Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «БКИТ»
Отчет по домашнему заданию

Выполнил:
студент группы ИУ5-35Б
Большаков Георгий
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Нардид А.Н.
Подпись и дата:

Москва, 2022 г.

Описание задания

Задание:

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

fib.py

```
def fib():
    prev, cur = 0, 1
    while True:
        yield cur
        prev, cur = cur, prev+cur

def get_fib_number_at_pos(pos):
    fib_gen = fib()
    number = 0
    for i in range(pos):
        number = next(fib_gen)
    return number

def get_fib_seq(n):
    fib_gen = fib()
    arr = []
    for i in range(n):
        arr.append(next(fib_gen))
    return arr
```

test.py

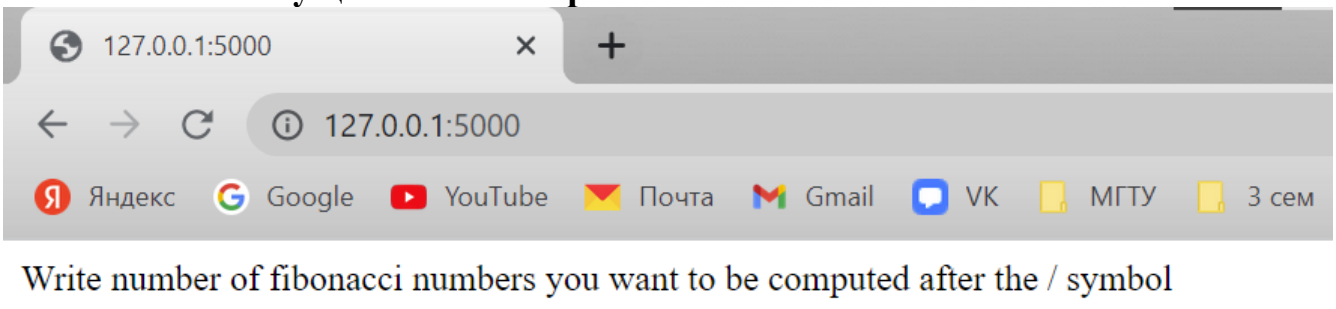
```
import unittest
from fib import get_fib_number_at_pos, get_fib_seq, fib


class TEST(unittest.TestCase):
    def test_Pos9(self):
        self.assertEqual(34, get_fib_number_at_pos(9))

    def test_SeqOf5(self):
        self.assertEqual([1, 1, 2, 3, 5], get_fib_seq(5))

    def test_LenSeqOf10(self):
        self.assertEqual(10, len(get_fib_seq(10)))

if __name__ == "__main__":
    unittest.main()
```

main.py

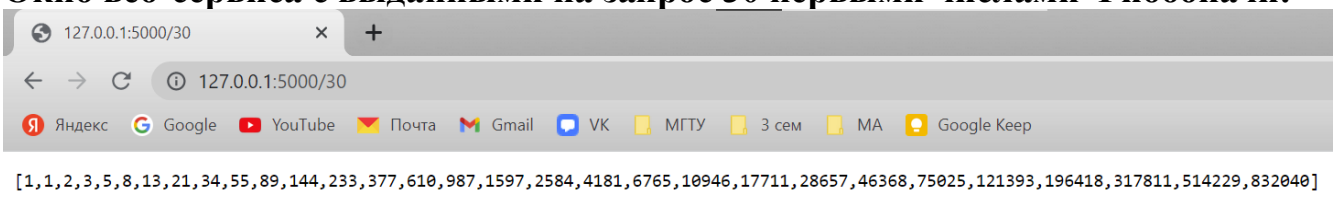
```
from fib import fib

from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Write number of fibonacci numbers you want to be computed
after the / symbol</p>"

@app.route("/<int:n>")
def fibonacci_number(n):
    fib_gen = fib()
    fib_numbers = []
    for i in range(n):
        fib_numbers.append(next(fib_gen))
    return fib_numbers
```

