

Patrones de diseño

Factory Method:

Utilización:

Se ha implementado el patrón de diseño Factory Method en las clases CreadorPersonaje y CreadorConVidaComprable, utilizadas por la clase Tienda.

La clase CreadorPersonaje es análoga a una clase Creator, mientras que CreadorDustin, CreadorMike, CreadorLucas etc. son los ConcreteCreator. En este caso, la clase Product está representada por Personaje, y los ConcreteProduct son Dustin, Mike, Lucas y los demás personajes del jugador.

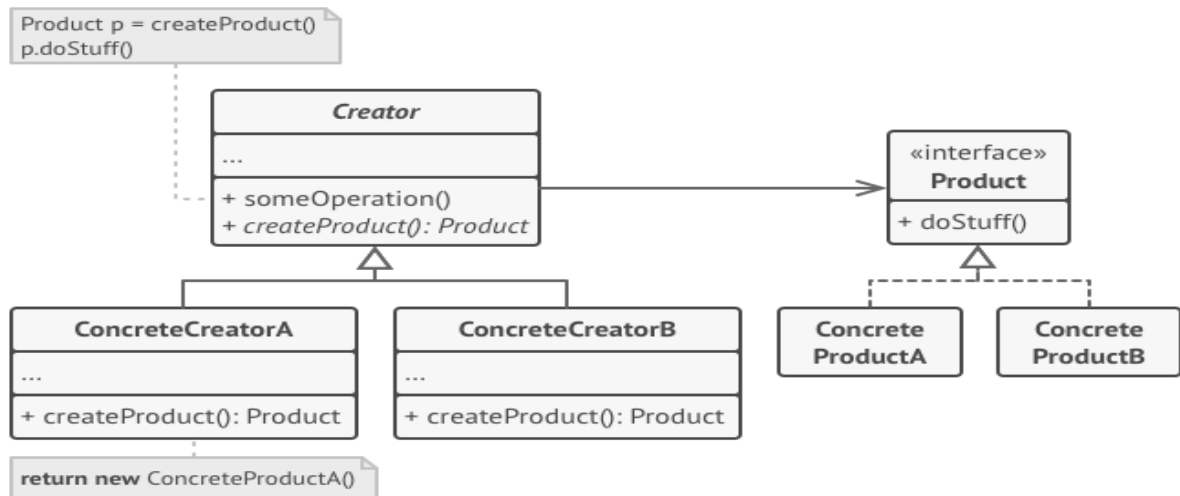
Tienda es la clase cliente que utiliza a los creadores. Implementa el método crearPersonaje delegando a creación del objeto al ConcreteCreator que recibe por parámetro del botón presionado cuando el jugador elige qué comprará.

Beneficios:

El beneficio principal de usar este patrón es poder modelar la creación de un objeto de clase Personaje sin saber de antemano qué clase la instanciará en ejecución. Así, la clase Tienda puede encargarse de la creación de personajes del jugador a partir de qué ConcreteCreator reciba por parámetro, evitándose la necesidad de un instanceof para diferenciar entre las posibles clases concretas que heredan de Personaje.

Además, si en un futuro se quisiera agregar personajes nuevos, sencillamente se tendría que agregar un nuevo ConcreteCreator para cada uno, mientras que la clase Tienda seguirá funcionando sin tener que modificarse.

A continuación se muestra la comparación entre el patrón Factory Method y la implementación usada:



Estructura del Factory Method.¹

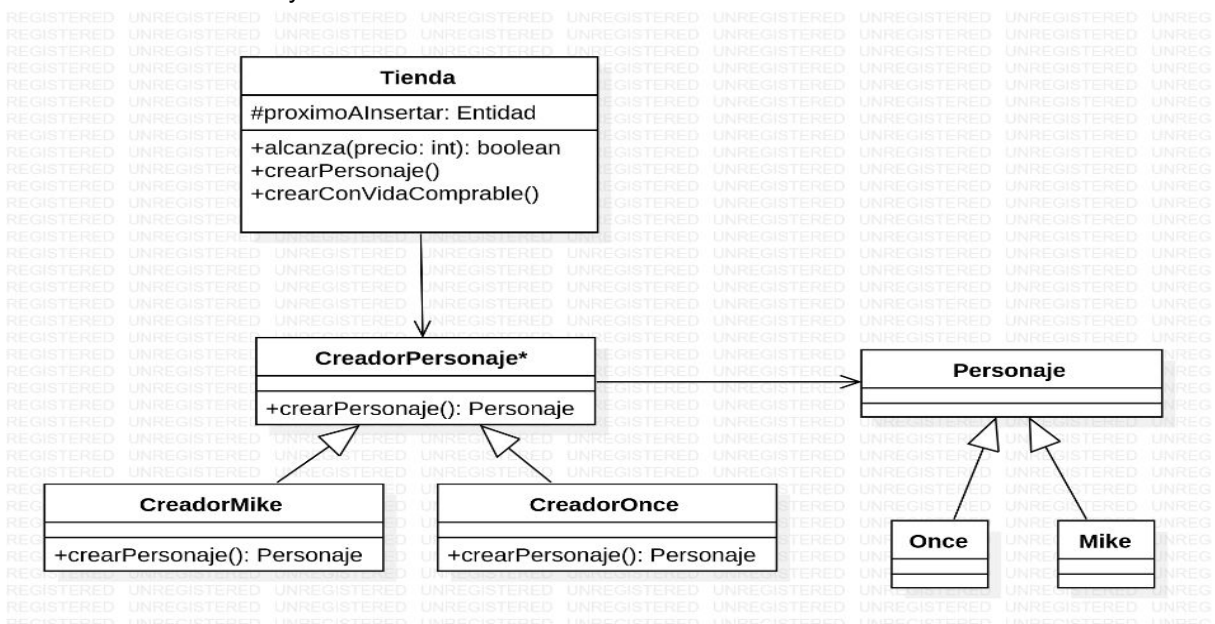


Diagrama de clases de CreadorPersonaje, usando Once y Mike como ejemplos.

Referencias:

1. <https://refactoring.guru/design-patterns/factory-method>