Assignment 1: Design Patterns

Following up from the class activity and lab in design patterns this assignment exposes to other patterns such as the Factory Method pattern (Factory method pattern - Wikipedia) and the Abstract Factory pattern (https://en.wikipedia.org/wiki/Abstract_factory_pattern).

Submission Instructions

Do all your work in a GitHub repository and submit in Canvas the link to the repository.

Abstract Factory Pattern

The Abstract Factory pattern provides a way to encapsulate a group of individual factories that have a common theme without specifying their concrete classes. Simple put, clients use the particular product methods in the abstract class to create different objects of the product.

Factory Method Pattern

The Factory Method pattern creates objects without specifying the exact class to create. The Factory Method design pattern solves problems like:

- How can an object be created so that subclasses can redefine which class to instantiate?
- How can a class defer instantiation to subclasses?

Use Case Scenario

We would like to use an Abstract Factory to create products for inventory and at the same time set the price of the product. The price of the product is set after the product is created and is read from a database (in this assignment that database can be file of product names and prices.). For setting the price of the product one can use a Factory Method pattern.

Exercise

1. Create a UML diagram of your design that includes a *ProductFactory* class (concrete implementation of an Abstract Factory class) that will create different product types. For the particular product types take advantage of the Factory Method pattern to set the price of the product based on the amount stored in a file.
2. Implement the design in Java and include a test driver to demonstrate that the code works using 2 examples of a product.