Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию «Вычисление последовательностей OEIS с использованием механизма итераторов или генераторов»

Выполнил:
Студент группы ИУ5-32Б:
Арзамасцев Артем
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2022 г.

Описание задания

1. С использованием механизма итераторов или генераторов реализуйте с помощью концепции ленивых вычислений одну из последовательностей OEIS. Примером могут являться числа Фибоначчи.
2. Для реализованной последовательности разработайте 3-5 модульных тестов, которые, в том числе, проверяют то, что последовательность поддерживает ленивые вычисления.
3. Разработайте веб-сервис с использованием фреймворка Flask, который возвращает N элементов последовательности (параметр N передается в запросе к сервису).
4. Создайте Jupyter-notebook, который реализует обращение к веб-сервису с использованием библиотеки requests и визуализацию полученных от веб-сервиса данных с использованием библиотеки matplotlib.

Текст программы

Задание 1

```
def catalan_numbers_gen():
    num, prev = 1, 1
    while True:
        yield prev
        prev = 2 * (2*num - 1) * prev // (num + 1)
        num += 1
```

Задание 2

```
import unittest
from catalan_numbers import catalan_numbers_gen
from collections.abc import Generator
from functools import reduce


class TestCatalan(unittest.TestCase):
```

```
def test_generator(self):
    sequence = catalan_numbers_gen()
    self.assertIsInstance(sequence, Generator)

    self.assertEqual(next(sequence), 1)
    self.assertEqual(next(sequence), 1)
    self.assertEqual(next(sequence), 2)

def test_sequence(self):
    gen = catalan_numbers_gen()
    sequence = [next(gen) for _ in range(10)]
    self.assertEqual(len(sequence), 10)
    self.assertListEqual(sequence, [1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862])

    exp = [16796, 58786]
    for ind, val in enumerate(gen):
        if ind > 1:
            break
        self.assertEqual(val, exp[ind])

def test_func(self):
    gen = catalan_numbers_gen()
    sequence = list(zip(range(5), gen))
    self.assertEqual(len(sequence), 5)
```

```
        self.assertListEqual(sequence, [(0, 1), (1, 1), (2, 2), (3, 5), (4, 14)])


        sequence = list(zip(range(5), gen))
        self.assertEqual(len(sequence), 5)
        self.assertListEqual(sequence, [(0, 42), (1, 132), (2, 429), (3, 1430), (4, 4862)])



if name == "main":
    unittest.main()
```

Задание 3

```
from flask import Flask
from catalan_numbers import catalan_numbers_gen



app = Flask(name)



@app.route("/")
def main_page():
    return "<h3>Catalan number sequence generator</h3>"



@app.route("/catalan/<int:num>")
```

```
def get_catalan_numbers(num):
    catalan = catalan_numbers_gen()
    return [next(catalan) for _ in range(num)]


if name == "main":
    app.run(host="localhost", port=8080)
```

Примеры выполнения программ