SDET Course

Design Patterns - Factory Method

3 Types of Design Patterns

- 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

Verisoft

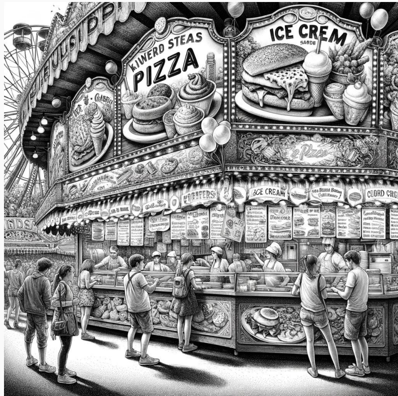
Descri

on

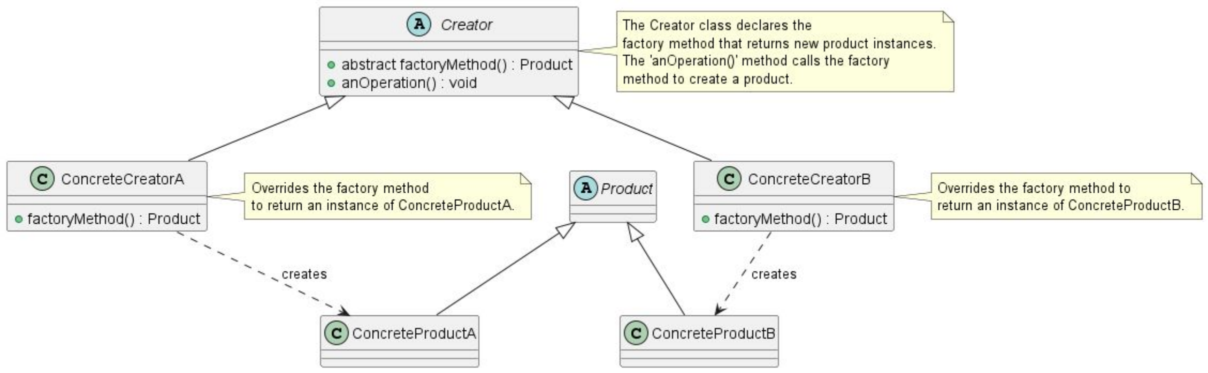
Introduction - Food stands in an amusement park

Description

The Factory Method Design Pattern is a creational pattern that provides an interface for creating objects in a superclass but allows subclasses to alter the type of objects that will be created. This pattern is particularly useful when there is a need to encapsulate the instantiation process of a product, making a class independent of how its objects are created, composed, and represented.



Class Diagram

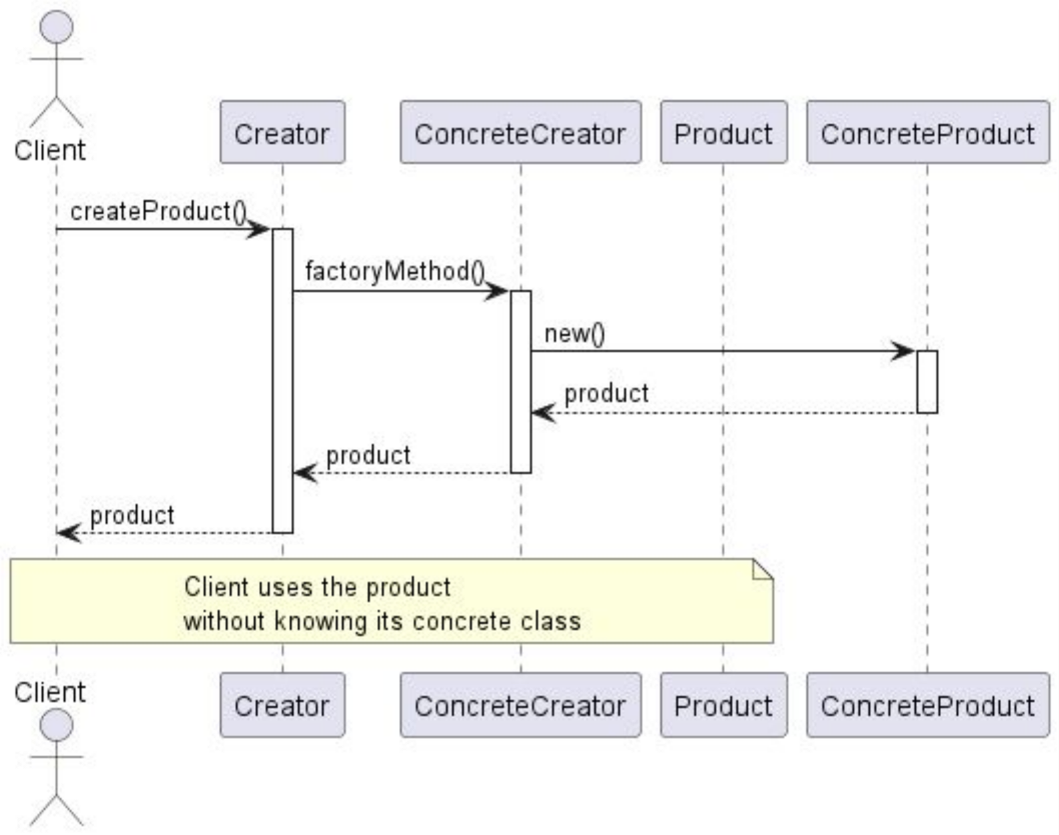


The Creator class declares the factory method that returns new product instances. The 'anOperation()' method calls the factory method to create a product.

Overrides the factory method to return an instance of ConcreteProductA.

Overrides the factory method to return an instance of ConcreteProductB.

Sequence Diagram



Code Sample

Use cases

- General
  - Browser Instance Creations
  - Data export (pdf, csv, xlsx)
  - UI component different OS

- In Test Automation
  - Browser instance creation
  - Test data generation
  - Configurable test environment setup
  - Page objects factory

Happy Coding