Департамент анализа данных и машинного обучения

Факультет информационных технологий и анализа больших данных

Форма обучения: **очная**, cеместр: **1, 2021-2022 учебный год**

**Список вопросов к экзамену «Алгоритмы и структуры данных на языке Python»**

**Введение**

1. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов.
2. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.

**Управление исполнением и списки**

1. Организация условий в языке Python, тернарный оператор.
2. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for.
3. Списки в Python. Обращение к элементам списка и создание срезов. Обход списка и поиск элементов в списке. Ключевые операции, проводящие к изменению списка и порождающие измененные списки, копирование списков.

**Словари, множества, кортежи**

1. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python. Операции с представлениями словарей.
2. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.
3. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств. Выполнение основных операций с парой множеств в Python.
4. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.
5. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах.
6. Функции стандартной библиотеки для работы с контейнерами.

**Функции**

1. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции. Получение информации о функции. Способы передачи параметров при вызове функции.
2. Передача переменного количества параметров (именованных и не именованных) в функции Python. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.

**Исключения и файлы**

1. Синтаксис и семантика обработки исключительных ситуаций в Python. Создание пользовательских исключений и инструкция assert.
2. Базовые операции для работы с файлами в Python. Использование инструкции with … as на примере работы с файлами.
3. Модуль Pickle, создание и работа с бинарными файлами.
4. Модуль CSV, методы создания и работы с CSV файлами.

**Модули и пакеты**

1. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.

