

Вопросы к экзамену

1. Модель памяти Python
2. Синтаксис языка
3. Где и зачем используются исключения?
4. Что будет если не обработать исключение?
5. Как работать с файлами в стиле Python?
6. Что такое комментарий в Python?
7. Что такое ООП? Пример.
8. Принципы ООП.
9. Отношения между классами.
10. Паттерны проектирования и их виды. Зачем они нужны? Примеры паттернов.
11. SOLID.
12. Встроенные функции для работы с классами.
13. Методы `__init__` и `__new__`
14. Понятие итератора и генератора и их отличия?
15. Зачем нужны генераторы
16. Что такое декоратор? Зачем он нужен?
17. Механизм генерации коллекций. Какие особенности? В чем отличие от генераторов?
18. Пакеты и модули отличия
19. Как импортировать модуль
20. Зачем нужен `__name__` и `__all__`
21. Общая схема импорта
22. Модуль `collections`. Примеры специальных контейнеров.
23. Что такое регулярные выражения? Модуль `re`. Примеры.
24. Что такое сериализация данных? Примеры форматов хранения данных `json`, `xml`. Модуль `pickle`.
25. Модуль для работы со случайными числами.
26. Чем отличается поток и процесс в Python, когда лучше использовать тот или иной подход?
27. Какие средства межпроцессного взаимодействия Вы знаете?
28. Примитивы синхронизации
29. Зачем нужна синхронизация?
30. Что такое `race condition` и `deadlock`. Как бороться?
31. Зачем нужны библиотеки `numpy`, `pandas`, `matplotlib`, `seaborn`? Когда лучше использовать какую-то из них?
32. Из чего состоит график `matplotlib`?
33. Какие способы воздействовать на этот график?
34. Зачем нужен `seaborn`?

- 35. Уровни тестирования, типы тестирования
- 36. Git
- 37. Стили документации кода Python
- 38. Системы генерации документации
- 39. План EDA и примеры.
- 40. Переобучение и недообучение.
- 41. Отложенная выборка и кросс-валидация. Оценка классификации и регрессии.
- 42. Решающие деревья, метод опорных векторов, линейная регрессия, логистическая регрессия.