



Práctica 11. Polimorfismo (IV)

Objetivos

- Repaso de conceptos sobre Polimorfismo

Índice

[Índice](#)

[Contexto de la práctica](#)

[Diagrama práctica 10](#)

[Punto de partida](#)

[Ejercicios](#)

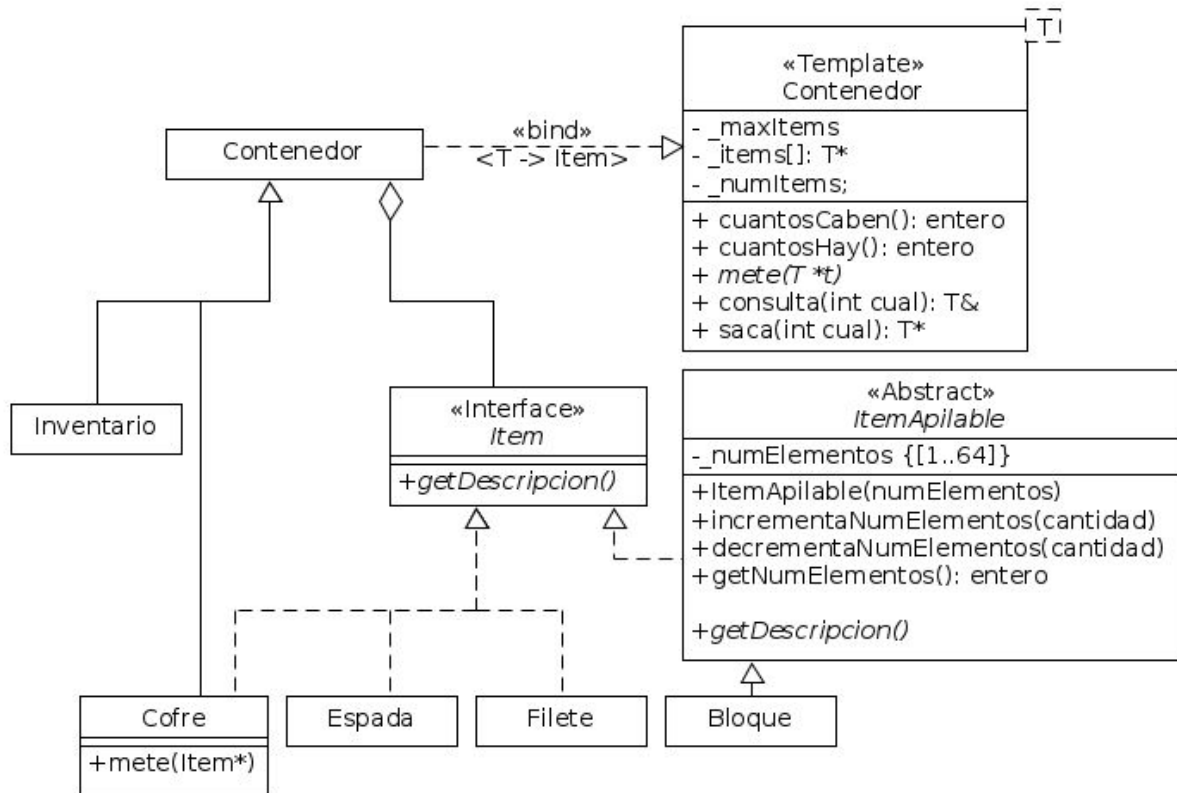
Contexto de la práctica

En esta práctica continuaremos introduciendo cambios sobre el modelo de la práctica anterior para repasar conceptos del Tema 4 relacionados con el Polimorfismo.

Para la realización de los ejercicios propuestos, partiremos del último diagrama de la sesión anterior:



Diagrama práctica 10



Punto de partida

El punto de partida de esta práctica es el programa de prueba con la solución de la práctica anterior.

Ejercicios

- Un **BancoDeTrabajo** es un **Contenedor** de hasta 9 Ítems que permite construir nuevos Items a partir de los materiales que contiene. Crear la **clase BancoDeTrabajo** y añadir un método **construirEspada()** que crea y devuelve un **Item Espada** si el banco contiene al menos tres unidades de un bloque de madera. En caso de construirse el objeto, se deberán actualizar convenientemente las unidades de material que se han utilizado, incluso eliminando el material del contenedor si es necesario. Si no se puede construir la espada se devolverá `nullptr`. *Nota: es necesario crear el ítem apilable **BloqueDeMadera**. Suponemos que en el banco solo puede existir una única instancia de cada tipo de objeto.*
 - Prueba 1:** En la función `main`, crear un banco de trabajo, añadirle un bloque de



- tierra y un bloque de madera con 6 unidades. Construir una espada con el banco y mostrar la descripción del ítem obtenido comprobando que es la de la espada. Consultar también el bloque de madera en el banco (2ª posición) y comprobar que sus unidades se han decrementando convenientemente.
- b. **Prueba 2:** En la función main, crear una segunda espada y comprobar que el número de elementos del banco ha disminuido al haberse destruido el bloque de madera por haberse quedado sin unidades
 - c. **Prueba 3:** En la función main, crear una tercera espada comprobando que se obtiene nullptr ya que no quedan suficientes bloques de madera.
2. Implementar el **constructor de copia de un Contenedor**. Como no sabemos de qué tipo será cada objeto contenido, obligaremos a que cualquier clase cuyos objetos puedan almacenarse en un contenedor implemente el método *Item* copia()* que devuelva una copia del objeto actual. Para ello, añadiremos el método virtual puro *Item* copia()* a la interfaz *Item* y lo redefiniremos en las clases donde sea necesario.
 - a. **Prueba:** En la función main, a partir de algún **cofre con diferentes elementos que haya creado, crear una copia** y visualizar sus elementos para comprobar que el constructor de copia ha tenido éxito
 3. Implementar el **operador de asignación** de un Contenedor.
 - a. **Prueba:** En la función main, crear un nuevo **cofre vacío, asignarle el cofre del ejercicio anterior** y comprobar que el nuevo cofre contiene una copia de los elementos del original
 4. Implementar el método **Contenedor::busca(std::string cadena)** que localiza y devuelve el primer elemento del contenedor igual a la cadena proporcionada. *Nota: implementar el operador==(std::string cadena) en las clases derivadas de Item; este operador comparará la cadena con la descripción del ítem.*
 - a. **Prueba 1:** En la función main, buscar en un cofre algún ítem que tenga una descripción determinada y visualizar su información para comprobar que se ha localizado correctamente
 - b. **Prueba 2:** En la función main, buscar en el cofre una descripción que no se corresponda con ninguno de los elementos contenidos y comprobar que el método busca devuelve nullptr
 5. Una de las modalidades de Minecraft te permite jugar en grupo. Crea la **clase Personaje** con un atributo de tipo string, donde se indique el nombre de usuario, y un inventario con 27 huecos.
 - a. ¿Habría que hacer algún cambio sobre el diagrama UML para reflejar lo descrito anteriormente?
 - b. **Prueba 1:** En la función main crea un contenedor donde se almacene tu personaje y el de algunos amigos
 - c. **Prueba 2:** En la función main, añade varios objetos al inventario de tu personaje y muéstralos por pantalla