



# SDET Course

Design Patterns - Singleton

- 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

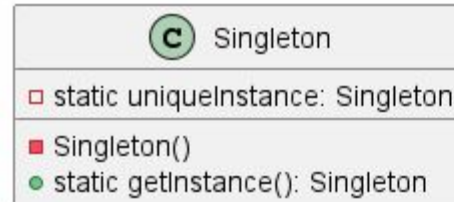


# Description

The Singleton Pattern is a design pattern that ensures a class has only one instance, and provides a global point of access to it. This pattern is particularly useful when exactly one object is needed to coordinate actions across the system.

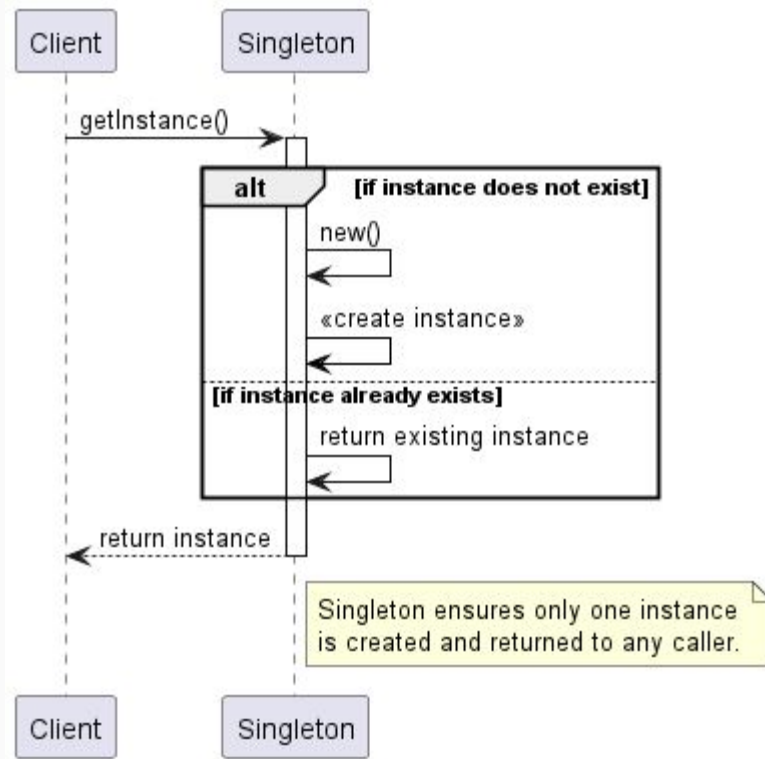


# Class Diagram



Singleton uses a private constructor and a private static instance variable. The `getInstance()` method ensures that only one instance is created and returned.

# Sequence Diagram



# Code Sample



- General
  - Global configuration file
  - Database connection
  - Logger
  - Global memory management
- In Test Automation
  - WebDriver instance
  - Report Instance
  - Capabilities Instance
  - Connector

# Happy Coding