**Составить вопросы к собеседованию кандидата в соответствии с требованиями профессионального стандарта "Инженер-программист".**

1. Часто говорят, что одна из самых важных задач объектно-ориентированного дизайна (и организации кода в целом) — это сильное сцепление и слабая связанность. Что это значит? Почему они так важны и как их достигнуть?
2. Почему в большинстве языков индекс массива начинается с нуля?
3. Как тесты и TDD влияют на организацию кода?
4. Напишите фрагмент кода, нарушающий принцип Don't Repeat Yourself (DRY). Объясните, почему у него плохая организация, и исправьте.
5. В чём разница между сцеплением и связанностью?
6. Для чего полезен рефакторинг?
7. Полезны ли комментарии в коде? Кто-то говорит, что их следует по возможности избегать, а в идеале они вообще не нужны. Вы согласны?
8. Какая разница между организацией кода и архитектурой?
9. Почему в TDD тесты пишутся прежде кода?
10. C++ поддерживает множественное наследование, а Java позволяет классу реализовать несколько интерфейсов. Какое влияние оказывает использование этих средств на ортогональность? Существует ли разница во влиянии между использованием множественного наследования и множественных интерфейсов? Есть ли разница между использованием делегирования и наследования?
11. Плюсы и минусы хранения логики предметной области в хранимых процедурах.
12. По вашему мнению, почему объектно-ориентированная организация столько лет доминирует на рынке?
13. Если у вашего кода плохая организация, как вы это поймёте?
14. Расскажите о трёх главных недостатках вашего любимого языка программирования.
15. Почему возрастает интерес к функциональному программированию?
16. Что такое замыкание и для чего оно полезно? Что общего у замыканий и классов?
17. Почему родные куки и сторонние куки обрабатываются настолько по-разному?
18. Как бы вы управляли нумерацией версий Web Services API?
19. С точки зрения бэкенда есть ли какие-нибудь недостатки или препятствия внедрению одностраничных приложений?
20. Почему мы обычно так сильно стремимся, чтобы сервис был stateless? Что такого хорошего в stateless-коде? Почему и когда хранение состояния (statefullness) — это плохо?
21. REST и SOAP: в каких ситуациях вы выберете один из этих подходов, а в каких другой?
22. В веб-разработке подходы Model-View Controller и Model-View-View-Model во многом похожи как в бэкенде, так и во фронтенде. Что они из себя представляют и почему целесообразно их использовать?
23. Как вы перенесёте приложение из одной базы данных в другую, например, из MySQL в PostgreSQL? Если вам придётся управлять таким проектом, какие проблемы вы ожидаете?
24. Как реализовать ленивую загрузку? Когда она полезна? Какие есть подводные камни?
25. Один из методов непрерывной интеграции называется сине-зелёный деплой. Он заключается в использовании двух сред производственной эксплуатации, максимально похожих — и деплой в одну из них во время работы другой, а затем безопасное переключение трафика из одной среды на вторую после тестирования в удобной обстановке. Этот метод усложняется, если деплой предполагает изменение структуры или содержимого БД. Хотелось бы обсудить с вами эту тему.
26. Почему осуществление ветвлений в Mercurial или Git легче, чем в SVN?
27. Каковы плюсы и минусы распределённых систем контроля версий, таких как Git, в сравнении с централизованными VCS вроде SVN?
28. Можете описать рабочий процесс GitHub Flow и GitFlow?
29. Что такое rebase (перемещение)?
30. Почему слияние проще осуществлять в Mercurial или Git, чем в SVN и CVS?
31. Зачем вообще нужен параллелизм? Объясните.
32. Почему так трудно тестировать многопоточный/распараллеленный код?
33. Что такое состояние гонки? Напишите пример на произвольном языке.
34. Что такое взаимная блокировка? Вы можете написать какой-нибудь код, страдающий взаимными блокировками?
35. Что такое ресурсный голод (process starvation)? Если нужно, давайте рассмотрим его определение.
36. Что представляет собой алгоритм неблокирующей синхронизации без ожиданий (wait-free алгоритм)?
37. Что такое гибкость (agility)?
38. Как вы будете работать с легаси-кодом?
39. Скажем, я ваш менеджер проекта и не эксперт в программировании. Попробуйте объяснить мне, что такое легаси-код и почему нужно заботиться о качестве кода.
40. Я исполнительный директор вашей компании. Объясните мне Канбан и убедите вложиться в него.