**Задачи к экзамену по дисциплине «Алгоритмы и структуры данных на языке Python»**

1. В строке содержащей последовательность слов, разделенных запятыми удалить все нечетные слова. Ответ представить в виде строки. Пример: строка 'SIX,SEVEN,EIGHT,NINE,TEN' будет преобразована в: 'SIX,EIGHT,TEN'. (20 баллов)
2. Из списка списков элементами которого являются текстовые символы собрать строку, в которой вложенные списки объединены в слова, а слова через запятую объединены в строку. Пример список вида [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e']] будет преобразован в строку ‘Eeny,meeny,miney,moe’ (20 баллов)
3. Используя генератор словарей (и не используя код вне него) инвертировать словарь, т.е. сделать ключи словарая, его значениями и наоборото. Значения, которые в исходном словаре повторяются не добавлять в итоговый словарь. Пример: {'a':1, 'b':3, 'c':4, 'd':3} -> {1:'a', 4:'c'} (20 баллов)
4. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами являются кортежи из целых чисел в словарь в котором ключом является среднее значение из чисел исходного ключа, значение оставить прежним. Пример: {(2,4):'a', (1,1,1):'b', (2,3):'c'} -> {3.0:'a', 1.0:'b', 2.5:'c'} (20 баллов)
5. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами и значениями являются целые числа в список, в котором содержатся суммы исходных пар ключей и значений, причем, в список включаются только суммы, являющиеся четными числами. Пример: {2:4, 3:2, 12:6, 5:4, 1:3} -> [6, 18, 4] (20 баллов)
6. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами и значениями являются целые числа в список, в котором содержатся суммы исходных пар ключей и значений, причем, в список включаются только суммы, являющиеся нечетными числами. Пример: {2:4, 3:2, 12:6, 5:4, 1:3} -> [6, 18, 4] (20 баллов)
7. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами являются кортежи из целых чисел в словарь в котором ключом является максимальное значение из чисел исходного ключа, значение оставить прежним. Пример: {(2,4):'a', (1,11,1):'b', (2,3):'c'} -> {4:'a', 11:'b', 3:'c'} (20 баллов)
8. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами являются кортежи из целых чисел в словарь в котором ключем является минимальное значение из чисел исходного ключа, значение оставить прежним. Пример: {(2,4):'a', (1,2,3):'b', (12,3):'c'} -> {2:'a', 1:'b', 2:'c'} (20 баллов)
9. Используя генератор списков (и не используя код вне него) преобразовать список кортежей в список кортежей по следующему правилу: если в кортеже четное количество элементов, то из него нужно удалить последний элемент. В остальных случаях кортежи оставить неизменнымии. Приемер: [(1,3,4), (2,1), (6,), (2,2,2,1)] -> [(1,3,4), (2,), (6,), (2,2,2,)] (20 баллов)
10. Используя генератор списков (и не используя код вне него) преобразовать два списка (в первом содержатся целые числа, во втором строки, содержащие один символ) в словарь, в котором соответствующие друг другу пары значений из исходных списков преобразованы в целочисленный ключ и строку состоящую из повторенных символов (количество повтарений равно значению ключа). Пример [2, 4, 1, 3], ['a', 'b', 'c', 'd'] -> {2:'aa', 4:'bbbb', 1:'c', 3:'ddd'} (20 баллов)
11. Используя генератор списков (и не используя код вне него) преобразовать список содержащий положительные целые числа в список, элементами которого являются списки с длиной равной соответствующему числу в первом списке. Содержимым вложенных списков являются последовательно идущие целые числа начиная с 1. Пример: [3, 1, 4] -> [[1, 2, 3], [1], [1, 2, 3, 4]] (20 баллов)
12. Используя генератор списков (и не используя код вне него) преобразовать строку по следующей логике: для каждого символа исходной строки создать в итоговом списке строку, содержащую копии символа в количестве, равном номеру символа рассчитанному с конца исходной строки. Пример: 'abcd' -> ['aaaa', 'bbb', 'cc', 'd'] (20 баллов)
13. Используя генератор списков (и не используя код вне него) преобразовать строку по следующей логике: для каждого символа исходной строки создать в итоговом списке строку, содержащую копии символа в количестве, равном номеру символа в исходной строки. Пример: 'abcd' -> ['a', 'bb', 'ccc', 'dddd'] (20 баллов)
14. Используя генератор списков (и не используя код вне него) преобразовать список кортежей в список кортежей по следующему правилу: если в кортеже четное количество элементов, то из него нужно удалить последний элемент. В остальных случаях кортежи оставить неизменнымии. Приемер: [(1,3,4), (2,1), (6,), (2,2,2,1)] -> [(1,3,4), (2,), (6,), (2,2,2,)] (20 баллов)
15. На основе строки, представляющей из себя предложение, построить вложенный список, содержащий символы всех слов в предложении. Пример: строка 'Eeny, meeny, miney, moe; Catch a tiger by his toe.' будет преобразована в: [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e'], ['C', 'a', 't', 'c', 'h'], ['a'], ['t', 'i', 'g', 'e', 'r'], ['b', 'y'], ['h', 'i', 's'], ['t', 'o', 'e']] (20 баллов)
16. Реализовать однонаправленный связанный список (реализовать класс для элементов списка). Преобразовать строку 'Eeny, meeny, miney, moe; Catch a tiger by his toe.' в связный список символов строки и удалить из него все элементы содержащие гласные буквы. (20 баллов)
17. Реализовать функцию nam\_par, которая принимает на вход заранее неизвестное количество параметров и необязательный параметр name в который можно передать строку. Функция возвращает словарь в котором переданные параметры являются значениями, ключами для них являются соответсвующие (сопоставленные по порядку следования) символы из строки name. Если строка name не задана, то значения присваиваются по порядку английского алфавита. Пример 1: nam\_par(7, 3, 1, 8, 10, 13, name='xyzafg') -> {'x':7, 'y':3, 'z':1, 'a':8, 'f':10, 'g':13} Пример 2: nam\_par(21, 'val', -3.5) -> {'a':21, 'b':'val', 'c':-3.5} (20 баллов)
18. Реализовать функцию par\_val, которая принимает на вход заранее неизвестное количество именованных параметров (значения параметров - строки) и возвращает список имен параметров, которым соответствуют строки, содержащие более двух слов. Пример: par\_val(pp='abba war', fan='oneword', zr='a x') -> [pp, zr] (20 баллов)
19. Реализовать функцию psort, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список значений параметров отсортированный по именам параметров. Пример: psort(c=21, a=22, ac=17, b=16) -> [22, 17, 16, 21] (20 баллов)
20. Реализовать функцию psort, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список имен параметров, отсортированный по значениям параметров. Пример: psort(c=21, a=22, ac=17, b=16) -> [b, ac, c, a] (20 баллов)
21. Реализовать функцию repl, которая принимает на вход строку и набор заранее неизвестных параметров. Результатом функции является строка, в которой слова совпадающие с именами параметров заменены на значения параметров. Пример: строка: 'replace my val abc', параметры my='s1', abc='fff' -> Результат: 'replace s1 val fff' (20 баллов)