

# System Analysis and Design

学号:

姓名:

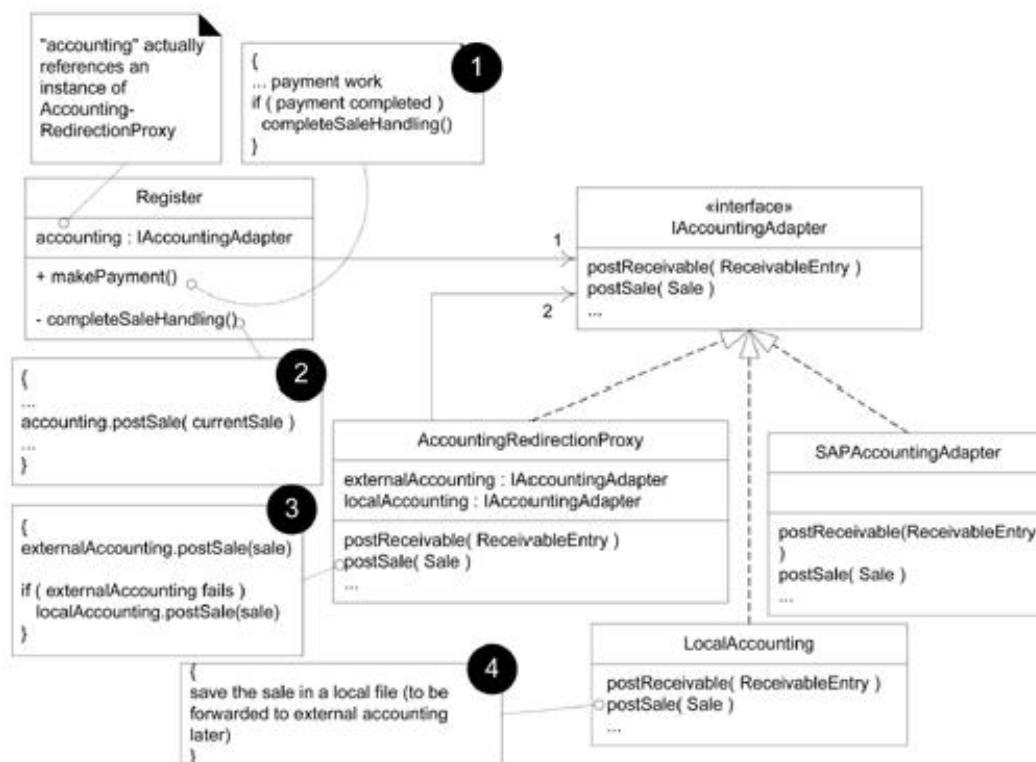
教学班:

---

1. What are the benefits of using layers in system architecture?
2. What ideas does model-view separation principle suggest?
3. Consider the following domain example: A store uses a set of external authorization services for payments. Each service associates a merchant ID with the store. For each service, different stores have different merchant IDs. Each store has different merchant IDs for different services. Where Should the merchantID be located in our domain model?
4. What is Class Responsibility Card? How can we do class responsibility card modeling?
5. In object-oriented design, what ideas does Dependency Inversion principle suggest?
6. If an object implements an interface, what type of its variable should be declared in your code for better programming?
7. What are the advantages of writing the test first?
8. Why metadata-based database mapper is considered to support protected variations?
9. What are the types and categories of Requirements?
10. There are some typical association relationships between objects in domain analysis, what are they?
11. What are the strengths and weaknesses of dynamic modeling using interaction diagrams?
12. What are the common forms of coupling from class X to class Y?
13. What are the main tasks we need to do during use case realization?
14. Write a basic structure (or structure pattern) of a testing method.
15. For solving the external services with varying interfaces problem, what patterns may be used in your design?
16. Cache can go stale quickly; it should be refreshed occasionally if possible. Try to give a strategy for cache refreshing using active object.
17. How can we model polymorphic messages in sequence diagrams?

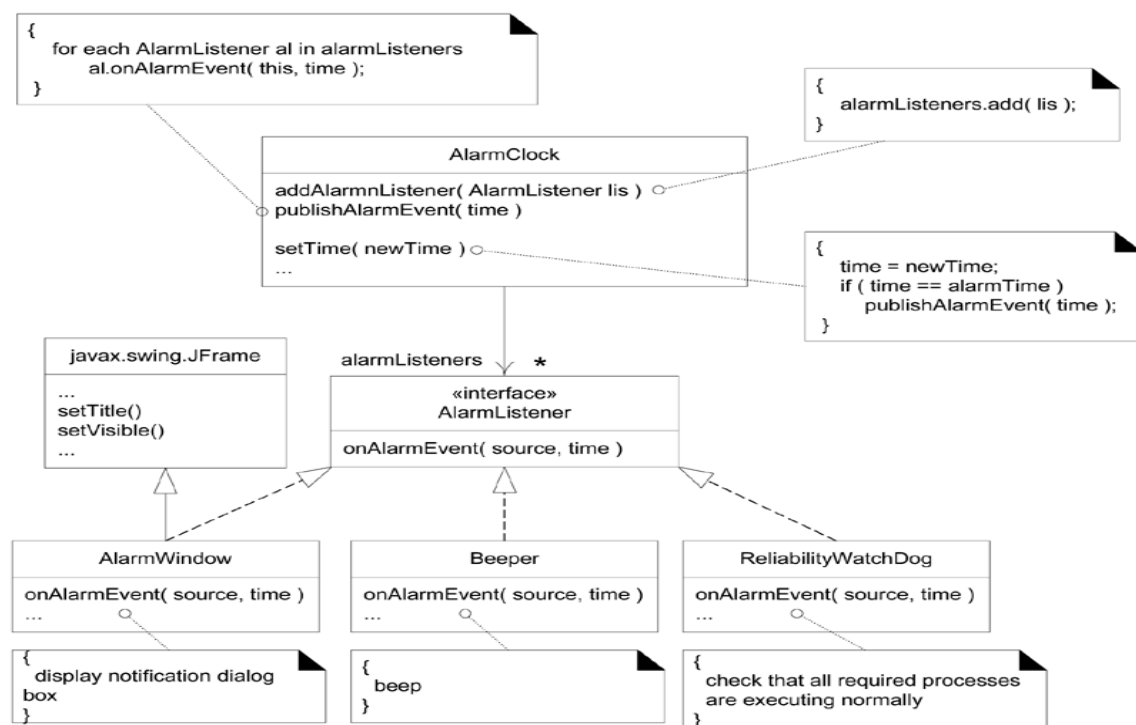
18. What are the goals of refactory?
19. What kind of problems can be solved by the GRASP Polymorphism pattern?
20. The NextGen POS system has multiple external third-party tax calculators. The system needs to be able to integrate with different ones. Each tax calculator has a different interface so need to adapt to each of these external fixed interfaces. What objects should be responsible for handling these varying external tax calculator interfaces?
21. Why we said that having classes that represent only domain-layer concepts in some design situations leads to problems?
22. Why it is a key in iterative and evolutionary methods to expect and plan for lots of change and deviation from the design during programming?
23. Why software is usually divided into modules?
24. Name all specific principles of OOD suggested by "S.O.L.I.D Principles".
25. What basic ideas does iterative and evolutionary development suggest ?
26. The design of objects can be broadly divided into two groups. What are they ?
27. Why the GRASP Pure Fabrication is an example of assigning responsibilities to support the GRASP Indirection?
28. Why we said that the GRASP Protected Variations pattern is a *very* important, fundamental principle of software design?
29. We usually have two points of changes in a system to be designed, what are they ?
30. Explain, in terms of GRASP terminology, how GoF adapter pattern supports GRASP Protected Variations.
31. What are the connections between Factory pattern and GRASP pure Fabrication pattern?
32. Why we prefer the lazy initialization in creation of Singleton object?
33. What is the problem to be solved by GoF Composite pattern?
34. Why cyclic dependency in a package is undesirable ?
35. In what usual way a framework can be used by a user?
36. Why object identification is needed in a persistent service?
37. What have been created during the object design modeling?
38. What is the problem to be solved by GRASP Information Expert pattern?

39. What pattern is indicated by the statement "If you don't need the advanced functionality or fine-grained control of the complex system, the simplified interface makes life easy".
40. What are the key points for understanding GoF Observer pattern?
41. What are the alternatives for loading the in-memory cache and the file-based cache?
42. How can you understand the mechanism of throwing exceptions in a system?
43. The following diagram illustrates the use of a redirection proxy for failover in NextGen POS Design. What are the differences from the structure of caching remote services in the same design?



