# Object-Oriented Methodology Quiz 10

*2024 Fall Semester*

21 CST H3Art

*Final Score: 97.5/100*

**1. What is the main responsibility of the getInstance method in the Singleton Pattern?**

A. To initialize the Singleton
B. To ensure thread safety
**C. To provide access to the Singleton instance**
D. To create multiple instances

**2. Which of the following is NOT a common use case for the Singleton Pattern?**

A. Thread pools
**B. Multiple copies of the same object**
C. Dialog boxes
D. Device drivers

**3. How is the Singleton instance typically created?**

A. Automatically by the JVM
**B. By calling a static method**
C. Using a public constructor
D. Through inheritance

**4. What problem does the Singleton Pattern solve?**

**A. Ensuring a unique global resource**
B. Polymorphism
C. Data encapsulation
D. Code reuse

**5. In which scenario is subclassing a Singleton typically not recommended?**

A. When the Singleton needs to be extensible
B. When the Singleton manages global resources
C. When the Singleton is used for logging
**D. When multiple instances are required**

**6. How can the Singleton Pattern be implemented to avoid synchronization overhead?**

**A. By using double-checked locking**
B. By using a global variable
C. By making the constructor public
D. By making the getInstance method non-static

**7. What is the purpose of the undo() method in the Command Pattern?**

A. To store the command for later use.
**B. To reverse the last action of the command.**

C. To log the command execution.
D. To execute the command again.

**8. In the Command Pattern, what is the role of the Invoker?**

A. It creates the receiver objects.
**B. It executes commands.**
C. It defines the actions to be performed.
D. It stores commands.

**9. In the remote control example, what does the NoCommand class represent?**

A. A command to log the actions.
B. A command to execute multiple actions.
C. A command to turn off all devices.
**D. A placeholder for an unassigned slot.**

**10. What is the primary purpose of the execute() method in a Command object?**

A. To check the validity of the command.
B. To store the command for later use.
**C. To perform the actions defined by the command.**
D. To log the command.

**11. What method does the Command interface define?**

A. perform()
**B. execute()**
C. doAction()
D. run()

**12. In the Command Pattern, what is the role of the Receiver?**

A. It stores the commands.
B. It assigns commands to slots.
C. It creates the commands.
**D. It executes the actions defined by the commands.**

**13. What is the main purpose of the Singleton Pattern?**

A. To provide encapsulation
**B. To ensure only one instance of a class exists**
C. To create multiple instances of a class
D. To achieve polymorphism

**14. Which of the following is a disadvantage of using global variables instead of Singletons?**

A. Reduced performance
B. Simpler implementation
**C. Poor encapsulation**
D. Increased memory usage

**15. What is the purpose of making the Singleton constructor private?**

A. To enable lazy instantiation
B. To improve performance

**C. To prevent multiple instances**
D. To allow subclassing

**16. What is the term used for an object that encapsulates a request to do something?**

**A. Command**
B. Invoker
C. Client
D. Receiver

**17. What is the name of the pattern that allows adding responsibilities to objects dynamically?**

A. Observer Pattern
B. Adapter Pattern
C. Prototype Pattern
**D. Decorator Pattern**

**18. What does the Command Pattern encapsulate?**

**A. Requests**
B. Data
C. Methods
D. Variables

**19. What is the role of the setCommand() method in the remote control example?**

A. To execute a command.
B. To retrieve a command from a slot.
**C. To assign a command to a slot.**
D. To create a new command.

**20. What is the name of the pattern that allows creating commands that execute multiple other commands?**

A. Composite Command
B. Multi Command
C. Sequence Command
**D. Macro Command**

**21. Multiple threads can safely access the Singleton instance without any synchronization.**

❌

**22. In the remote control example, the RemoteControlWithUndo class tracks the last command executed to support undo functionality.**

✅

**23. Lazy instantiation is a common technique used in the Singleton Pattern to create the instance only when it is needed.**

✅

**24. The getInstance method in the Singleton Pattern is a static method that provides access to the Singleton instance.**

✅

**25. The Singleton Pattern ensures that only one instance of a class exists in an application.**

✅

**26. The Singleton Pattern is the most complex design pattern in terms of its class diagram.**

❌

**27. The Command Pattern can be used to implement queuing requests.**

✅

**28. The execute() method in a Command object always performs the same action.**

❌

**29. The NoCommand class is an example of a null object pattern.**

✅

**30. The Singleton constructor is usually declared as public.**

❌

**31. Macro commands can execute multiple commands when their execute() method is called.**

✅

**32. The Command Pattern supports undoable operations by implementing an undo() method in Command objects.**

✅

**33. The undo() method is always implemented in Command objects.**

❌

**34. The Singleton Pattern can be used to manage multiple copies of the same object.**

❌

❌ **35. In the remote control example, each slot can hold only one command.**

✅

Correct Answer: ❌

**36. Global variables are a better alternative to Singletons because they provide similar functionality with less code.**

❌

**37. The Command Pattern decouples the object making a request from the objects that execute the request.**

✅

**38. In the remote control example, the RemoteControl class acts as both the Invoker and the Receiver.**

❌

**39. The uniqueInstance variable in the Singleton Pattern is a static variable that holds the one and only instance of the Singleton.**

✅

**40. Eager instantiation creates the Singleton instance as soon as the class is loaded, regardless of whether it is needed.**

> ✅