

Задания по курсу Python

Задание 3

Д.В. Иртегов

27 марта 2018 г.

Задачи необходимо сдать до 14 апреля. Решения необходимо сдавать путем отправки pull request в каталог problems-3 репозитория

<https://github.com/dmitry-irtegov/NSUPython2018>.

Датой сдачи задания считается дата отправки первого pull request. Если запрос не принят из-за моих замечаний, у вас есть неделя на их исправление.

Если запрос принят, задание считается засчитанным. Если запрос не принят, в комментарии вы можете узнать, почему.

В одном запросе следует отправлять не более одного решения. Если решение состоит из нескольких файлов, в запрос должны быть включены они все. Все запросы одного студента должны отправляться в каталог с именем, соответствующим его учетной записи. Например, для задачи 3 из группы задач 2, сдаваемой студентом v-purkin, рекомендуемое имя файла problems-3/v-purkin/task3.py.

Задача 1. Напишите скрипт, который считывает файл данных. Файл имеет формат, аналогичный <http://parallels.nsu.ru/~fat/Python/log-2>. Можете предполагать, что файл содержит только символы ASCII. Обратите внимание на размер файла! Вам необходимо:

- Выбрать все строки, начинающиеся со слова open.
- В каждой такой строке, выбрать время (число, размещенное перед словом usec). Эти времена и есть значения, которые вам необходимо обработать.
- Отбросить первое значение (в примере это 15685 usec).
- Подсчитать среднее всех остальных значений и их среднеквадратичное отклонение.
- Вывести вычисленные значения.

Задание необходимо выполнить без хранения всех значений одновременно.

Задача 2. Для того же файла данных, что и в задании 1, вместо среднеквадратичного значения подсчитайте верхний дециль, то есть такое число, что 90% всех значений не превосходят этого числа. Определить это значение без хранения хотя бы части данных, насколько я знаю, невозможно. Храните только минимально необходимые данные и используйте для их хранения оптимальную по асимптотической производительности структуру.

Задача 3. Реализуйте ленивое буферизованное чтение из файла. Реализуйте генератор или итерабельный объект, которому в качестве параметра конструктора передается открытый файл. Далее этот объект должен последовательно вернуть все байты или символы (в зависимости от того, в каком режиме был открыт файл) этого файла, включая переводы строки. При этом, объект должен читать данные из файла блоками по 512 байт или символов.

Задача 4. В задаче «Вектор» из группы заданий 2, реализуйте полиморфный конструктор. Если в качестве параметра передана коллекция, итерабельный объект или генератор с элементами, приводимыми к числовому типу, то должен создаваться вектор той же размерности, что и мощность коллекции. При этом, значения координат вектора должны быть равны значениям элементов коллекции/итерабельного объекта. Если передано N скалярных параметров, приводимых к числовому типу, должен создаваться вектор размерности N с соответствующими значениями. Во всех остальных случаях должно выкидываться исключение `TypeError`.

Во всех случаях, когда вектор создается, все элементы вектора должны быть приведены к одному типу, соответствующему самому «сложному» из переданных в качестве параметра. «Сложность» числового типа определяется списком `int`, `float`, `complex`, где `int` – самый «простой», а `complex` – самый «сложный»

Задача 5. В задаче «Вектор» из группы заданий 2, реализуйте полиморфную операцию умножения. Если этой операции передан вектор той же размерности, должно производиться скалярное умножение. Если передано число или значение, приводимое к числу, должно проводиться умножение на скаляр. Во всех остальных случаях, должно выбрасываться исключение `TypeError`.

Задача 6. Реализуйте класс `Vector3D`, который является наследником вашего класса `Vector`. Этот класс должен допускать только трехмерные вектора, и для него должна быть дополнительно определена операция векторного произведения.