiendo dicho método y escribiendo el código adecuado para el sonido que emite un pato de goma.
* Cuando Joe agrego el nuevo comportamiento **volar** a la clase **Pato**, también agrego ese comportamiento a todas las sub clases que la heredan.
* El detalle con esto es que este comportamiento no es apropiado para todas las subclases, como por ejemplo este comportamiento no es apropiado para el pato de goma. Un pato de goma no vuela.
* Hasta aquí hemos solucionado estos problemas sobre escribiendo los métodos **sonido** y **volar** en la clase **PatoDeGoma**