**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчёт по домашнему заданию.

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31б  Садиров Ф.Э. |  | преподаватель каф. ИУ5  Гапанюк Ю.Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

**Москва, 2022 г.**

**Задание**

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**

***fibo.py***

def fibonachi(n):  
 f = [0, 1]  
 for i in range(n):  
 yield f[i]  
 f.append(f[i] + f[i + 1])

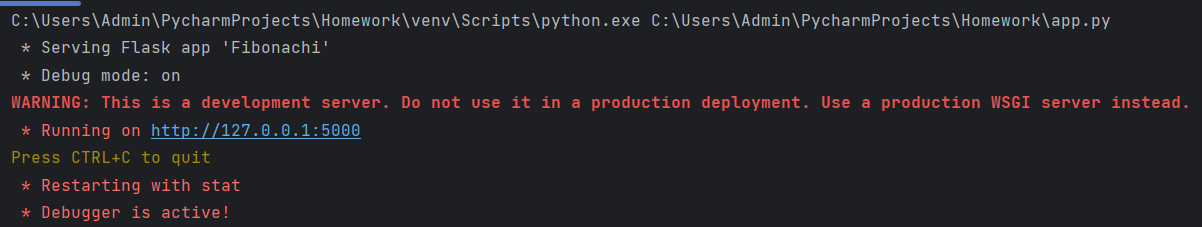
***testfibo.py***

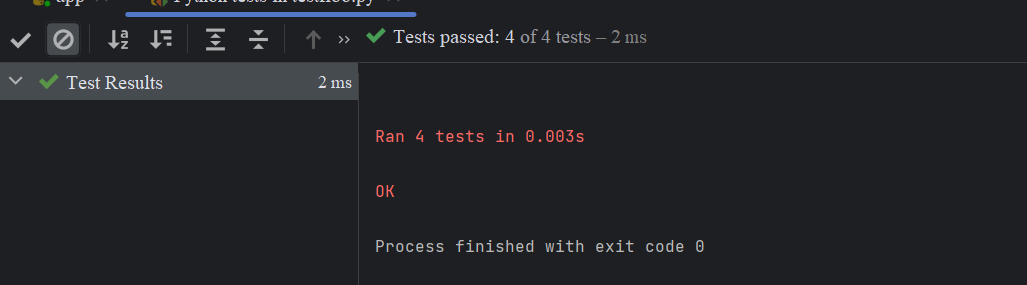
import unittest  
from fibo import \*  
import time  
  
value\_for\_timer = 1000000  
  
  
class TestFibonachi(unittest.TestCase):  
 def test\_zero\_if\_number\_is\_zero(self):  
 self.assertEqual(list(fibonachi(0)), [])  
  
 def test\_one\_if\_number\_is\_more\_then\_zero(self):  
 self.assertEqual(list(fibonachi(1)), [0])  
  
 def test\_two\_if\_number\_is\_high(self):  
 self.assertEqual(list(fibonachi(5)), [0, 1, 1, 2, 3])  
  
 def test\_time(self):  
 start = time.time()  
 a = fibonachi(value\_for\_timer)  
 end = time.time()  
 self.assertLess(end - start, 1)

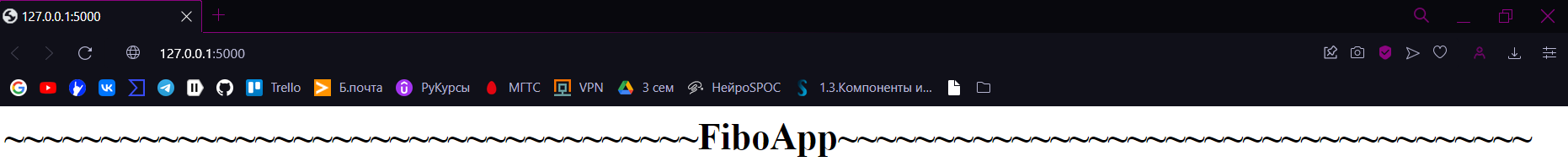
***app.py***

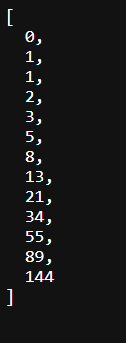
from flask import Flask  
from fibo import \*  
  
app = Flask('Fibonachi')  
  
  
# <h1> used to increase text size  
@app.route('/')  
def welcome():  
 return "<h1>~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~FiboApp~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~<h1>"  
  
  
@app.route('/<int:n>')  
def index(n):  
 a = list(fibonachi(n))  
 return a  
  
  
@app.errorhandler(404)  
def page\_not\_found(e):  
 return "Please enter integers!"  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

**Анализ результатов**









***Notes.ipynb***

