

Домашнее задание по теме «Работа с итераторами и генераторами»

Формулировка задания

Решить несколько задач на python с использованием свойств итераторов и генераторов. Каждая задача представляет собой написание функции или класса. Результирующий код должен быть читаемым, с единой системой отступов и адекватными названиями переменных.

Описание плана работы

Решить несколько задач на python с использованием свойств итераторов и генераторов.

Желательна реализация в файлах py. Сохранить задачи (сделать коммиты для каждой) в локальном git и опубликовать в удаленном репозитории.

Для отчета по работе выполнить задание в файле.py или .ipynb. Сделать снимки экрана корректного выполнения программы в IDE.

Задача 1

Создать программу, которая работает со списком. Пользователь вводит список в программу. Реализовать следующие действия:

1. С помощью функции any проверить содержит ли этот список хотя бы одно положительное число.
2. С помощью функции all проверить состоят ли все элементы списка только из чисел.
3. Отсортировать список с помощью функции sorted.

Задача 2

Написать класс CyclicIterator. Итератор должен итерировать по итерируемому объекту (например, по range, list, tuple, set и т. д.), и когда достигнет последнего элемента, начинать сначала. Проверить работу итератора.

Задача 3

Задайте функцию-генератор, которая на основе символов строки chars:

```
from string import ascii_lowercase, ascii_uppercase
```

```
chars = ascii_lowercase + ascii_uppercase + "0123456789!@#*$"
```

формирует и выдает случайно сгенерированные пароли длиной 12 символов. Количество выдаваемых паролей функцией должно быть неограниченным.

Случайный выбор символа из последовательности chars можно реализовать с помощью функции choice модуля random. Выведите первые пять сгенерированных паролей на экран.

Задача 4

Создать класс Movie. У Movie есть название и расписание, по каким дням он идёт в кинотеатрах. Название Movie - это текст. Расписание хранится периодами дат. Например, прокат фильма проходит с 1 по 7 января, а потом показ возобновляется с 15 января по 7 февраля:

```
[
    (datetime(2024, 11, 1), datetime(2024, 11, 7)),
    (datetime(2024, 12, 15), datetime(2024, 12, 31))
]
```

Реализуйте у класса Movie метод schedule. Он будет генерировать дни, в которые показывают фильм. Проверить работу генератора.

Ожидаемый вывод программы:

```
2024-11-01 00:00:00
2024-11-02 00:00:00
2024-11-03 00:00:00
2024-11-04 00:00:00
2024-11-05 00:00:00
2024-11-06 00:00:00
2024-11-07 00:00:00
2024-12-15 00:00:00
2024-12-16 00:00:00
2024-12-17 00:00:00
2024-12-18 00:00:00
2024-12-19 00:00:00
2024-12-20 00:00:00
2024-12-21 00:00:00
2024-12-22 00:00:00
2024-12-23 00:00:00
2024-12-24 00:00:00
2024-12-25 00:00:00
2024-12-26 00:00:00
2024-12-27 00:00:00
2024-12-28 00:00:00
2024-12-29 00:00:00
2024-12-30 00:00:00
2024-12-31 00:00:00
```

Перечень необходимых инструментов

- Python
- venv
- Jupiter Notebook
- IDE VS Code
- GigaIDE

Форма предоставления результата

1. В поле ссылки загрузить ссылку на удаленный репозиторий с доступом для наставника.
2. В поле файла загрузить архив с папкой, в которой разместить отчет со скриншотами по заданиям и решение задач 1-4. Решения должны быть представлены в формате .ipynb или .py.

Шкала оценивания

- 1.0 – отлично
- 0.7–0.9 – хорошо
- 0.5–0.6 – удовлетворительно
- Менее 0.5 – задание не выполнено