

Plantillas y ganchos

las plantillas y los ganchos son una estrategia de programación comúnmente usada para introducir hotspots en los frameworks orientados a objetos. Pueden implementarse utilizando herencia o composición como mecanismos para separar los aspectos comunes de los aspectos variables

el patrón de diseño "método plantilla" documentado en el libro "Patrones de diseño" propone abstraer los pasos de la operación en un método implementado en la clase abstracta y redefinir los pasos variables en sus subclases. La herencia es el mecanismo principal para separar lo que es constante, de lo que varía

el comportamiento constante se encapsula en el método plantilla y se separa del comportamiento variable que se implementa en los ganchos, en las subclases. Si bien el comportamiento constante se separa del variable en métodos diferentes, ambos comportamientos pasan a formar parte de un único objeto.

en resumen, las plantillas y ganchos se pueden usar en un framework para separar lo frozen de lo hotspots. Las plantillas implementan lo que se mantiene constante, los ganchos lo que varía.

se puede usar herencia o composición para separar la plantilla de los ganchos, si uso herencia la plantilla se implementa en una clase abstracta y los ganchos en sus subclases. Si uso composición, un objeto implementa la plantilla y delega la implementación de los ganchos

con *herencia*

- es más simple para casos con pocas alternativas
- al implementar los métodos gancho puedo usar las variables de instancia y todo el comportamiento heredado de la clase abstracta
- si hay muchas variantes o combinaciones, empiezo a tener muchas clases y duplicación de código

con *composición*

- evita la duplicación de código y el creciente número de clases cuando hay muchas alternativas y combinaciones
- al implementar los métodos gancho debo pasar como parámetro todo lo que necesiten
- puedo cambiar el comportamiento en tiempo de ejecución sin mayor dificultad