

Домашнее задание к лекции 2.4 «Работа с путями»

Задача №1

Нужные для работы файлы и заготовка кода находятся на [GitHub](#).

Представьте, что вам нужно отыскать файл в формате sql среди десятков других, при этом вы знаете некоторые части этого файла (на память или из другого источника).

1. Программа ожидает строку, которую будет искать (`input()`).
2. После того, как строка введена, программа ищет её во всех файлах, выводит имена найденных файлов построчно и выводит количество найденных файлов.
3. Программа снова ожидает ввод, но теперь поиск происходит только среди отобранных на предыдущем этапе файлов.
4. Программа снова ожидает ввод.
- 5....

Выход из программы программировать не нужно. Достаточно принудительно ее останавливать, для этого можно нажать Ctrl + C. Для получения списка всех файлов используйте стандартные функции из `os` и `os.path`.

Пример на настоящих данных:

```
python3 find_procedure.py
Введите строку: INSERT
... большой список файлов ...
Всего: 301
Введите строку: APPLICATION_SETUP
... большой список файлов ...
Всего: 26
Введите строку: A400M
... большой список файлов ...
Всего: 17
Введите строку: 0.0
Migrations/000_PSE_Application_setup.sql
Migrations/100_1-32_PSE_Application_setup.sql
Всего: 2
Введите строку: 2.0
Migrations/000_PSE_Application_setup.sql
Всего: 1
```

Не забываем организовывать собственный код в функции.

Задача №2. Дополнительная (не обязательная)

Генерировать абсолютный путь до папки с миграциями. Скрипт при этом лежит в одной папке с папкой 'Migrations', но запускать мы его можем из любой директории - он будет работать корректно.

`find_procedure.py` - Поиск только по названиям файлов.

`find_procedure_2.py` - Поиск только по названиям файлов. !!! Внимание: `find_procedure_2.py` ищется по всему диску, ссылка на который в переменной окружения ОС 'HOMEDRIVE'. `find_procedure_2.py` должен быть один на диске.

`find_procedure_3.py` - Поиск по содержимому файлов. Работает только с текстовыми файлами.

`find_procedure_4.py` - Поиск по названию и содержимому файлов. Работает только с текстовыми файлами.

Для `find_procedure_3.py` сделал три версии - экспериментировал с влиянием объема кода (количеством циклов в программе) на быстродействие. Для всех четырех версий `find_procedure_3.py` быстродействие оказалось примерно одинаковым. В `find_procedure_3_v3.py` подготовка к выполнению занимает столько же времени, как и первый запрос в других версиях, зато все запросы выполняются моментально.