

КОНТРОЛЬНЫЕ ВОПРОСЫ для подготовки к ЭКЗАМЕНУ

Часть 1

1. Свойства и особенности языка программирования Python. Стандартная библиотека Python. Концептуальная структура программы. Файловая структура программы.
2. Python как интерпретируемый язык программирования. Трансляция в байт-код. Виртуальная машина Python. Расширения Python.
3. Объект. Встроенные типы объектов. Литералы и конструкторы для создания объектов встроенных типов. Изменяемые и неизменяемые типы. Булево значение объекта.
4. Последовательность. Встроенные типы объектов последовательностей. Операции над последовательностями: индексация, конкатенация, повторение, длина, проверка принадлежности.
5. Словарь. Создание словаря. Операции над словарем. Методы словаря.
6. Списковое включение и его аналоги. Списковое включение с конструкцией выбора. Списковые включения с несколькими уровнями вложенности. Примеры.
7. Файловый объект. Создание файлового объекта. Методы файловых объектов. Итерирование файлового объекта.
8. Строки документации и комментарии. Правила оформления строк документации. Атрибут `__doc__`. Инструмент **PyDoc**.
9. Синтаксический шаблон составного оператора. Оператор выбора **if**. Оператор сопоставления **match**. Операторы цикла **for** и **while**.
10. Переменная, ее создание, использование и удаление. Оператор присваивания и его формы. Неявные присваивания. Пространства имен.
11. Пользовательская функция. Оператор определения функции **def**. Определение функции со значениями по умолчанию для аргументов. Определение функции с произвольным количеством аргументов.
12. Позиционные и ключевые аргументы при вызове функции. Позиционный и ключевой режимы сопоставления объектов-значений и аргументов-переменных при вызове функции.
13. Объект функции. Аннотация типов при определении функции. Определение и применение **lambda**-функций. Примеры.
14. Переменная. Типы областей видимости переменной. Правило лексических областей при поиске переменной. Операторы объявления **global** и **nonlocal**.
15. Механизм итераций. Итерируемый объект и его элементы. Объект итератора.
16. Механизм итераций. Ручное и автоматическое выполнение итераций. Итерационные инструменты. Протокол итераций.
17. Генераторный объект. Генераторное выражение. Генераторная функция. Операторы **yield** и **yield from**. Преимущества генераторных объектов перед коллекциями.

18. Функции **map** и **filter** для организации циклов на итерируемых объектах. Примеры.
19. Спецификации функций **enumerate** и **zip**. Использование функций **enumerate** и **zip** в циклах **for**. Использование функции **zip** для работы со словарями.
20. Функция-замыкание. Фабричная функция. Пример определения функции-замыкания со счетчиком вызовов функции.
21. Модуль. Пакет. Операторы импортирования модуля **import** и **from**. Загрузка и перезагрузка модуля.
22. Модуль. Глобальная область видимости переменной. Объект модуля. Пространство имен объекта модуля. Атрибут **__name__** объекта модуля.
23. Оператор создания класса **class**. Объект класса. Экземпляр класса. Атрибуты и методы класса. Объект связанного метода. Концепция инкапсуляции. Метод инициализации экземпляра класса.
24. Дерево наследования. Суперклассы. Подклассы. Множественное наследование. Настройка кода за счет замены метода. Настройка кода за счет расширения кода метода.
25. Перегрузка операций и встроенных функций. Методы строкового представления. Метод вызова. Методы операции сложения.
26. Перегрузка операций и встроенных функций. Методы операций сравнения. Методы булевого значения. Методы операций индексации.
27. Исключение. Операторы для работы с исключениями. Стандартный обработчик исключений. Встроенные исключения. Пользовательские исключения. Диспетчер контекста.
28. Исключение. Оператор **try** для перехвата и обработки исключений. Операторы **raise** и **assert** для генерации исключений.
29. Расширение **numpy**. Тип данных **ndarray**. Атрибуты массива. Итерирование массивов. Создание массивов. Индексация массивов.
30. Расширение **numpy**. Методы для изменения структуры массива. Атрибут **shape**. Представление и копия массива. Атрибут **base**.
31. Расширение **numpy**. Векторизация вычислений. Представление вектора и матрицы с помощью массива. Матричное произведение. Скалярное произведение. Транспонирование матрицы.
32. Пакет **matplotlib**. Базовые функции двумерной графики. Функции для создания и оформления текстовых объектов в графической области.
33. Пакет **matplotlib**. Графические объекты для представления изображения. Иерархическая структура изображения. Объектно-ориентированный подход для построения изображения.

