ООП | Вопросы к экзамену

1. Объясните, как семантическое версионирование (SemVer) влияет на управление зависимостями в коммерческом проекте. Назовите основные файлы в git проекте. Приведите пример конфликта версий в Python-проекте с pyproject.toml и обоснуйте решение через MAJOR.MINOR.PATCH.

2. Сравните Git Flow и Trunk-Based Development для Open-Source проекта. При каких условиях (размер команды, частота релизов) Git Flow становится менее эффективным?

3. Опишите как выбирать цель проекта. Что такое SMART оценка?
   Приведите примеры 3 таких целей и обоснуйте

4. Какие виды проектов существуют в рамках программирования?
   Приведите их особенности. Как у данных проектов строится целеполагание?

5. На чем основывается выбор языка? Назовите 5 основных пунктов и опишите их важность.

6. Что такое git flow? Опишите процесс разработки проекта с git flow. Какие ветки бывают?
   Что такое Pull Request и зачем он нужен?

7. Какие инструменты сборки C++ бывают? Назовите 4 варианта.
   Опишите их принцип работы, ограничения, достоинства и недостатки.

8. Какие факторы учитывать при выборе между std::vector и std::deque для многочисленных заполнений данных? Обоснуйте с учетом локальности данных и алгоритмической сложности

9. Какие контейнеры используют древовидную структуру? Объясните их основные особенности

10. Перечислите основные STL контейнеры, их особенности, применимость, достоинства и недостатки

11. Что такое Big O нотация? Зачем она нужна, и как её применять в коде.
    Приведите примеры разных нотаций

12. Что такое Big O нотация? Зачем она нужна, и как её применять в коде.
    Как оценивать сложность алгоритмов?

13. Чем полезны STL контейнеры и как они улучшают код? Обоснуйте свой ответ

14. Сравните сортировку слиянием и introsort для обработки больших данных (10,000+ элементов)

15. Опишите процесс интеграции C++20 Ranges в legacy код с raw-циклами. Какие технические проблемы возникают при работе с views::filter и не итерируемыми типами?

16. Проанализируйте сценарий, где бинарный поиск (std::binary_search) даёт ложный результат для частично отсортированного диапазона

17. Какие парадигмы программирования существуют?
    Опишите их, приведите примеры кода (достаточно псевдокода), их основные различия

18. Что такое функциональное программирование? Какая идеология, недостатки и достоинства?
    Как интегрировать с ООП?

19. Что такое процедурное программирование? Какая идеология, недостатки и достоинства?
    Чем отличается от функционального программирования?

20. Как и зачем интегрировать ООП с другими парадигмами? Что мы от этого выигрываем, чем жертвуем?

21. Как принцип Liskov (LSP) влияет на проектирование иерархии исключений в C++? Приведите пример нарушения LSP при переопределении виртуальной функции (virtual void save(File&)).

22. Какие виды полиморфизма существуют в C++? Приведите примеры и назовите их особенности

23. Как работает перегрузка операторов в C++? Какие существуют ограничения? Чем перегрузка помогает?

24. Проанализируйте влияние глубины иерархии наследования (6+ уровней) на скорость вызова виртуальных методов

25. Какой вид управления памятью в C++? Объясните принцип работы RAII на примере управления собственным классом-оберткой (напр., ArrayWrapper)

26. Какие основные проблемы ручного управления памятью решает парадигма RAII? Приведите примеры кода (псевдокод), иллюстрирующие проблемы

27. Сформулируйте «Правило 3» и «Правило 5» в C++. В каких случаях они применяются? Приведите примеры классов, где нарушение этих правил приводит к проблемам, и их корректные реализации.

28. Для чего используются умные указатели - std::unique_ptr , std::shared_ptr , std::weak_ptr? Опишите семантику владения каждого типа и приведите сценарии их использования.

29. Что такое «специализация шаблонов»? Объясните разницу между полной (explicit) и частичной (partial) специализацией. Приведите примеры кода

30. Как CRTP (Curiously Recurring Template Pattern) позволяет реализовать статический полиморфизм без накладных расходов на виртуальные функции? Приведите пример

31. Опишите технику «Выражения шаблонов» (Expression Templates). Какую проблему она решает (на примере операции сложения матриц)? Как она связана с оптимизацией и ООП?

32. Что такое «идеальная передача» (perfect forwarding) и как она реализуется с помощью std::forward? Какой выигрыш она дает при использовании в шаблонных функциях создания объектов?

33. Каковы ключевые концепции «Пост-ООП» (Post-OOP)? Приведите примеры их реализации (например, в Rust и Go)

34. Сравните реализацию обобщенного программирования (Generics/шаблоны) в C++, Rust и Go. Укажите различия в синтаксисе, механизмах ограничений (концепты, трейты, интерфейсы), производительности (мономорфизация vs боксы) и гибкости.

35. Объясните разницу между процессом и потоком с точки зрения выделяемых ОС ресурсов и адресного пространства. Приведите примеры ресурсов, общих для потоков одного процесса, и ресурсов, уникальных для каждого потока.

36. Каковы основные проблемы, возникающие при переходе от синхронного однопоточного к асинхронному многопоточному приложению? Опишите ограничения каждого подхода.

37. Что такое «гонка данных» (Data Race)? Приведите конкретный пример и объясните, почему результат работы программы непредсказуем.

