



SDET Course

Design Patterns - Abstract Factory

- Creational
 - Singleton
 - Builder
 - Prototype
 - Factory Method
 - **Abstract Factory**
- Structural
 - Adapter
 - Composite
 - Proxy
 - Flyweight
 - Bridge
 - Facade
 - Decorator
- Behavioral
 - Strategy
 - Observer
 - Command
 - Memento
 - State
 - Template Method
 - Mediator
 - Chain of Responsibility
 - Interpreter
 - Visitor
 - Iterator

Agenda

- Description
- Diagram
- Code sample (Java)
- Use cases

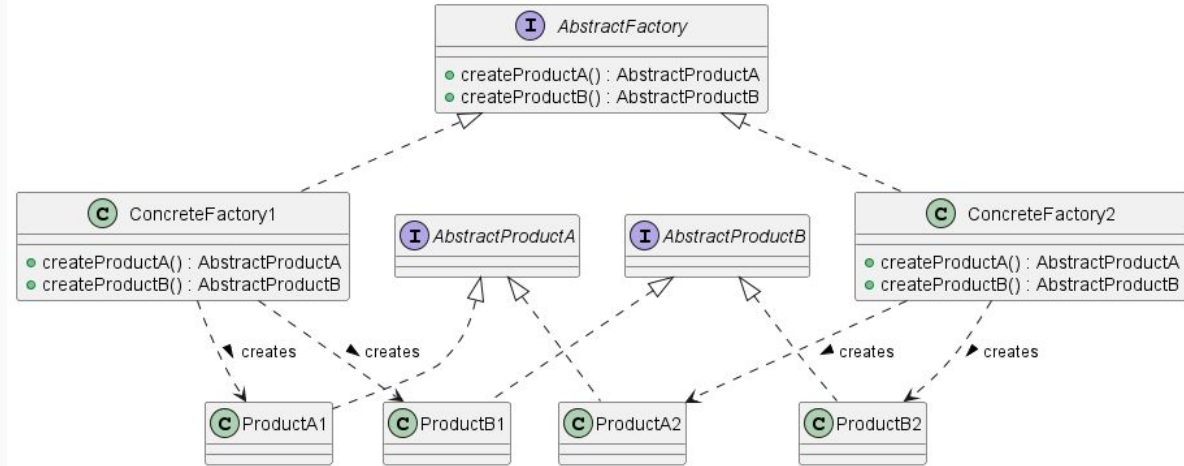


Description

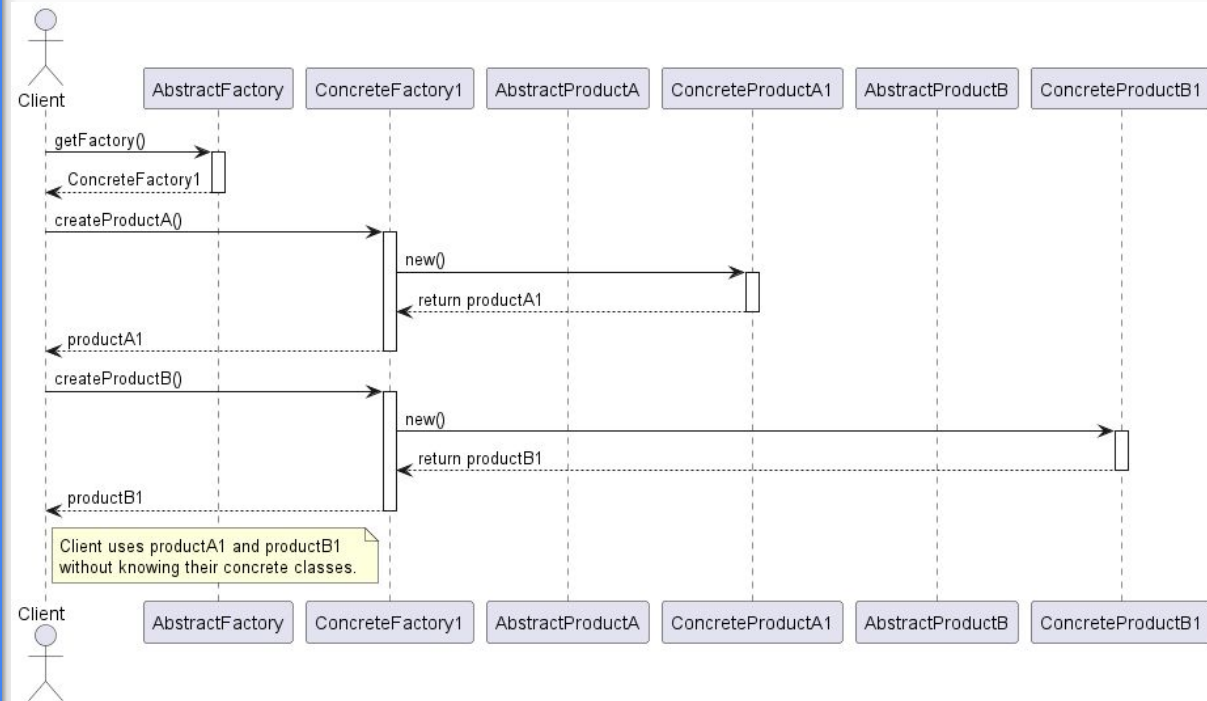
The Abstract Factory Design Pattern provides an interface for creating families of related or dependent objects without specifying their concrete classes. This pattern is useful when there is a need to ensure that the created objects can work together and you want to hide the implementation details from the client.



Class Diagram



Sequence Diagram



Code Sample

- General
 - Vehicle Manufacturing
 - Database Abstraction Layer
 - Cross-Platform File System Operations
 - Theme for applications
- In Test Automation
 - Cross-Platform UI Testing
 - Browser Compatibility Testing
 - Mobile App Testing on Different Devices
 - Similar page objects with small differences (e.g different types of calendar object)



Happy Coding