Часть 2

34. Приведите спецификации функций, необходимых для создания анимации, которая содержит два статических объекта – произвольную кривую на плоскости и точку А кривой, и два динамических объекта – точку В кривой, отличную от точки А и стремящуюся к ней вдоль кривой, а также секущую прямую, проходящую через точки А и В.
35. Определение правильного многоугольника Рёло. Определение ширины правильного многоугольника Рёло. Решите математическую задачу вычисления радиуса R окружности, описанной вокруг правильного n-угольника, если задана ширина r правильного

многоугольника Рёло, построенного на окружностях, центры которых лежат в вершинах этого n-угольника.

36. Определение правильного многоугольника Рёло. Определение ширины правильного многоугольника Рёло. Получите аналитическое выражение для описания стороны правильного n-угольника Рёло при условии, что заданы его ширина r и координаты его вершин `vertices`. Напишите код для вычисления стороны n-угольника Рёло в виде массива координат точек, принадлежащих этой границе. Количество точек N для описания одной стороны является заданным.
37. Определите понятия поступательного и вращательного движений тела. Используя эти понятия, определите процесс качения треугольника Рёло по квадрату. Сформулируйте допущения о поступательном и вращательном движениях треугольника Рёло, принимаемые при моделировании процесса его качения по квадрату.
38. Для фрагмента кода из Лабораторной работы 5 «Анимация качения треугольника Рёло по квадрату» прокомментируйте вычисление представленных ниже выражений и назначение каждой переменной, в них входящих

```
frame = 20

matrix = rotate_matrix(theta[frame])
relo_matrix_changed = (np.array([np.dot(matrix,row) for row in relo_matrix])
                      + center_frame[frame])
```

39. Определение класса `Complex` для описания комплексного числа: метод инициализации, метод строкового представления, методы перегрузки операций.
40. Определение класса `Stack` для описания стека: метод инициализации, метод извлечения элемента из стека, метод вставки элемента в стек.
41. Бинарное дерево поиска. Проектирование классов `BinaryTree`, `BinaryNode`, `EmptyNode` для описания бинарного дерева поиска.
42. Вставка новой вершины в непустое бинарное дерево поиска: выполните трассировку объектного уровня вычисления выражений, описывающих этот процесс. Бинарное дерево поиска описано с помощью классов `BinaryTree`, `BinaryNode`, `EmptyNode` и состоит из пяти вершин с произвольными числовыми значениями.
43. Поиск минимального значения в непустом бинарном дереве поиска: выполните трассировку объектного уровня вычисления выражений, описывающих этот процесс. Бинарное дерево поиска описано с помощью классов `BinaryTree`, `BinaryNode`, `EmptyNode` и состоит из пяти вершин с произвольными числовыми значениями.
44. Поиск максимального значения в непустом бинарном дереве поиска: выполните трассировку объектного уровня вычисления выражений, описывающих этот процесс. Бинарное дерево поиска описано с помощью классов `BinaryTree`, `BinaryNode`, `EmptyNode` и состоит из пяти вершин с произвольными числовыми значениями.
45. Подсчет количества вершин в непустом бинарном дереве поиска: выполните трассировку объектного уровня вычисления выражений, описывающих этот процесс. Бинарное дерево поиска описано с помощью классов `BinaryTree`, `BinaryNode`, `EmptyNode` и состоит из пяти вершин с произвольными числовыми значениями.

46. Проверка принадлежности некоторого значения непустому бинарному дереву поиска: выполните трассировку объектного уровня вычисления выражений, описывающих этот процесс. Бинарное дерево поиска описано с помощью классов `BinaryTree`, `BinaryNode`, `EmptyNode` и состоит из пяти вершин с произвольными числовыми значениями.
47. Центрированный обход непустого бинарного дерева поиска: выполните трассировку объектного уровня вычисления выражений, описывающих этот процесс. Бинарное дерево поиска описано с помощью классов `BinaryTree`, `BinaryNode`, `EmptyNode` и состоит из пяти вершин с произвольными числовыми значениями.

ЛИТЕРАТУРА

1. Лекционные материалы и методические материалы для выполнения лабораторных работ на сайте <http://km.mmf.bsu.by/courses/2023/km2python.html>.
2. *Лутц М. Изучаем Python, том 1* (5-е изд.) — СПб.: ООО "Диалектика", 2019.
3. *Лутц М. Изучаем Python, том 2* (5-е изд.) — СПб.: ООО "Диалектика", 2020.

Преподаватель к.ф.-м.н., доцент

_____ О.А. Лаврова

Заведующий кафедрой ДУиСА
к.ф.-м.н., доцент

_____ Л.Л. Голубева

Дата утверждения 25 апреля 2024 года, протокол № 12.