Object-Oriented Methodology Quiz 12

2024 Fall Semester

21 CST H3Art

Final Score: 97.5/100

❌ **1. What is the purpose of the Null Iterator class?**

A. To provide a default implementation of the Iterator interface
B. To simplify the creation of composite structures
C. To handle empty collections
D. To return null when there are no more elements

Correct Answer: D

2. In the refactored design, what object does the GumballMachine delegate behavior to?

A. The State interface
B. The current State object
C. The context object

3. Which pattern allows access to the elements of an aggregate object sequentially without exposing its underlying representation?

A. Strategy
B. Facade
C. Observer
D. Iterator

4. What happens when the insertQuarter() method is called in the NoQuarterState?

A. The machine dispenses a gumball
B. The machine returns the quarter.
C. The machine transitions to the HasQuarterState

5. Which of the following statements about the Iterator Pattern is true?

A. It cannot be used with arrays
B. It does not support polymorphism
C. It requires clients to know the internal structure of the collection
D. It encapsulates the iteration logic

6. Which of the following is NOT a benefit of the Composite Pattern?

A. Increases the coupling between client and component code
B. Supports recursive operations
C. Provides a flexible way to represent part-whole hierarchies
D. Simplifies client code by allowing uniform treatment of composites and leaves

7. In the Composite Pattern, what is the term used to describe objects that do not contain other objects?

A. Leaf nodes
B. Composites
C. Branches
D. Root nodes

8. Which of the following is NOT a responsibility of the Iterator interface?

A. Providing a next() method
B. Managing the storage of collection elements
C. Providing a remove() method
D. Providing a hasNext() method

9. In the Composite Pattern, what is the role of a Composite object?

A. To manage the entire hierarchy of objects
B. To represent a single item in the structure
C. To provide an interface for both leaf nodes and composite nodes
D. To simplify complex operations

10. What does the State Pattern allow an object to do?

A. To alter its behavior when its internal state changes.
B. To share behaviors with other objects.
C. To have a single, fixed behavior.

11. What is the purpose of the createIterator() method in the Composite Pattern example?

A. To retrieve an iterator for traversing the menu items
B. To get the description of the menu
C. To remove elements from the menu
D. To add elements to the menu

12. What is the role of the MenuComponent interface in the Composite Pattern example?

A. To provide a common interface for menus and menu items
B. To manage the order of menu items
C. To define the behavior of menu items
D. To handle user input

13. What is the purpose of the WinnerState in the GumballMachine?

A. To handle the case where the machine dispenses two gumballs instead of one.
B. To handle the case where the machine runs out of gumballs.
C. To handle the case where the machine doesn't accept a quarter.

14. What is the main difference between the State Pattern and the Strategy Pattern?

A. The Number of Classes
B. The Structure
C. The Intent

15. Why is encapsulation important in the Iterator Pattern?

A. To make code more readable
B. To improve performance

C. To simplify the creation of new objects

D. To hide the implementation details of the collection

16. What is the main purpose of the Iterator Pattern?

A. To simplify the interface of a group of classes

B. To allow a group of objects to be notified when some state changes

C. To provide a way to traverse a collection of objects without exposing its implementation

D. To change the interface of a class

17. What class is responsible for the behavior of the machine when it has a quarter but hasn't yet turned the crank?

A. SoldState

B. NoQuarterState

C. HasQuarterState

18. In the given example, which class throws an UnsupportedOperationException when isVegetarian() is called?

A. MenuItem

B. CafeMenu

C. DinerMenu

D. Menu

19. What is the main goal of the State Pattern?

A. To define a family of algorithms, encapsulate each one, and make them interchangeable

B. To encapsulate state-based behavior and delegate it to the current state

C. To encapsulate interchangeable behaviors

20. What is the primary advantage of using design patterns like Iterator and Composite?

A. They reduce the amount of code needed

B. They simplify debugging

C. They make code more flexible and reusable

D. They increase the speed of execution

21. The insertQuarter() method in the HasQuarterState returns the quarter to the user.

❌

22. The Strategy Pattern and the State Pattern have the same class diagram but differ in intent.

✅

23. The State Pattern is used to encapsulate state-based behavior and delegate it to the current state.

✅

24. The State Pattern is used to encapsulate interchangeable behaviors.

❌

25. The Winner State dispenses one gumball instead of two.

❌

26. The Sold State transitions to the NoQuarterState after dispensing a gumball if there are more gumballs left.

✅

27. The MenuComponent interface in the Composite Pattern example includes methods like add(), remove(), and getChild().

✅

28. The Composite Pattern can be used to represent hierarchical structures in code.

✅

29. The Iterator Pattern and Composite Pattern can be used together to provide a flexible way to traverse and manipulate hierarchical structures.

✅

30. The print() method in the Menu class recursively calls itself to print all menu items and submenus.

✅

31. The GumballMachine class in the refactored design does not contain any conditional statements.

✅

32. In the original implementation, the authors identified that the code was easy to extend and modify.

❌

33. In the refactored design, the GumballMachine class contains conditional statements to handle state transitions.

❌

34. In the given example, the Menu class extends the MenuComponent abstract class.

✅

35. The Null Iterator class always returns null when next() is called.

✅

36. The printVegetarianMenu() method in the Waitress class uses the Composite Iterator to find vegetarian items.

✅

37. The Composite Pattern is used when there is no need to treat individual objects and compositions of objects uniformly.

❌

38. In the Iterator Pattern example, both DinerMenu and PancakeHouseMenu implement the Menu interface.

✅

39. In the original implementation of the Gumball Machine, the authors used conditional statements to handle state transitions.

✅

40. The createIterator() method is used to add new elements to a menu in the Composite Pattern example.

❌