Экранные формы

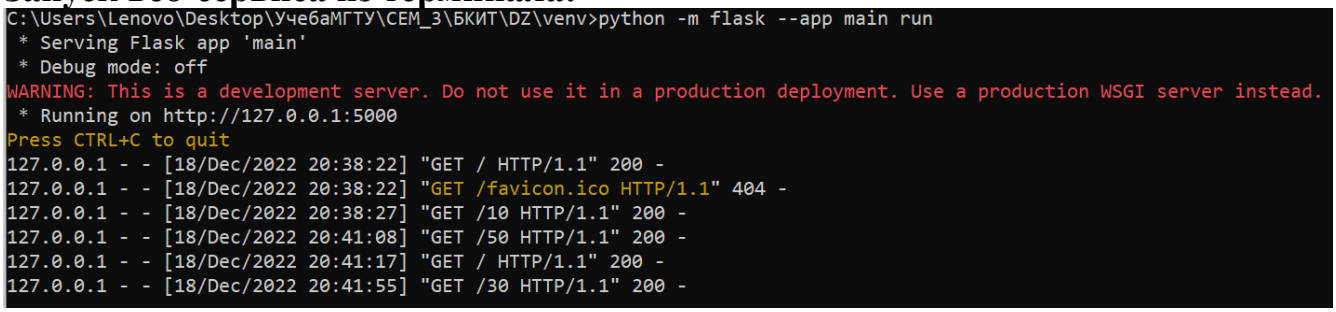
Главное окно запущенного веб-сервиса:



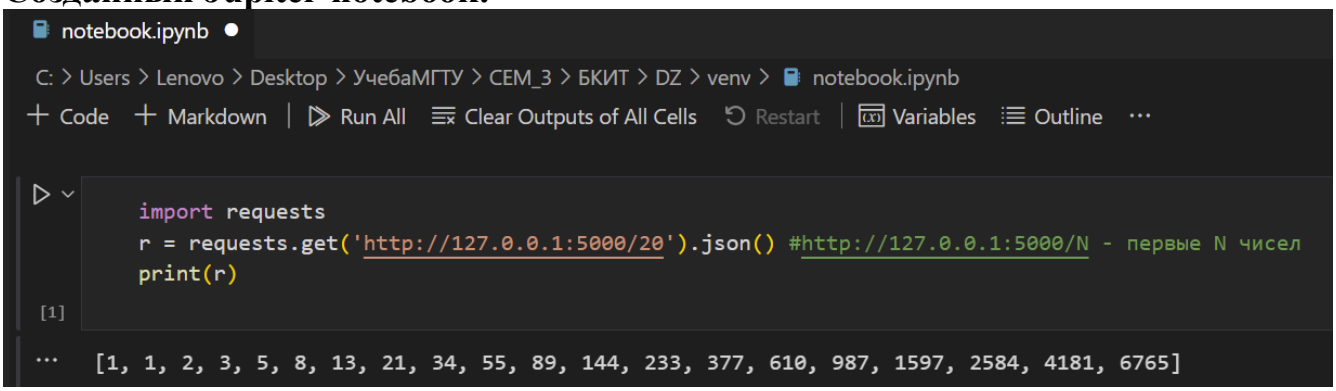
Окно веб-сервиса с выданными на запрос 30 первыми числами Фиббоначи:



Запуск веб-сервиса из терминала:

