SDET Course

Design Patterns - Template 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

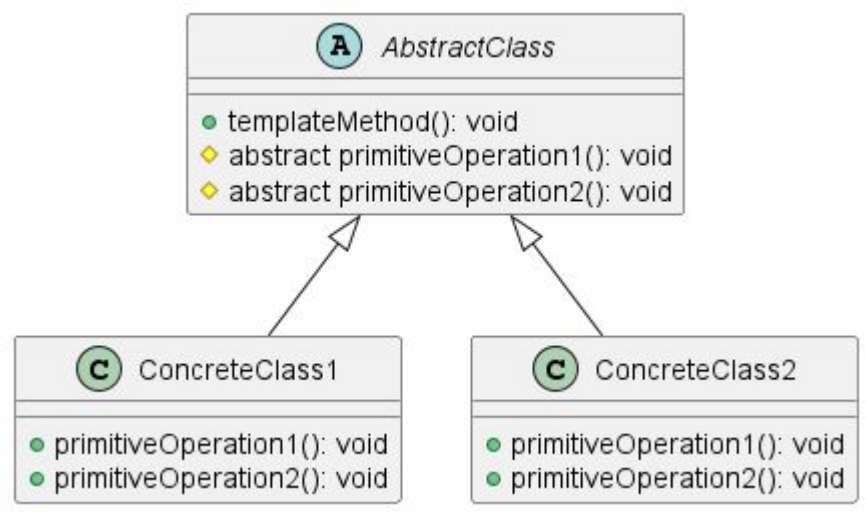
VERISOFT

Descrip
on

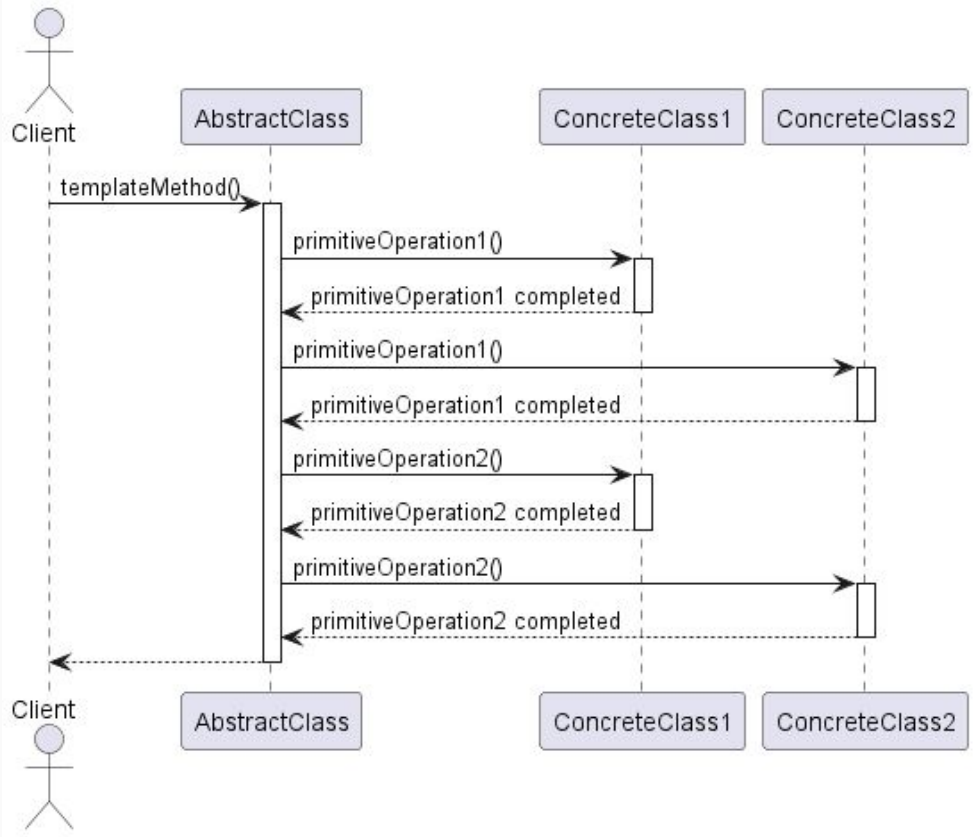
Description

The Template Method design pattern is a behavioral design pattern that defines the program skeleton of an algorithm in a method, deferring some steps to subclasses. This pattern allows subclasses to redefine certain steps of an algorithm without changing the algorithm's structure. The template method within a parent class contains a series of method calls that each perform a step of the algorithm. Some of these steps are implemented directly in the template method, while others are abstract or virtual and must be implemented in subclasses. This design pattern is particularly useful for algorithms with a fixed sequence of steps, but where the details of one or more steps may vary depending on the specific context or data. By leveraging the Template Method pattern, software developers can encapsulate varying behaviors within a subclass while ensuring that the overarching algorithm remains unchanged and consistent.

Class Diagram



Sequence Diagram



Code Sample

Use cases

- General
  - Software Build Processes: Preprocess, Compile, Link, Package
  - Educational Content Delivery
  - This presentation?
  - Maven: Validate, Compile, Test, Package, Verify, Install, Deploy

- In Test Automation
  - All types of testing - it is a lifecycle design pattern
  - The Retry Mechanism

Happy Coding