Explicación: Factory Method

Hemos usado el patrón de diseño Factory Method para incorporar a un juego la función crear nuevos personajes según su tipo.

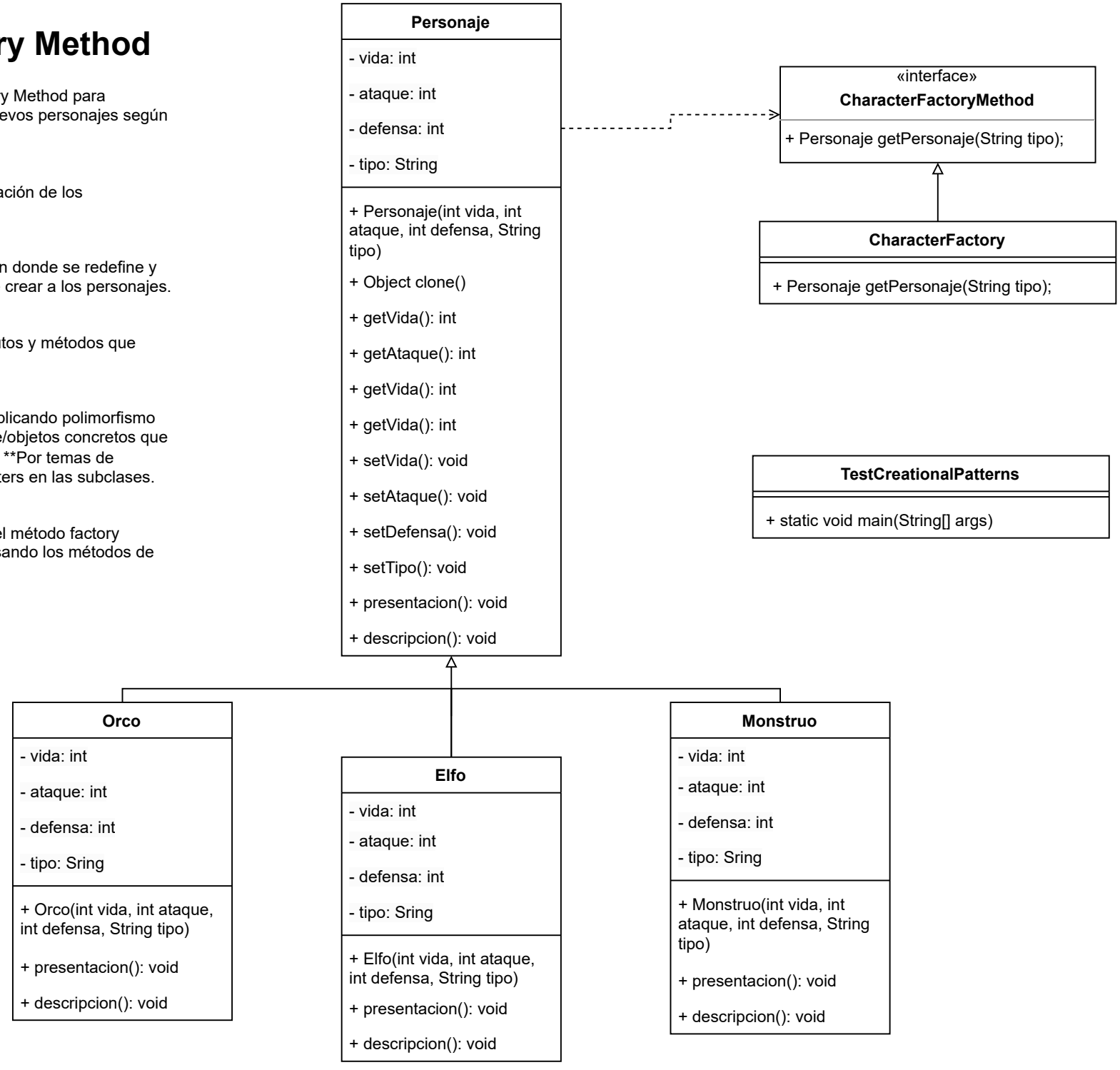
CharacterFactoryMethod
Interfaz que especifica el método de creación de los personajes.

CharacterFactory
Es la clase que implementa la interfaz, en donde se redefine y se determina la lógica para el método de crear a los personajes.

Personaje
Una clase abstracta que define los atributos y métodos que tiene un personaje genérico.

Orco, Elfo, Monstruo
Subclases que heredan de *Personaje*, aplicando polimorfismo personalizamos. Estos son los personaje/objetos concretos que se devuelven en el método de creación. **Por temas de espacio omitimos poner los getters y setters en las subclases.

TestCreationalPatterns
Es donde se ejecuta el programa. Para el método factory generamos un personaje de cada tipo usando los métodos de getPersonaje().



Explicación: Prototype

Con el patrón de Prototype simulamos tener una colección de personajes de un nivel especifico del juego, por lo que solo clonariamos a esos personajes.

CharacterPrototype
Actúa como gestor de prototipos. Es donde creamos la colección de personajes genéricos e implementamos la lógica de clonación.

Personaje
Una clase abstracta que define los atributos y métodos que tiene un personaje genérico. Incluyendo el

Orco, Elfo, Monstruo
Subclases que heredan de *Personaje*, aplicando polimorfismo personalizamos. Estos son los personaje/objetos concretos que se devuelven en el método de creación y posterior clonación.
**Por temas de espacio omitimos poner los getters y setters en las subclases.

TestCreationalPatterns