38. Опишите проблему «условия гонки» (Race Condition). Чем она принципиально отличается от «гонки данных»? Приведите пример и объясните, как std::condition_variable решает эту проблему.

39. Что такое «взаимная блокировка» (Deadlock)? Приведите пример, демонстрирующий deadlock. Опишите методы решения проблемы

40. Какие типы мьютексов представлены в стандартной библиотеке C++?
    Для каждого типа приведите конкретный сценарий использования

41. Объясните назначение и принцип работы умных указателей (std::unique_ptr, std::shared_ptr) в контексте управления памятью в многопоточных программах. Какие проблемы ручного управления они решают?

42. Как ключевое слово volatile взаимодействует с многопоточностью? Почему его использования недостаточно и какие механизмы нужно применять дополнительно?

43. Опишите механизм передачи исключений из рабочего потока в основной поток с использованием std::promise и std::future. Почему это важно в многопоточном программировании?

44. Сравните подходы к обеспечению безопасности памяти и параллелизма в C++, Rust и Go.
    Какие ключевые механизмы использует каждый язык и какие гарантии они дают?

45. Что такое «Антипаттерн»? Опишите антипаттерн «Божественный объект» (God Object).
    Приведите пример кода, демонстрирующий этот антипаттерн, и его рефакторинг с разделением ответственности.

46. Объясните принцип «Разделения обязанностей» (Separation of Concerns) в контексте паттернов.
    Приведите пример, как этот принцип реализуется с помощью паттерна «Фасад» (Facade).

47. В чем разница между паттернами «Фабричный метод» (Factory Method) и «Абстрактная фабрика» (Abstract Factory)? Приведите примеры их использования

48. Опишите назначение и структуру паттерна «Строитель» (Builder).
    Как он помогает создавать сложные объекты? Приведите пример

49. Какие проблемы решает паттерн «Наблюдатель» (Observer)? Приведите пример его использования. Как он реализует принцип «Ослабления связей» (Loose Coupling)?

50. В чем суть паттерна «Стратегия» (Strategy)? Приведите пример.
    Как он связан с принципом «Повторное использование кода» (Reusability)?

51. Объясните разницу между паттернами «Декоратор» (Decorator) и «Адаптер» (Adapter).
    Для каких целей используется каждый? Приведите примеры

52. Что такое «Состояние» (State) как паттерн? Приведите пример. Как он помогает управлять поведением объекта в зависимости от внутреннего состояния?

53. Опишите цикл разработки TDD (Test-Driven Design). Какие преимущества он дает? Как TDD связан с паттернами проектирования? Приведите пример тестирования паттерна «Фабричный метод» с использованием TDD.

54. Почему чистый код не всегда равен качественному? Приведите примеры компромиссов (например, читаемость vs производительность в C++).

55. Сравните Clang-Tidy и Cppcheck для C++. Какие типы ошибок они обнаруживают?
    Как настроить кастомные правила именования переменных в Clang-Tidy?

56. Как принцип единственной ответственности (SRP) и инверсии зависимостей (DIP) применяются вне ООП? Приведите пример модуля, нарушающего SRP.

57. Дайте определение «жесткости», «хрупкости», «вязкости». Как связана жесткость с явным указанием типов? Покажите на примере кода

58. Что такое Long Method, Large Class, Data Clumps? Как списковые включения в Python помогают бороться с дублированием кода?

59. Объясните, когда стоит рефакторить код по «правилу трех».
    Приведите пример замены «магических чисел» константами и выделения метода.

60. Как Dependency Injection улучшает тестируемость? Покажите на примере кода.
    Как использовать Mock-объекты для тестов?

61. Какие правила именования переменных/функций улучшают читаемость?
    Сравните calc(a, b) и calculate_area(width, height).

62. Какие 4 шага рефакторинга приоритетны для старого кода (например, удаление мертвого кода)?
    Объясните на примере

63. Объясните ключевые принципы модульной архитектуры (декомпозиция, слабая связанность, интерфейсы). Приведите пример организации модулей в C++ с использованием пространств имен.

64. Сопоставьте концепции ООП (инкапсуляция, полиморфизм, наследование) с их аналогами в модульной архитектуре. Как модули решают проблему жестких зависимостей?

65. Зачем передавать зависимости извне модуля? Покажите на примере кода (Python/C++), как инъекция зависимостей улучшает тестируемость и гибкость модуля.

66. Перечислите основные ошибки в модульной архитектуре. Приведите примеры кода для каждой и объясните, как их избежать.

67. Как идиома PIMPL реализует инкапсуляцию? Опишите механизм работы на примере. Какие преимущества она дает для скорости компиляции?

68. Чем модули C++20 отличаются от заголовочных файлов? Объясните роль ключевых слов module, export, import. Как модули решают проблему «ужаса включений»?

69. Назовите характеристики RESTful API. Как оформить эндпоинты для сущности «пользователь» в C++?

70. Сравните JSON и MessagePack. В каких сценариях предпочтительнее бинарный формат?
    Приведите пример структуры данных в обоих форматах.

71. Какие коды относятся к категориям 2xx, 4xx, 5xx? Приведите примеры кода для ошибок 400 Bad Request, 403 Forbidden, 503 Service Unavailable.