Diseño de Software Orientado a Objetos
Práctico 5: Patrones Factory Method y Abstract Factory

Ejercicio 0. Descargue el código provisto en el repositorio git a continuación:

https://github.com/pponzio/oo-design-2024.git

La carpeta `5-factory-pizza-store` contiene el código complementario a esta práctica. Se proveen scripts gradle para compilar automáticamente el código y ejecutar los tests (ej. ejecutar *gradle test* en línea de comandos). Se recomienda utilizar algún IDE (ej. IntelliJ Idea) para facilitar la codificación, el refactoring, la ejecución de tests y el debugging.

Ejercicio 1. Utilice el código provisto de las distintas versiones de Pizza Store para experimentar y entender las distintas versiones del patrón Factory. Utilice la metodología de TDD.

a) Agregue una nueva variedad argentina de pizza a la versión de PizzaStore que utiliza Simple Factory (paquete `pizzastore.simplefactory`).
b) Agregue una nueva versión de `PizzaStore` con variedades Argentinas de Pizzas a la implementación que utiliza Factory Method (paquete `pizzastore.factorymethod`)
c) Agregue una nueva familia de ingredientes para la `PizzaStore` Argentina del inciso anterior a la implementación que utiliza Abstract Factory para los ingredientes (paquete `pizzastore.abstractfactory`)
d) ¿Tuvo que modificar la implementación de alguna de las clases existentes, a excepción quizás de las factories, para resolver los puntos a, b y c anteriores?

Ejercicio 2. Transforme la versión de `PizzaStore` que usa el patrón Factory Method en una versión equivalente pero que use el patrón Abstract Factory. Utilice la metodología de TDD. Reutilice los tests que escribió para el ejercicio 1.b para asegurarse que su implementación funciona correctamente para los casos dados. Modifique los tests lo mínimo posible para soportar el nuevo diseño.

Ejercicio 3. Utilice el patrón Factory para encapsular la creación de bebidas decoradas del ejercicio 1 de la práctica anterior (práctica 4 del patrón Decorator).

Ejercicio 4. Queremos agregar soporte para diferentes versiones en el juego de acción y aventuras desarrollado como solución al ejercicio 3 de la práctica anterior. Por ejemplo, queremos poder instanciar la versión 1.0 del juego, y también la versión 2.0. Las nuevas

versiones sólo pueden agregar nuevos tipos de personajes y nuevas armas respecto de versiones anteriores. ¿Qué patrón utilizaría para dar soporte a las distintas versiones? Realice un diagrama de clases y provea una implementación que satisfaga estos requisitos. Utilice la metodología de TDD durante todo el ejercicio.