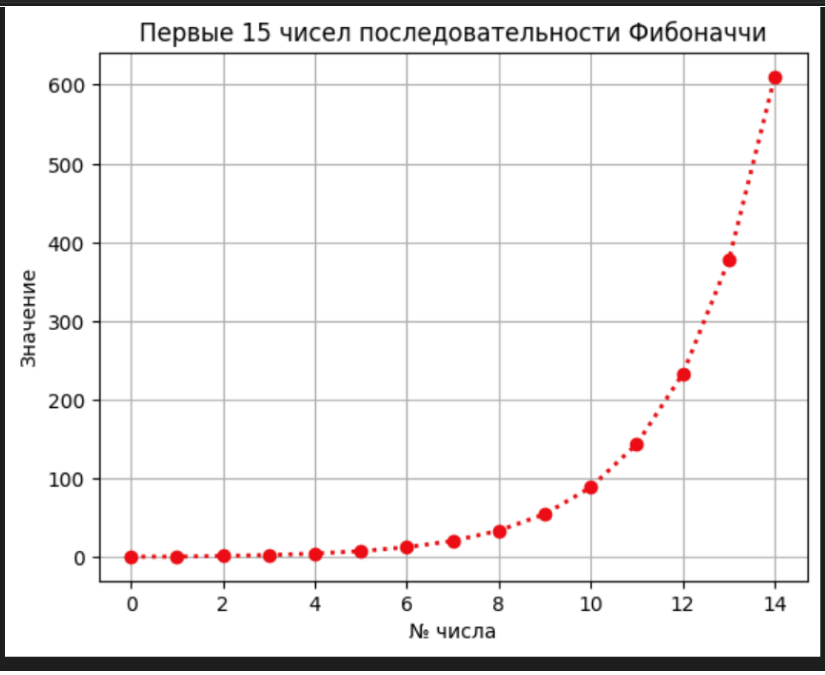


Созданный Jupiter-notebook:



```python
import matplotlib.pyplot as plt
import requests

fib_numbers_amount = 15

fib_numbers = requests.get('http://127.0.0.1:5000/{}'.format(fib_numbers_amount)).json()

x = [x for x in range(fib_numbers_amount)]
y = fib_numbers

fig, ax = plt.subplots()
plt.xlabel('№ числа')
plt.ylabel('Значение')
plt.title('Первые {} чисел последовательности Фибоначчи'.format(fib_numbers_amount))

ax.plot(x, y, color='red', marker='o', linestyle=':',
     linewidth=2, markersize=6)
ax.grid(True)

plt.show()
```

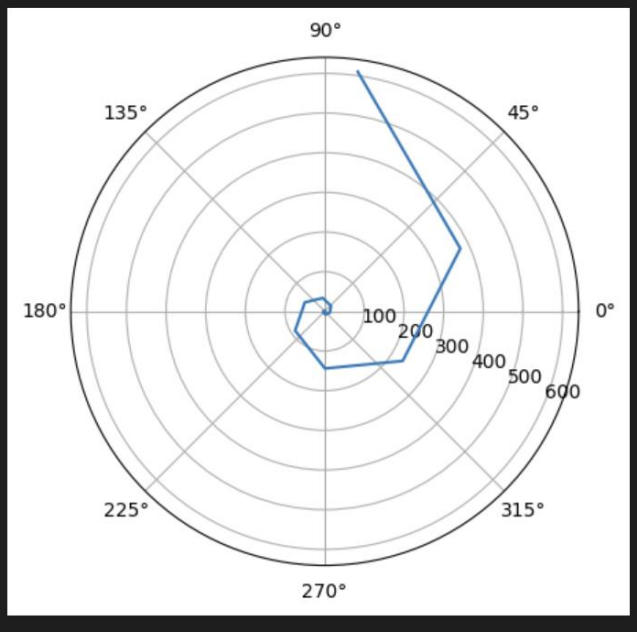
[2] Python



```python
import matplotlib.pyplot as plt
import requests


fib_numbers_amount = 15

fib_numbers = requests.get('http://127.0.0.1:5000/{}'.format(fib_numbers_amount)).json()

xs = [xs for xs in range(fib_numbers_amount)]
ys = fib_numbers

plt.figure(figsize=(8,8))

fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
ax.set_rlabel_position(-22.5)  # Move radial labels away from plotted line
ax.plot(xs, ys)
ax.grid(True)

plt.show()
```

[3] Python

```
<Figure size 800x800 with 0 Axes>
```



TDD-тестирование используемого генератора:

```
C:\Users\Lenovo\Desktop\УчебаМГТУ\СЕМ_3\БКИТ\DZ\venv>python -m unittest -v test.py
test_LenSeqOf10 (test.TEST) ... ok
test_Pos9 (test.TEST) ... ok
test_SeqOf5 (test.TEST) ... ok

----------------------------------------------------------------------
Ran 3 tests in 0.002s

OK
```