Soru 1

Yanlış

Soruyu işaretle

What is the primary advantage of using enums in (Rick's) application redesign?

a. Reduces the need for type safety.
b. Avoids errors by restricting values to predefined options.
c. Simplifies the graphical representation of the data.
d. Allows dynamic updates without recompilation. ⊗
e. Ensures backward compatibility with older code.

Soru 2

Yanlış

Soruyu işaretle

How does the principle of separation of concerns simplify debugging?

a. By isolating functionalities so issues can be traced to specific modules
b. By merging all related functionalities into a single file ⊗
c. By ensuring that no module can function independently
d. By allowing all developers to work on the same module simultaneously
e. By hardcoding the expected results to reduce errors

Soru 29

Doğru

Soruyu işaretle

What does it mean for a software module to have "single responsibility"?

a. It handles all aspects of the system.
b. It focuses on one specific functionality or purpose. ✓
c. It is restricted to a single developer for maintenance.
d. It operates independently of all other modules.
e. It uses only one type of programming language.

Soru 30

Doğru

Soruyu işaretle

What principle does encapsulation support in software design?

a. Open-Closed Principle ✓
b. Single Responsibility Principle
c. Dependency Inversion Principle
d. Encapsulation Principle
e. Least Privilege Principle

Soru 27

Yanlış

Soruyu işaretle

What does the Open-Closed Principle (OCP) emphasize?

a. Classes should be open for modification and closed for extension.
b. Classes should be open for extension but closed for modification.
c. Every class should allow unlimited changes after implementation.
d. Classes should be closed for testing but open for debugging.
e. Classes should be locked after deployment to avoid errors. ⊗

Soru 28

Yanlış

Soruyu işaretle

What issue can arise from focusing too much on feature-driven development?

a. Neglecting how features interact in real-world scenarios
b. Spending excessive time on use case diagrams
c. Losing focus on the modularity of the system ⊗
d. Increasing the complexity of individual features
e. Overloading the system with redundant tests

Soru 25

Doğru

Soruyu işaretle

How can developers ensure the Liskov Substitution Principle (LSP) is maintained in their design?

a. By avoiding polymorphism in subclassing
b. By ensuring subclasses do not alter the behavior of the base class ✓
c. By writing methods with static return types only
d. By using inheritance exclusively for implementation reuse
e. By removing all overridden methods from subclasses

Soru 26

Yanlış

Soruyu işaretle

Which one is NOT an Object-Oriented Design Principle?

a. Encapsulate what varies
b. Code to an interface rather than to an implementation ✗
c. Each class in your application should have only one reason to change
d. Make your code as flexible as possible. Change is inevitable.
e. Favor composition over inheritance

Soru 23

Doğru

Soruyu işaretle

What principle is demonstrated by breaking a problem into smaller parts that can be solved independently?

a. Divide and Conquer ✓
b. Don't Repeat Yourself
c. Dependency Inversion
d. Liskov Substitution
e. Single Responsibility

Soru 24

Doğru

Soruyu işaretle

What challenge might arise if domain analysis is skipped in the OOA&D process?

a. Difficulty in testing edge cases during implementation
b. Overlapping responsibilities between modules
c. Misalignment between the system's objects and customer requirements ✓
d. Delayed deployment due to lack of features
e. Reduced runtime performance of the system

Soru 21

Doğru

Soruyu işaretle

Why is it essential to test for non-existent properties in a system?

a. To guarantee 100% code coverage in unit tests
b. To ensure the database schema supports all possible queries
c. To increase runtime performance under load
d. To validate the system's error handling and resilience ✓
e. To simplify the user interface

Soru 22

Doğru

Soruyu işaretle

What is a common characteristic of a well-designed architecture?

a. It includes redundant components for reliability
b. It is scalable, maintainable, and flexible for future changes ✓
c. It eliminates the need for user feedback
d. It depends heavily on a single framework
e. It avoids using abstraction to simplify the design

Soru 19

Doğru

Soruyu işaretle

What is the main focus of feature-driven development?

a. Completing individual features one at a time before moving to the next ✓
b. Developing the entire application at once to save time
c. Focusing on user scenarios rather than specific functionalities
d. Eliminating unnecessary testing during implementation
e. Prioritizing database schema design over user interaction

Soru 20

Yanlış

Soruyu işaretle

How does iteration ensure the success of the OOA&D process?

a. It focuses exclusively on the backend logic rather than user interactions
b. It reduces the total time required for testing and debugging
c. It eliminates the need for initial design and planning ✗
d. It ensures that all modules are developed simultaneously
e. It allows continuous refinement of each module based on customer feedback

Soru 17

Yanlış

Soruyu işaretle

How does abstraction help in managing large problems during design?

a. By emphasizing specific features instead of use cases
b. By eliminating the need for modularity ⊗
c. By replacing high-level modules with low-level implementations
d. By hiding unnecessary details and focusing on the core functionality
e. By increasing the complexity of the design

Soru 18

Doğru

Soruyu işaretle

What is the primary goal of the Don't Repeat Yourself (DRY) principle?

