Quiz Performance Report

Name: Chaitu

Topic: OOP

Date: 2025-09-16 11:10:25

Final Score: 4 / 10

AI Performance Summary:

Could not generate a detailed analysis. Your final score was 4/10. Please try again later.

AI Recommendations & Resources:

No recommendations available.

Question Breakdown:

Question 1: Consider a scenario where you need to model different types of vehicles (cars, trucks, motorcycles) in an object-oriented programming system. Which design principle would be MOST effective in ensuring maintainability and extensibility if you later need to add new vehicle types like buses or bicycles?
Your Answer: Employing the Strategy pattern to define vehicle-specific algorithms for actions like starting and braking. (Incorrect ■)
Correct Answer: Applying the Template Method pattern to define a skeletal algorithm for vehicle operation with customizable steps.
Difficulty: medium

Question 2: Consider a scenario where you need to model different types of vehicles (cars, trucks, motorcycles) in an object-oriented program. Which design principle would be MOST effective in ensuring extensibility and maintainability while avoiding code duplication?
Your Answer: Employing inheritance, with a base 'Vehicle' class defining common attributes and methods, and derived classes (Car, Truck, Motorcycle) inheriting and extending them. (Correct ■)
Correct Answer: Employing inheritance, with a base 'Vehicle' class defining common attributes and methods, and derived classes (Car, Truck, Motorcycle) inheriting and extending them.
Difficulty: medium

Question 3: Which of the following best describes the concept of polymorphism in Object-Oriented Programming?
Your Answer: The mechanism that restricts access to certain methods or variables within a class, protecting data integrity. (Incorrect ■)
Correct Answer: The ability of objects of different classes to respond to the same method call in their own specific way.
Difficulty: medium

Question 4: Which of the following best describes the concept of polymorphism in Object-Oriented Programming?
Your Answer: The ability of objects of different classes to respond to the same method call in their own specific way. (Correct ■)
Correct Answer: The ability of objects of different classes to respond to the same method call in their own specific way.
Difficulty: medium

Question 5: Consider a scenario where you need to implement the Strategy pattern to dynamically change the sorting algorithm used on a list of objects. You have three concrete strategy classes: QuickSort, MergeSort, and BubbleSort, each implementing the same interface 'SortingStrategy'. Which of the following design choices would BEST address the potential for the 'Context' class (the class using the sorting strategies) to become overly complex and difficult to maintain as the number of sorting strategies grows, while still upholding the principles of the Strategy pattern?
Your Answer: Use a generic type parameter within the Context class to constrain the type of sorting algorithm, enabling the compiler to enforce correct type relationships. (Incorrect ■)
Correct Answer: Create a single 'SortingStrategyFactory' class that encapsulates the strategy creation logic, abstracting the strategy instantiation away from the Context.
Difficulty: hard

Question 6: Consider a scenario where you need to implement the Strategy pattern to manage different sorting algorithms for a list of objects. You have three concrete strategy classes: MergeSort, QuickSort, and HeapSort. However, you also need to introduce a new sorting algorithm, RadixSort, that requires access to the internal structure of the sortable objects (e.g., to efficiently access individual

digits for radix sorting). Which of the following design choices best avoids breaking the Strategy pattern's encapsulation and maintains flexibility for adding future algorithms?
Your Answer: Modify the 'SortStrategy' interface to include a method providing access to the internal object structure. This allows RadixSort to access this information, but might break encapsulation for other strategies. (Incorrect ■)
Correct Answer: Use a factory pattern to instantiate the correct sorting strategy based on runtime parameters, allowing the factory to manage object access for specialized sorting strategies like RadixSort without modifying existing interfaces or strategies.
Difficulty: hard

Question 7: Consider a scenario where you need to implement a 'Flyweight' design pattern to manage a large number of similar objects. Which of the following statements BEST describes a crucial consideration when determining whether an object is suitable for flyweight sharing, considering the intrinsic and extrinsic states?
Your Answer: An object is suitable for flyweight sharing if its intrinsic state can be easily copied and its extrinsic state is immutable. (Incorrect ■)
Correct Answer: An object is suitable for flyweight sharing if its intrinsic state is immutable and its extrinsic state is passed to the object as parameters at runtime, avoiding redundant storage.
Difficulty: hard

Question 8: Consider a scenario where you need to implement the Strategy pattern to allow different sorting algorithms (Bubble Sort, Merge Sort, Quick Sort) to be used interchangeably with a list of objects. Which statement regarding the design and implementation of this scenario is MOST accurate, considering efficiency and extensibility?
Your Answer: Each sorting algorithm (Bubble Sort, Merge Sort, Quick Sort) should be implemented as a separate class that implements a common interface (e.g., ISortingStrategy). The context class will then delegate sorting to an instance of the chosen strategy. Further algorithms can be added without modifying existing code. (Correct ■)
Correct Answer: Each sorting algorithm (Bubble Sort, Merge Sort, Quick Sort) should be implemented as a separate class that implements a common interface (e.g., ISortingStrategy). The context class will then delegate sorting to an instance of the chosen strategy. Further algorithms can be added without modifying existing code.
Difficulty: hard

Question 9: Which of the following best describes the concept of polymorphism in Object-Oriented Programming?
Your Answer: The ability of objects of different classes to respond to the same method call in their own specific way. (Correct ■)
Correct Answer: The ability of objects of different classes to respond to the same method call in their own specific way.
Difficulty: medium

Question 10: Which of the following statements best describes polymorphism in object-oriented programming?
Your Answer: Polymorphism is the process of creating new classes from existing classes, inheriting their properties and methods. (Incorrect ■)
Correct Answer: Polymorphism allows a single method name to perform different actions depending on the object it's called on, enhancing code flexibility and reusability.
Difficulty: medium