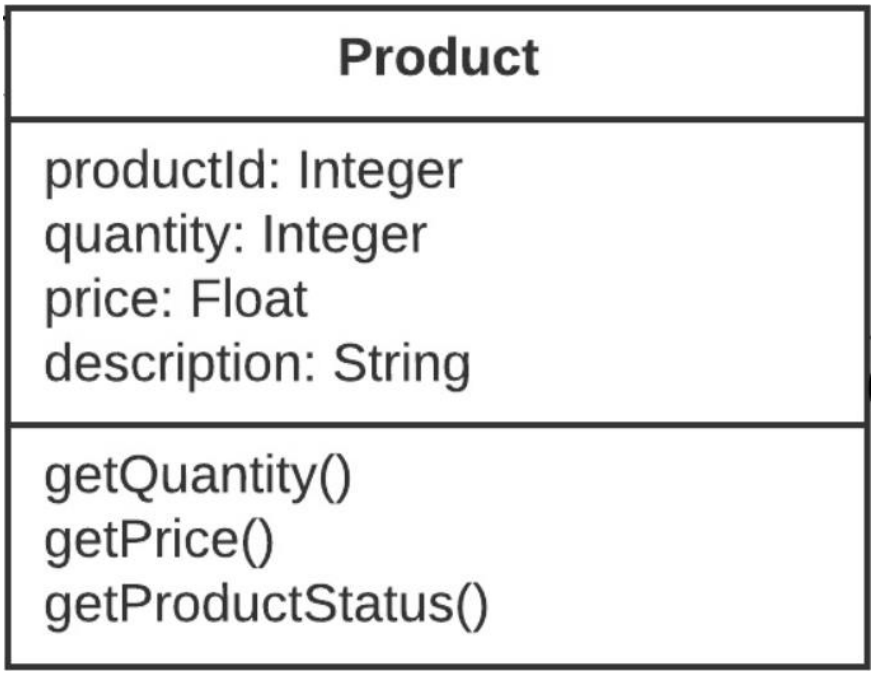
Abstract Factory Pattern

Scenario:

Abstract factory pattern is used to wrap the methods of the function and objects and other factory classes can use the methods without knowing the details. This factory is also called a factory of factories. It's a creational pattern that provides objects for other factories to use.

Implementation:

For this project, we have a products class that has product price, quantity, and other product details with exposing all methods of the main product class we made the product class as abstract and imported base classes as Electronics and furniture which are factory classes in our cases by creating this class, we can set prices and other details for each category of the products.

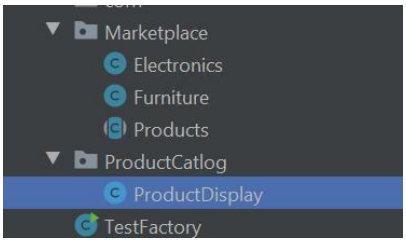


All the objects are made private and methods to access them are made public so that other classes can access the data without knowing the details of the main product line

Code:

We have a marketplace package consisting of the following classes

Marketplace (Package)



Abstract class → Products

Factory class 1 → Electronics

Factory class 2 → Furniture

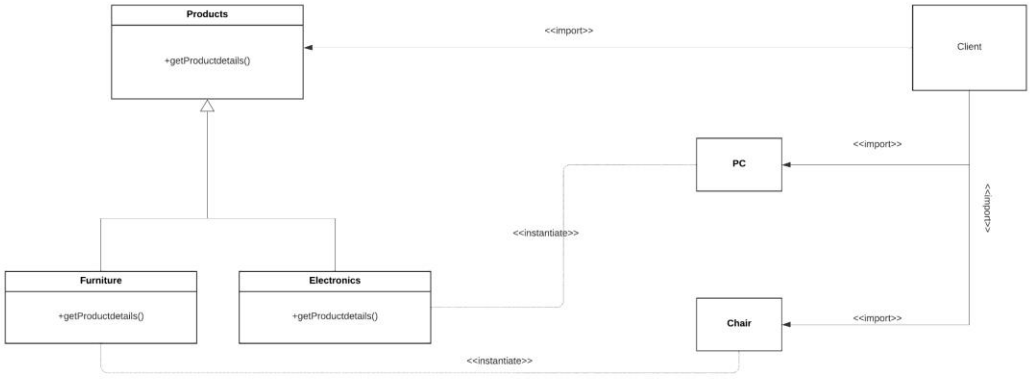
Product Display (Package)

Factory Driver Class → ProductDisplay

Test Factory class → Client access to the Program

Output:

Class Diagram:



Sequence State Diagram: