Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Objektorientierte Analyse und Design

Wintersemester 2024
10. Aufgabenblatt

Erinnerung: Dies Aufgabenblatt dient der eigenen Erfolgskontrolle, Ergebnisse sind nicht abzugeben.

(Wiederholungsfragen, 0 Punkte)

1. Wie hat sich das Themengebiet „SW-Engineering“ chronologisch entwickelt?

2. Warum sind Prozessbeschreibungen im Arbeitsleben sinnvoll?
3. Beschreiben Sie die typischen Inhalte eines (Geschäfts-)Prozessmodells.
4. Nennen und beschreiben Sie typische Phasen der SW-Entwicklung.
5. **Beschreiben Sie den schrittweisen Weg von Kundenkontakten hin zu als Text formulierten Anforderungen.**
6. Was versteht man unter Stakeholdern?
7. Was ist ein Use Case, wie wird er dokumentiert?
8. Was ist der Unterschied und der Zusammenhang zwischen Business Use Cases und System Use Cases?
9. Wie kann man die Use Case –Dokumentation schrittweise verfeinern?
10. Beschreiben Sie die Modellierungsmöglichkeiten mit Aktivitätsdiagrammen.
11. Welche Probleme können auftreten, wenn Kunden Anforderungen formulieren?
12. Erklären Sie den Zweck und die Nutzungsmöglichkeiten der Anforderungsschablone nach Rupp.
13. Welche Arten von nicht-funktionalen Anforderungen kennen Sie? Nennen Sie Beispiele.
14. Was sind Lasten- und Pflichtenhefte, was steht drin, worin unterscheiden sie sich üblicherweise?

15. Nennen und beschreiben Sie die in einem Klassendiagramm möglicherweise enthaltenen Informationen.
16. Wie kann man aus einem Anforderungstext ein Klassendiagramm ableiten?
17. Was für verschiedene Arten von Beziehungen zwischen Klassen (also Verbindungen im Klassendiagramm) kann es geben, was bedeuten sie?
18. Wie können verschiedene Arten von Objekt-Sammlungen in der UML modelliert werden?
19. Was sind Sequenzdiagramme, wie sind sie aufgebaut, wofür werden sie eingesetzt?
20. Was sind Kommunikationsdiagramme, wozu werden sie eingesetzt?
21. Was versteht man unter Anforderungsverfolgung (Tracing), wozu ist sie da?
22. Was versteht man unter Boundary-Control-Entity.
23. Wie kann man die GUI-Entwicklung in die Entwicklung der Geschäftsklassen integrieren?
24. Welche Automatisierungsmöglichkeiten und Schwierigkeiten gibt es bei der Umsetzung von Klassendiagrammen in Quellcode?
25. Wie können Klassen und ihre unterschiedlichen Beziehungen in welchen Programmcode übersetzt werden?

Szenario: Modellieren Sie folgendes System (z. B. Projektverwaltung mit mehreren Projekten und Mitarbeitenden, Mitarbeitende arbeiten im Projekt, ein mitarbeitende Person macht die Projektleitung) als Klassendiagramm, schlagen Sie Modellierungsvarianten vor (z. B. mit Vererbung, Delegation, ...), wo würden Sie folgende Methoden (z. B. Mitarbeitende erzeugen, Mitarbeitende aus Projekt löschen) realisieren?

26. Welche besondere Bedeutung haben Schnittstellenklassen in der Software-Entwicklung?
27. Was versteht man üblicherweise unter einer Softwarearchitektur?

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing.-Wissenschaften und Informatik
- Software-Entwicklung -

Objektorientierte Analyse und Design

Wintersemester 2024
10. Aufgabenblatt

28. Was ist eine Mehrschichtenarchitektur, warum gibt es sie?
29. Welche Regeln und allgemeine Ansätze gibt es zur Bildung von Software-Paketen?
30. Wann heißen Software-Pakete abhängig, was bedeutet diese Abhängigkeit für die Praxis?
31. Wie kann man Abhängigkeiten zwischen Software-Paketen umdrehen?
32. In der Entwicklung wird häufig von verschiedenen Sichten auf das System gesprochen. Welche Sichten kennen sie, was ist ihre Bedeutung?
33. Welche Informationen können in einem Komponentendiagramm dargestellt werden?
34. Welche Informationen können in einem Einsatz- bzw. Verteilungsdiagramm (deployment diagram) dargestellt werden?
35. Was bedeutet es, dass eine Komponente eine Schnittstelle anbietet, wie kann dies realisiert sein?
36. Was versteht man generell unter Design-Pattern, warum sollten sie von Software-Entwicklern beherrscht werden?
37. Was versteht man unter „Design by Contract“?
38. Erklären Sie anschaulich den Model-View-Controller-Ansatz.
39. **Beschreiben Sie den Sinn und die Funktionsweise folgender Pattern mit Hilfe eines Klassendiagramms: Fassade, Beobachter (Oberserver Observable), Adapter, Singleton, Decorator, Proxy, Strategy, Command, Visitor.**
40. Nennen Sie typische GRASP-Pattern erklären Sie sie anhand von Beispielen.
41. Was ist Method Chaining?
42. Erklären Sie mit Beispielen den Aufbau des Redux-Patterns.
43. Welchen Sinn haben Factorys?
44. Wie funktioniert die typische Stream-Verarbeitung ab Java 8?
45. Was sind functional Interfaces in Java?
46. Was ist die Grundidee von Reflexion, welche Möglichkeiten werden geboten?
47. Was wird unter Metamodellierung bzgl. der UML verstanden?
48. Was versteht man unter Dependency Injection?
49. Wie werden Zustandsdiagramme in der Software-Entwicklung eingesetzt, wozu gibt es hierarchische Zustände, wie kann Nebenläufigkeit modelliert werden?
50. Was kann ein „Ereignis“ in Zustandsdiagrammen sein?
51. Welche Beschriftungen sind an Transitionen von Zustandsdiagrammen möglich?
52. Was versteht man und Microstep- und Macrostep-Semantik?
53. Welchen Sinn hat die Object Constraint Language?
54. Wie kann die typische Spezifikation einer Methode in der OCL aussehen?