

Вопросы ИК по предмету Проектирование систем ПО

1. Опишите основные понятия и цели проектирования программных систем.
2. Отличие проектирования от разработки программного обеспечения. Роль проектирования в жизненном цикле программного обеспечения.
3. Виды проектирования программных систем.
4. Что понимается под термином «архитектура программного обеспечения»?
5. Опишите требования заказчика на этапе проектирования.
6. Этапы процесса проектирования программных систем.
7. Итеративный процесс проектирования.
8. Приведите стандарты процесса проектирования программного обеспечения.
9. Приведите задачи проектирования программных систем.
10. Что такое модульность и почему она важна?
11. Что такое сцепление (coupling) и как оно влияет на проектирование программных систем?
12. Что такое связность (cohesion) и почему она важна для модульности и поддерживаемости системы?
13. Чем сцепление отличается от связности и как эти два понятия взаимосвязаны в контексте проектирования?
14. Приведите основные принципы модульного программирования.
15. Структурное программирование, и как оно связано с модульной структурой программы?
16. Модульная структура программы. Преимущества использования модульной структуры при разработке программного обеспечения.
17. Опишите основные структуры структурного программирования.
18. Какие основные правила следует соблюдать для поддержания чистоты и понятности кода при структурном программировании?
19. Опишите функциональные и нефункциональные требования.
20. Спецификация требований.
21. Методы для анализа требований перед проектированием.

22. Опишите этапы процесса разработки системных и функциональных требований.
23. Приведите цели и задачи документации по системным и функциональным требованиям.
24. Влияние системных и функциональных требований на разработку системы.
25. Системные требования и их основные компоненты.
26. Особенности системных требований к интерфейсам.
27. Методы и инструменты для документирования и анализа системных требований.
28. Функциональные требования.
29. Влияние функциональных требований на выбор архитектуры и технологий проекта.
30. Этапы жизненного цикла программного обеспечения.
31. Модель «водопад» (Waterfall). Преимущества и недостатки этой модели.
32. Особенности гибких методологий (Agile).
33. Модель V-цикла. Преимущества и недостатки модели.
34. Водопадная модель. Преимущества и недостатки модели.
35. Спиральная модель разработки. Преимущества и недостатки модели.
36. Инкрементная модель разработки. Преимущества и недостатки модели.
37. UML (Unified Modeling Language — унифицированный язык моделирования).
38. Виды диаграмм UML.
39. Диаграмма классов и его основные элементы.
40. Диаграмма последовательностей и как она помогает проектировать системы?
41. Диаграмма вариантов использования и как он связан с моделированием поведения?
42. Диаграмма состояний и его основные элементы.
43. Диаграмма компонентов и его основные элементы.

44. Диаграмма развертывания и его основные элементы.
45. Инструменты для создания UML-диаграмм.
46. ER-диаграмма (Entity-Relationship, «сущность — связь») и как она применяется при проектировании баз данных?
47. Моделирование данных и какие цели оно преследует?
48. Опишите основные уровни моделирования данных.
49. ER-модель (Entity-Relationship model, модель "сущность-связь") и где она используется?
50. Опишите этапы процесса моделирования данных.
51. Инструменты для создания моделей данных.
52. Моделирование поведения в проектировании программных систем.
53. Виды диаграмм для моделирования поведения.
54. Преимущества рефакторинг кода.
55. Шаблоны (паттерн) проектирования.
56. Основные категории паттернов проектирования?
57. Преимущества использования паттернов проектирования.
58. Порождающие паттерны и область их применения.
59. Шаблон (паттерн) Singleton?
60. Шаблон (паттерн) Factory Method. В каких случаях лучше применять Factory Method?
61. Шаблон (паттерн) Abstract Factory. Как реализовать Abstract Factory в реальном проекте?
62. Шаблон (паттерн) Builder.
63. Шаблон (паттерн) Prototype.
64. Структурные шаблоны (паттерны) и область их применения.
65. Шаблон (паттерн) паттерн Adapter.
66. Шаблон (паттерн) Bridge.
67. Шаблон (паттерн) Composite.

- 68.Шаблон (паттерн) Decorator. Как использовать паттерн Decorator для добавления функциональности?
- 69.Шаблон (паттерн) Facade. Какие задачи решает паттерн Facade?
- 70.Шаблон (паттерн) Flyweight.
- 71.Шаблон (паттерн) Proxy.
- 72.Поведенческие шаблоны (паттерны) и область их применения.
- 73.Шаблон (паттерн) Chain of Responsibility.
- 74.Шаблон (паттерн) Command.
- 75.Шаблон (паттерн) Interpreter. Какие задачи решает паттерн Interpreter?
- 76.Шаблон (паттерн) Iterator.
- 77.Шаблон (паттерн) Mediator.
- 78.Шаблон (паттерн) Memento.
- 79.Шаблон (паттерн) Observer.
- 80.Шаблон (паттерн) State. Какие задачи решает паттерн State?
- 81.Шаблон (паттерн) Strategy.
- 82.Шаблон (паттерн) Template Method.
- 83.Шаблон (паттерн) Visitor. Какие задачи решает паттерн Visitor?
- 84.Метрики проектирования, и их основная цель.
- 85.Типы метрик для оценки качества проектирования программного обеспечения.
- 86.Метрика сложности системы (Cyclomatic Complexity).
- 87.Метрики для оценки масштабируемости программной системы.
- 88.Метрики для оценки модульности системы.
- 89.Статический анализ кода.
- 90.Основная цель статического анализа кода в процессе разработки
- 91.Инструменты для статического анализа кода.
- 92.Методы для оценки качества проектных решений.
- 93.Метрики производительности для оценки решений по проектированию.
- 94.Конфиденциальность данных пользователей при проектировании.

- 95. Важность использования существующих алгоритмов при разработке программ.
- 96. Качественные характеристики решения проектных задач.
- 97. Методы и подходы для оценки качественных характеристик решений.
- 98. Роль надежности в качественных характеристиках проектного решения.
- 99. Методы для оценки отказоустойчивости систем при проектировании.
- 100. Метрики для оценки долговечности и надежности проектных решений.