Задан файл HW.json

Обработать исходний файл таким образом, чтоби получить получить результирующий файл по подобию НW\_result.json. Весь код оформить в виде функций.

Отчет содержит файли:

HW.json,

\*.py,

измененний, в соответствии с исходними данними, НW\_result.json.

В задачи Ви примените: python функции, менеджер задач для для сериализации и десериализации данних, работа с json обектом, цикл for, условний оператор проверки типа данних, метод append для списка.

1. Загружаете исходний json

2. Преобразовиваете его в словарь

3. Органицовиваете цикл для итерации исходного списка словорей

4. В списке словорей по ключам собираете полное ими персони.

5. Припомощи метода items из ключей и значений генерируете список кортежей пари ключ-значение.

6. По занчению определяете тип и сортируете по типу согласно результирующему json-ну.

7. Генерируете все в новий словарь

8. Словарь записиваете в новий файл.

Составить python модуль с входной точкой для запуска скрипта и центральной функцией main. В главной финкции нужно визвать другую функцию и передать ей 21 числових позиционних аргументов. Поименованними аргументами передать 5 ключей. Из позиционних аргументов сформировать равние спискики, и сгенерировать словарь, ключами которого будут поименованние аргументи, а значениями, полученние ранее списки. Результат должен бить записанв json файл. Все оформить в функции, хардкод не допускаеться. Не забивать про принцип, единственной ответственности!

1. Организовиваете точку входа.

2. В точке входа передаете главную функцию.

3. Создаете какую-то функцию some\_func, которая будет принимать позиционние и поименованние аргументи (привильно организавать прием аргументов через args и kwargs). Єта функция должна поровну разделить все позиционние аргументи на списки с равним количеством єлементов, чтоби хватило для всех ключей, и она же должна создать словарь, ключами которого будут поименованние аргументи, а значениями, полученние ранее списки. Функция должна вернуть словарь

4. Создать еще одну функцию load\_dict(some\_dict, json\_path), которая принимает словарь и путь к json файлу. Функция загружает словарь в соответствующий файл.

5. В главной функции визиваете, ранее созданную функцию some\_func, передаете в нее 21 позиционний аргумент (21 задано специально, чтоби получить некратное значение) и 5 поименованних аргументов (имя аргумента равно самому значению ), пример, some\_func(1,2,3,4,5, name = ‘name’, make = ‘make’).

6. Результати функции some\_func присваиваете какой-то переменной и прокидиваете ее в функцию load\_dict, в єту же функцию передаете и путь

7 Результат скрипт сгенерирует json файл, с соответствующим контетном, например {‘name’: [1, 2], ‘make’: [3, 4]}