

Práctica 6 ejercicio 2

Responda las preguntas correspondientes al ejercicio de plantillas y **herencia**:

- ¿Qué debo hacer si aparece una nueva fuente de energía (por ejemplo, paneles solares con baterías)?
debería crear una nueva clase para ese tipo de fuente de energía y heredar los métodos abstractos
¿Cuántas y cuáles clases debo agregar en caso de querer todas las variantes de robots posibles para este nuevo tipo de fuente de energía? 1
- Puedo cambiarle, a un robot existente, el sistema de armas sin tener que instanciar el robot de nuevo? Sí, debería tener algún método para hacerlo
- Dónde almacenaría usted el nivel de carga de la batería? Lo almacenaría en el robot mismo. Qué implicaría eso si antes de disparar el láser hay que garantizar que la fuente de energía puede satisfacer el consumo del arma? Realizar una consulta a la variable

Responda las preguntas correspondientes al ejercicio de plantillas y **composición**

- ¿Qué debo hacer si aparece una nueva fuente de locomoción (por ejemplo, motor con ruedas con tracción 4x4)? Crear una nueva clase y definir los pasos propios a ese tipo de fuente de locomoción ¿Cuántas y cuáles clases debo agregar en caso de querer todas las variantes de robots posibles para este nuevo tipo de sistema de locomoción? Debo agregar la clase correspondiente a la fuente de locomoción
- ¿Puedo cambiarle, a un robot existente, el sistema de armas sin tener que instanciar el robot de nuevo? Sí
- ¿Dónde almacenaría usted el nivel de carga de la batería? ¿Qué implicaría eso si antes de disparar el láser hay que garantizar que la fuente de energía puede satisfacer el consumo del arma? Hay que garantizar que sea pasado como parámetro

Explique las ventajas y desventajas de las dos formas de extensión del framework (herencia y composición).

herencia	composición
<ul style="list-style-type: none">• simple para casos de pocas alternativas o pocas combinaciones• al implementar los métodos gancho puedo usar variables de instancia y todo el comportamiento heredado de la clase abstracta• si hay muchas variables, empiezo a tener muchas clases y duplicar código porque cada vez que hay una nueva variable, tengo que definir una nueva clase	<ul style="list-style-type: none">• evita duplicar código y el creciente número de clases cuando hay varias alternativas porque solo implemento nuevos métodos• puedo cambiar el comportamiento en tiempo de ejecución sin mayor dificultad, caso contrario a herencia que tenía que definir un nuevo objeto

///falta