

Вопросы и задачи к экзамену

Билет включает в себя 3 задания: 2 из раздела «Вопросы» и 1 из раздела «Задачи». Условия задач в билетах могут быть изменены с сохранением темы и типа задачи.

Вопросы

0. Определение понятий «данные», «программа», «алгоритм».
1. Основные этапы разработки программы.
2. Понятие типа данных. Система типов и типизация в языке Python. Как определить тип значения во время выполнения программы?
3. Понятия «переменная», «функция». Соглашения по именованию переменных и функций в языке Python.
4. Встроенные функции преобразования типов в языке Python.
5. Пустой тип.
6. Логический тип, операции над ним. Какие значения других типов в Python интерпретируются как логические значения? Какие значения других типов в этом случае соответствуют логическим значениям True и False?
7. Целочисленный тип. Операции над целыми числами в языке Python. Приоритет операций.
8. Тип «число с плавающей запятой». Основные операции. Приоритет операций.
9. Строчный тип. Строки как последовательности, основные операции. Код символа.
10. Строчный тип. Методы форматирования строк.
11. Строчный тип. Способы разделения строки на подстроки по подстрокам-разделителям.
12. Строчные литералы в Языке Python. Символ \ в строках.
13. Методы класса str для определения класса символа.
14. Последовательности в языке Python и операции над ними.
15. Поиск элемента в последовательности. Способ ускорения поиска.
16. Сортировка: назначение, средства языка, пример.
17. Алгоритм сортировки методом пузырька.

18. Тип “словарь”, назначение, основные операции и методы.
19. Абстрактный тип данных “стек”, основные операции, примеры применения.
20. Абстрактный тип данных “очередь”, основные операции, примеры применения.
21. Операции сравнения в языке Python.
22. Оператор `in`: особенности применения к строкам, спискам, кортежам, словарям, множествам.
23. Управляющие конструкции языка Python.
24. Циклы с переменной. Использование встроенной функции `range`. Ключевые слова `break` и `continue`.
25. Циклы с предусловием. Ключевые слова `break` и `continue`. Способ организовать цикл с постусловием.
26. Средства для организации ветвлений.
27. Функции и их аргументы. Способы определения функций в программах на Python. Подстановка аргументов при вызове функции и возврат значения в программу.
28. Функции с переменным числом аргументов и значения аргументов функции по умолчанию в языке Python.
29. Функции высшего порядка. Применение функций высшего порядка для обработки последовательностей. Встроенные функции высшего порядка для обработки последовательностей.
30. Списковые выражения (`list comprehensions`), примеры.
31. Способы организации повторяющихся вычислений на языке Python. Примеры.
32. Объектно-ориентированное программирование: классы, объекты, свойства, методы.
33. Определение класса. Подстановка аргументов методов и конструкторов класса.
34. Переопределение методов класса и операторов над объектами класса. Пример.
35. Исключения и их обработка в программах на языке Python.
36. Файл, файловая система, путь к файлу. Абсолютные и относительные пути. Отличия путей в ОС Windows и ОС Linux, MacOS X. Особенности записи пути к файлу в программах.
37. Файловые объекты: назначение, создание, основные методы.
38. Средства ввода-вывода в языке Python.

39. Способы ввода данных в программу и вывода результатов вычислений из программы.
40. Блок-схема, назначение, основные типы блоков и их соответствие конструкциям языка Python. Пример.
41. Формат файлов CSV, назначение. Средства стандартной библиотеки языка Python для работы с файлами в формате CSV.
42. Модуль `shelve` из стандартной библиотеки языка Python: назначение, основные операции, пример использования.
43. Встроенная функция `eval`, назначение, примеры применения.
44. Модульная организация программы на языке Python.

Примеры задач

0. Начертите блок-схему и напишите код функции на языке Python, которая принимает последовательность чисел a_1, a_2, \dots, a_n и возвращает среднее квадратичное этих чисел, вычисленное по формуле:

$$s = \sqrt{\frac{a_1^2 + a_2^2 + \dots + a_n^2}{n}}$$

Напишите фрагмент кода, демонстрирующий работу этой функции.

1. Напишите определение класса на языке Python, который предназначен для хранения таблицы размером $m \times n$ ячеек и включает в себя методы для:
 - записи значения в указанную ячейку таблицы,
 - чтения значения из указанной ячейки таблицы,
 - получения всех значений заданной строки таблицы в виде списка,
 - получения всех значений заданного столбца таблицы в виде списка.
2. В программе на языке Python определен класс, который реализует абстрактный тип данных "очередь" и имеет метод `enqueue` для добавления элемента в конец очереди и `dequeue` для получения элемента с конца очереди. Предложите набор юнит-тестов для этого класса.
3. Напишите код предиката на языке Python, который проверяет, делится ли целое число на 2 и 3 без остатка. Предусмотрите ситуацию, когда предикату передано значение, не являющееся целым числом, программа в этом случае не должна аварийно завершаться.

4. Начертите блок-схему и напишите код программы на языке Python, которая читает из тестового файла слова, разделенные пробельными символами, и выводит в другой файл: самое длинное слово из исходного файла на отдельной строке, самое короткое слово из исходного файла на отдельной строке.
5. Напишите код программы на языке Python, которая вычисляет таблицу значений функции одной переменной $f(x)$ на интервале $x[a, b]$ с приращением Δx .

Программа запрашивает у пользователя:

- Функцию в виде выражения на языке Python, содержащую переменную x . Например, если требуется вычислить значения функции $f(x) = 2x^2$, пользователь вводит выражение `2 * x ** 2`,
- Начальное значение аргумента функции,
- Конечное значение аргумента функции,
- Приращение аргумента.

Затем программа выводит в консоль таблицу пар значений “аргумент – значение функции”.

Что нужно предпринять, чтобы перенаправить вывод программы в текстовый файл?

Предложите набор входных данных, которые вызовут ошибку и аварийное завершение программы. Поясните, почему это произойдет.

6. Начертите UML-диаграммы, отражающие отношения между сущностями “студент”, “курс”, “группа”, “учебная дисциплина”, “преподаватель”.
7. В программе определена функция:

```
def trib(n):  
    if n in [0, 1]: return 0  
    elif n in [2, 3]: return 1  
    else:  
        return trib(n-1) + trib(n-2) + trib(n-3)
```

Каким будет результат выражения `[trib(a) for a in range(3, 6)]`?

8. Как на языке Python можно реализовать представления множеств, не используя встроенный класс `set`? Напишите функции для выполнения основных операций над множествами, представленных предложенным вами способом.