
Buenas prácticas en la programación orientada a objetos

Eliminar el if

Cuando trabajamos en el paradigma orientado a objetos debemos respetar la sintaxis “objeto mensaje”. Como Python no es un lenguaje puro de objetos, su sintaxis nos permite hacer cosas que no son propias del paradigma de objetos. Por ejemplo, en Python el if es una sentencia. Esto implica que el lenguaje no es de objetos.

En los lenguajes de clasificación, el if se implementa con polimorfismo. Con lo cual, usar if implica que no estamos usando polimorfismo. Esto implica que tenemos diseños menos mantenibles y además diseños NO orientados a objetos.

¿Cómo sacar el if?

Para sacar el if podemos realizar los siguientes pasos:

1. Crear una jerarquía polimórfica con una abstracción por cada condición.
2. Usando el mismo nombre de mensaje repartir el cuerpo del if en cada abstracción (usar polimorfismo)
3. Nombrar el mensaje del paso anterior.
4. Nombrar las abstracciones.
5. Reemplazar if por envío de mensaje polimórfico.
6. Buscar objeto polimórfico si es necesario.

Conclusiones

Al implementar el if utilizando polimorfismo los objetos son los que toman la decisión. Además, cuando aparece algo nuevo en el dominio de problema, aparece algo nuevo en mi modelo. Esto es lo que se espera en un buen diseño.

Por otro lado, al hacer implementar los if con polimorfismo se puede utilizar meta-programación, permitiendo que el programa pueda referirse sobre sí mismo.

El mecanismo propuesto tiene un límite y es que no se puede sacar el if cuando colaboran objetos de distintos dominios en la condición.

Do Not Copy or Post

Eliminar código repetido

El código repetido refleja la falta de algún objeto que represente el motivo de dicho código. Esto hace que el software sea más difícil de mantener.

Cabe destacar que código repetido no significa “texto repetido”, sino que código repetido significa patrones de colaboraciones repetidas.

Lo que hay que hacer con el código repetido es reificarlo y darle un significado por medio de un nombre.

¿Cómo sacar código repetido?

Para sacar el código repetido podemos realizar los siguientes pasos:

1. Copiar el código repetido a “un lugar”.
2. Parametrizar lo que cambia.
3. ¡Ponerle un nombre!

Muchas veces cuando hay que parametrizar lo que cambia será necesario que uno o más parámetros sean abstracciones de código. Y en Python para esto utilizamos las expresiones lambda.

El paso más complicado del proceso es ponerle un nombre a la función, porque implica entender qué significa esa repetición de código.