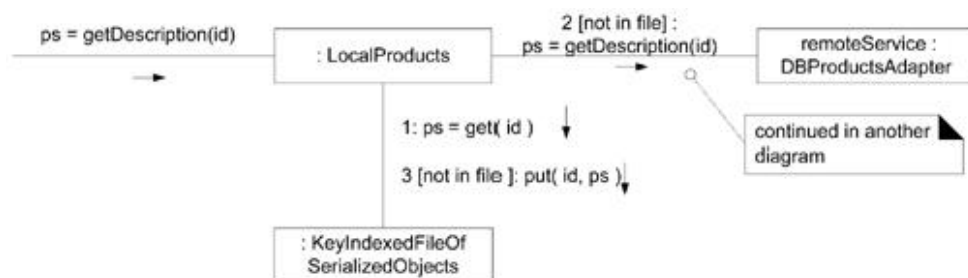
44. List some sample systems which can be easily modeled as state machines.
45. In what condition or situation we can define a conceptual subclass in a domain model?
46. What does the composition relation imply?
47. In actual coding, why changes in a package can ripple through your entire program?
48. How do you think why we separate a set of functionally related interfaces in a package from their implementation classes?

49. What is the problem to be solved by GRASP High Cohesion pattern ?
50. Use case realizations could be designed directly from the use case text or from one's domain knowledge. Why operation contract is necessary for use case realization?
51. What artifacts may be the UP implementation model?
52. What is the relationship between the disciplines and phases in UP?
53. What pattern is illustrated by the following diagram? Simply explain what and how the problem is solved.

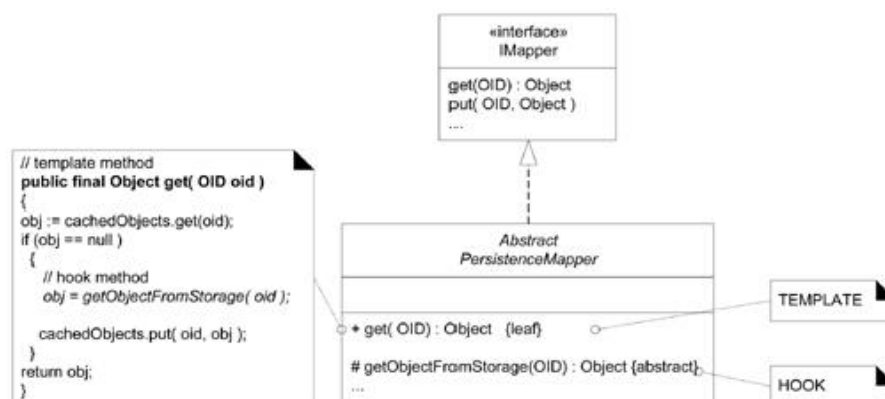


54. What basic strategy is illustrated by "Packaging by work and by clusters of unstable classes"?
55. What is responsibility-driven design?
56. When related alternatives or behaviors vary by type (class), how can you assign responsibility for the behavior to the types for which the behavior varies?
57. What artifacts are started in inception phase of a UP process?
58. Why requirements analysis is hard?
59. How many ways are there for requirement elicitation?
60. What should be a use case model in the unified process?
61. How to find use cases when you do system analysis?
62. What elements are included in the Supplementary Specification artifact?

63. The NextGen POS system needs to support external tax calculator services. A solution is to add a level of *indirection* with objects that adapt the varying external interfaces to a consistent interface used within the application. Draw a simple class diagram to show the adapter for POS system.
64. Why UML class diagram means different things in different contexts?
65. By what steps can we build the domain model in system analysis?
66. How can we do an Iterative and Evolutionary domain modeling?
67. What is a software architecture?
68. What functionality is illustrated by the following design?



69. The following diagram is an example design for mapper objects in a persistent framework. Draw a diagram to illustrate how it is used for user's design for a concrete mapper.



70. What kind of relationships in DCD may be usually implemented with the introduction of a collection object, such as a *List* or *Map*, or even a simple array?