a. To reduce the number of lines in the codebase
b. To avoid duplicating code and logic across the application ✓
c. To focus on reducing runtime performance issues
d. To create a single method for all operations
e. To minimize testing by reusing test cases

Soru 15

Yanlış

Soruyu işaretle

Which situation demonstrates a violation of the Open-Closed Principle (OCP)?

a. Adding a layer of abstraction to enable flexibility
b. Using an interface to define new behaviors in a class ⊗
c. Creating a new class to handle additional requirements
d. Modifying an existing class to add new functionality instead of extending it
e. Refactoring a method to reduce redundancy

Soru 16

Yanlış

Soruyu işaretle

When would use case-driven development be more appropriate than feature-driven development?

a. When customer feedback is unavailable during development
b. When individual features need to be tested and implemented quickly
c. When the system involves complex user interactions and transactional processes
d. When the development team is small and lacks experience ⊗
e. When the system has no defined requirements

Soru 13

Yanlış

Soruyu işaretle

What is the role of alternate paths in a use case?

a. To handle scenarios where the primary path fails or changes.
b. To document unused features of the system.
c. To simplify the main path of the use case.
d. To reduce complexity by removing edge cases.
e. To ensure the system has no bugs. ⊗

Soru 14

Yanlış

Soruyu işaretle

What is a major drawback of violating the Dependency Inversion Principle (DIP)?

a. The overall design becomes too generic for specific use cases. ⊗
b. Abstractions lose their flexibility in the system.
c. High-level modules become tightly coupled with low-level modules.
d. Interfaces are ignored in favor of concrete implementations.
e. Modules become harder to scale due to shared responsibilities.

Soru 11

Yanlış

Soruyu işaretle

What role does textual analysis play in bridging use cases and design?

a. Prioritizing technical implementation over user interactions
b. Replacing the need for UML diagrams in the design phase ⊗
c. Simplifying the database schema by focusing on specific features
d. Identifying candidate classes and operations that map directly to system requirements
e. Avoiding redundancy in testing processes

Soru 12

Yanlış

Soruyu işaretle

How does architectural layering support system scalability?

a. By using a flat structure to simplify system design ⊗
b. By merging all layers into one to avoid communication overhead
c. By prioritizing frontend design over backend functionality
d. By ensuring all layers share the same responsibilities
e. By isolating functionalities into layers with minimal interdependencies

Soru 9

Doğru

Soruyu işaretle

What is a common trade-off when emphasizing encapsulation in design?

a. Poor error handling in edge cases
b. Difficulty in implementing interfaces
c. Reduced flexibility in extending the system
d. Higher coupling between unrelated modules
e. Increased runtime complexity due to indirect property access ✓

Soru 10

Yanlış

Soruyu işaretle

What is an "alternate path" in a use case?

a. A backup feature for system errors
b. A deviation from the main path to handle specific conditions
c. A simplified version of the main use case ✗
d. A testing procedure for identifying bugs
e. A redesigned functionality to meet customer demands

Soru 7

Yanlış

Soruyu işaretle

Why is it important to ensure loose coupling when breaking a large problem into modules?

a. To make all modules share the same dependencies
b. To reduce the need for testing and debugging
c. To focus on implementation over architecture
d. To minimize the overall codebase size ⊗
e. To enable modules to function independently without relying on others

Soru 8

Doğru

Soruyu işaretle

Why was (InstrumentSpec/Spec Class) refactored to use a properties map instead of hardcoded fields?

a. To support dynamic and flexible instrument specifications ✓
b. To improve performance during search operations
c. To enable direct integration with SQL databases
d. To enforce stricter compile-time type checking
e. To avoid using abstract classes entirely

Soru 5

Doğru

Soruyu işaretle

Why is modularity critical when breaking a problem into smaller components?

a. It ensures each module has a single responsibility and is easier to maintain ✓
b. It reduces the need for abstraction layers
c. It allows for hardcoding features in a single module
d. It eliminates the use of design patterns in system architecture
e. It simplifies testing by combining multiple responsibilities in one module

Soru 6

Yanlış

Soruyu işaretle

What is the primary purpose of iterating during software development?

a. To complete the system as quickly as possible without testing
b. To refine and improve the system through continuous feedback
c. To prioritize adding features over resolving design flaws
d. To avoid rewriting code during later stages of development ✗
e. To replace user feedback with automated tests

Soru 3

Doğru

Soruyu işaretle

What is the primary advantage of starting with a feature list in the OOA&D process?

a. It eliminates the need for iterative testing
b. It helps define the high-level functionality of the system early on ✓
c. It ensures all use cases are fully implemented before development begins
d. It focuses on database design rather than user interaction
e. It skips the analysis phase to save time

Soru 4

Yanlış

Soruyu işaretle

Why is violating the Interface Segregation Principle (ISP) problematic?

a. It allows multiple interfaces to serve unrelated modules
b. It eliminates the need for modularity in the application
c. It forces classes to implement methods they do not need
d. It prevents the use of inheritance in object-oriented programming ✗
e. It simplifies testing by combining interfaces into one