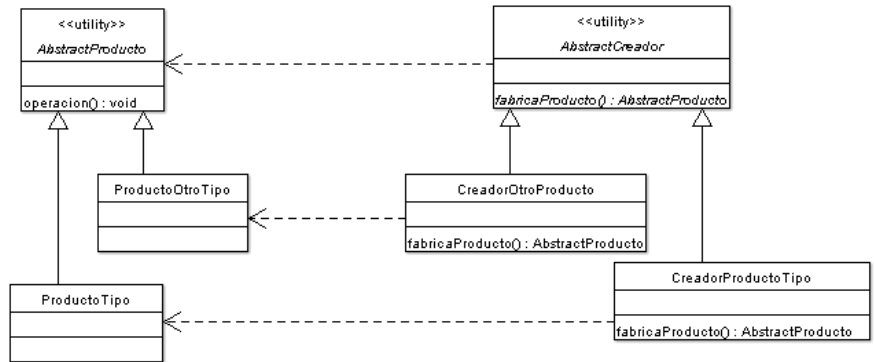


Práctica 4. Factory Method

1 INTRODUCCIÓN

En diseño de software, el patrón de diseño Factory Method consiste en utilizar una clase constructora (al estilo del Abstract Factory) abstracta con unos cuantos métodos definidos y otro(s) abstracto(s): el dedicado a la construcción de objetos de un subtipo de un tipo determinado. Es una simplificación del Abstract

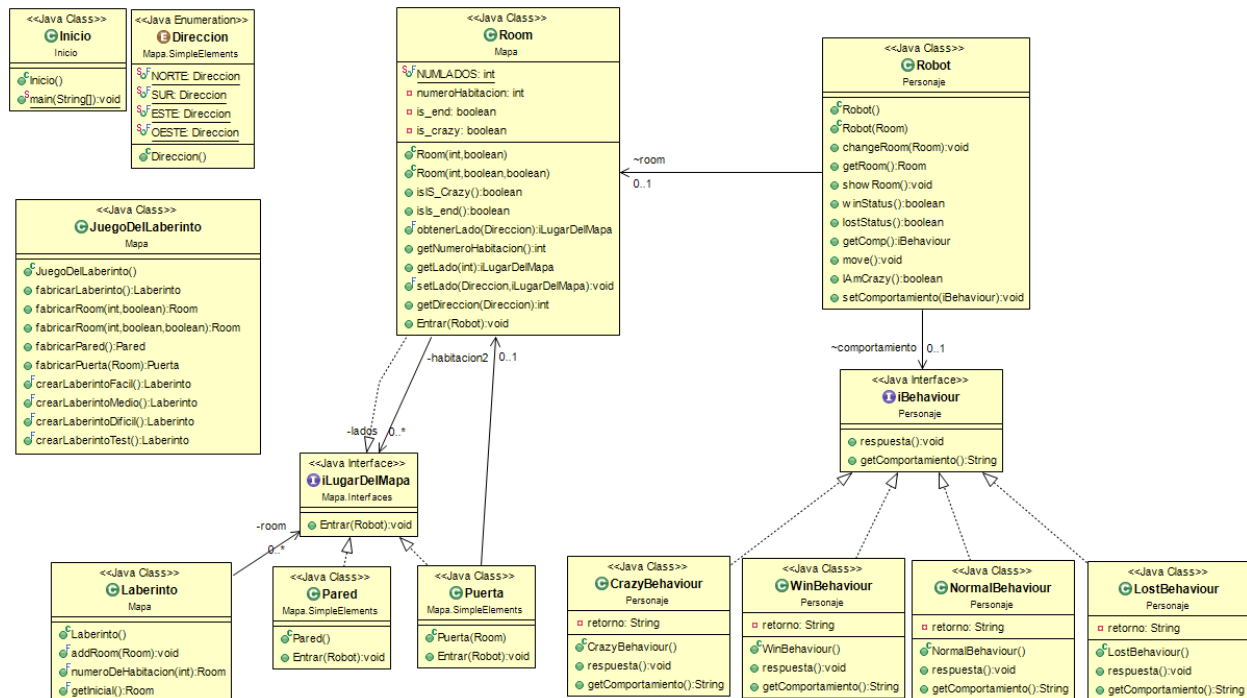
Factory, en la que la clase abstracta tiene métodos concretos que usan algunos de los abstractos; según usemos una u otra hija de esta clase abstracta, tendremos uno u otro comportamiento.



2 OBJETIVO

En esta práctica se pretende crear el “tablero” de un juego con determinadas características. Aunque el tablero se instanciara en la forma de un laberinto, el planteamiento utilizado puede ser aplicado a otros casos de uso. En este caso que nos ocupa, se define el laberinto como un conjunto de habitaciones. Una habitación conoce sus lindes, que pueden ser otra habitación, una pared, o una puerta a otra habitación. Las clases Habitación, Puerta y Pared definen los componentes del Laberinto. Cada habitación tiene cuatro lados.

3 DIAGRAMA UML



4 CÓDIGO GENERADO

El código generado para la práctica se ha basado en los códigos aportados por la profesora. En él hemos implementado algunos métodos, y añadido algunas clases.

Para crear una habitación deberemos de controlar dos parametros, el numero de la habitación, el cual deberá de ser único, y lo siguiente será una variable booleana que controlará si se trata de una habitación común, o por otro lado se trata de una habitación final. Por otra parte la forma de crear una puerta, se hará indicando la habitación a la que comunica.

Para definir una habitación deberemos de agregar un valor a cada una de las direcciones, las cuales podrían ser Puertas o Paredes. Por último añadiremos estas habitaciones creadas, al laberinto.

La traza de ejecución del programa sería generar un Laberinto a partir de las habitaciones y de los elementos de mapa, y comenzaremos a decidir los movimientos que deberemos de realizar para conseguir nuestro objetivo, llegar a la habitación final.

Enlace al código en Github [aquí](#).