

Вопросы к экзамену «Проектирование ПО»

Юрий Литвинов
yurii.litvinov@gmail.com

1. Понятие архитектуры, профессия «Архитектор».
2. Архитектурные виды.
3. Роль архитектуры в жизненном цикле ПО.
4. Пример архитектуры: Apache Hadoop. Prescriptive и descriptive-архитектура.
5. Понятие декомпозиции. Модульность, связность, сопряжение, сложность.
6. Понятия класса и объекта, абстракция, инкапсуляция, наследование.
7. Принципы выделения абстракций предметной области.
8. Принципы SOLID.
9. Закон Деметры. Принципы хорошего объектно-ориентированного кода.
10. Моделирование, визуальные модели, виды моделей, метафора визуализации.
11. Язык UML. Проектирование структуры системы, диаграммы классов.
12. Диаграммы объектов, диаграммы пакетов UML.
13. Диаграммы компонентов, диаграммы развёртывания UML.
14. Диаграмма случаев использования UML.
15. Диаграмма активностей UML.
16. Диаграммы конечных автоматов UML.
17. Диаграммы последовательностей UML.
18. Диаграммы коммуникации UML.
19. Диаграммы составных структур, коопераций, временные диаграммы.
20. Диаграммы обзора взаимодействия, диаграммы потоков данных.
21. Диаграммы IDEF0, характеристик. Feature tree
22. Моделирование требований в SysML.
23. Язык BPMN.
24. Моделирование данных: диаграммы «Сущность-связь».
25. Концептуальное моделирование, диаграммы ORM.

26. Сети Петри, их анализ и применение.
27. Понятие и примеры CASE-систем.
28. Паттерн «Компоновщик».
29. Паттерн «Декоратор».
30. Паттерн «Стратегия».
31. Паттерн «Адаптер».
32. Паттерн «Заместитель».
33. Паттерн «Фасад».
34. Паттерн «Приспособленец».
35. Паттерн «Мост».
36. Паттерн «Фабричный метод».
37. Паттерн «Абстрактная фабрика».
38. Паттерн «Одиночка».
39. Паттерны «Ленивая инициализация» и «Пул объектов».
40. Паттерн «Прототип».
41. Паттерн «Строитель».
42. Паттерн «Посредник».
43. Паттерн «Команда».
44. Паттерн «Цепочка ответственности».
45. Паттерн «Наблюдатель».
46. Паттерн «Состояние».
47. Паттерн «Шаблонный метод».
48. Паттерн «Посетитель».
49. Паттерн «Хранитель».
50. Паттерн «Интерпретатор».
51. Паттерн «Итератор».
52. Антипаттерны «Круговая зависимость», «Последовательная связность», «Вызов предка», «Проблема Йо-Йо».

53. Антипаттерны «Активное ожидание», «Соккрытие ошибки», «Магические числа», «Магические строки».
54. Антипаттерны «Божественный объект», «Поток лавы».
55. Антипаттерны «Функциональная декомпозиция», «Полтергейст», «Золотой поток».
56. Понятие архитектурного стиля, трёхзвенная архитектура.
57. Model-View-Controller, Sense-Compute-Control.
58. Слоистый стиль, «Клиент-сервер».
59. Гексагональная архитектура, луковая архитектура.
60. Чистая архитектура.
61. Пакетная обработка, каналы и фильтры, Blackboard.
62. Событийно-ориентированные стили, Publish-Subscribe.
63. Peer-to-peer.
64. Понятие Domain-Driven Design, единый язык, изоляция предметной области.
65. DDD, основные структурные элементы модели предметной области.
66. DDD, паттерн «Агрегат».
67. DDD, паттерны «Фабрика», «Репозиторий».
68. Паттерн «Спецификация».
69. Ограниченный контекст, непрерывная интеграция, карта контекстов.
70. Подходы к интеграции контекстов.
71. Смысловое ядро, приёмы дистилляции, абстрактное ядро.
72. Крупномасштабная структура, метафора системы, разбиение по уровням.
73. Типичные уровни в производственных и финансовых системах.
74. Стили «Уровень знаний», «Подключаемые компоненты».
75. Понятие распределённой системы, заблуждения при проектировании распределённых систем.
76. Межпроцессное сетевое взаимодействие, модель OSI, стек протоколов TCP/IP, сокет, протоколы «запрос-ответ», протокол HTTP.
77. RPC, RMI. Пример: gRPC.
78. Веб-сервисы, SOAP. WCF.

79. Очереди сообщений, RabbitMQ.
80. Архитектурные стили распределённых приложений: Big Compute, Big Data.
81. Web-queue-worker, N-звенная архитектура.
82. Микросервисная архитектура.
83. Архитектурный стиль REST.
84. Принципы дизайна распределённых приложений: самовосстановление, избыточность.
85. Принципы дизайна распределённых приложений: минимизация координации, проектирование для обслуживания.
86. Docker, Docker Compose.
87. Kubernetes.
88. Облачная инфраструктура, AWS, Terraform.