**Министерство науки и высшего образования Российской Федерации** **Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Домашнее заданиепо дисциплине «Разработка комплексного приложения на языке Python»

по предмету Базовые компоненты интернет технологий

Выполнила:

студентка группы ИУ5-33Б

Валова С. В.

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Ю. Е.

2022 г.

Условие задания:

1. С использованием механизма итераторов или генераторов реализуйте с помощью концепции ленивых вычислений [одну из последовательностей OEIS.](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D1%8D%D0%BD%D1%86%D0%B8%D0%BA%D0%BB%D0%BE%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F_%D1%86%D0%B5%D0%BB%D0%BE%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D1%85_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9) Примером могут являться [числа Фибоначчи.](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8)
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](https://requests.readthedocs.io/en/latest/) и визуализацию полученных от веб-сервиса данных с использованием библиотеки [matplotlib.](https://matplotlib.org/)

Текст программы

Файл app.py

from flask import Flask  
from fibonacci import func\_fib  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def hello():  
 return 'There are the Fibonacci numbers:'  
  
@app.route('/<int:NumberOfNumbers>')  
def num(NumberOfNumbers):  
 b = func\_fib()  
 fib\_numbers = [next(b) for i in range(NumberOfNumbers)]  
 return fib\_numbers  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)

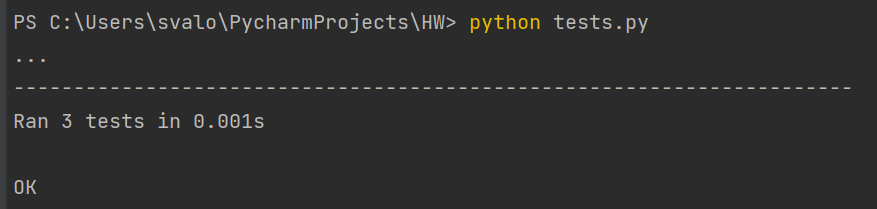
Файл fibonacci.py

def func\_fib(n=25):  
 fib1, fib2 = 0, 1  
 for i in range(n):  
 yield fib1  
 fib1, fib2 = fib2, fib1 + fib2

Файл test.py

import unittest  
from fibonacci import func\_fib  
  
class test\_fib(unittest.TestCase):  
 def test\_1(self):  
 b = func\_fib()  
 fib\_numbers = [next(b) for i in range(5)]  
 self.assertEqual(fib\_numbers, [0, 1, 1, 2, 3])  
  
 def test\_2(self):  
 b = func\_fib()  
 fib\_numbers = [next(b) for i in range(2)]  
 self.assertEqual(fib\_numbers, [0, 1])  
  
 def test\_3(self):  
 b = func\_fib()  
 fib\_numbers = [next(b) for i in range(10)]  
 self.assertEqual(fib\_numbers, [0, 1, 1, 2, 3, 5, 8, 13, 21, 34])  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

Результат выполнения тестов:



Результаты выполнения:

