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
Трифонов Дмитрий
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
```

test.py

```
import unittest
from fib import get_fib_number_at_pos

class FibTestCase(unittest.TestCase):
    def test_number_1(self):
        self.assertEqual(1, get_fib_number_at_pos(1))
    def test_number_3(self):
        self.assertEqual(2, get_fib_number_at_pos(3))
    def test_number_7(self):
        self.assertEqual(13, get_fib_number_at_pos(7))
    def test_number_8(self):
        self.assertEqual(21, get_fib_number_at_pos(8))
    def test_number_9(self):
        self.assertEqual(34, get_fib_number_at_pos(9))
```

main.py

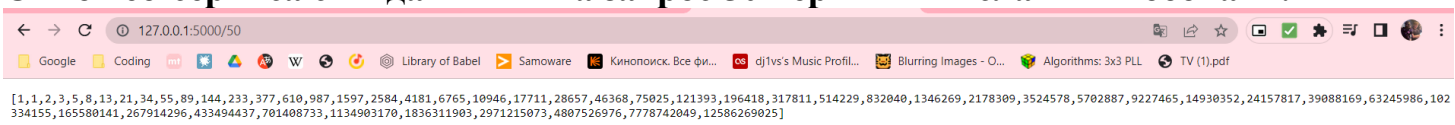
```
from fib import fib

from flask import Flask

app = Flask(__name__)

@app.route("/")
```

```
def hello_world():
    return "<p>Write number of fibonachi numbers you want to be computed after the /
symbol</p>"

@app.route("/<int:n>")
def fibonachi_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:

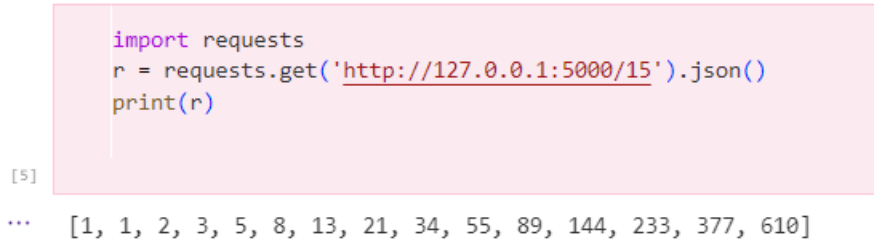
Окно веб-сервиса с выданными на запрос 50 первыми числами Фиббоначи:



Запуск веб-сервиса из терминала:

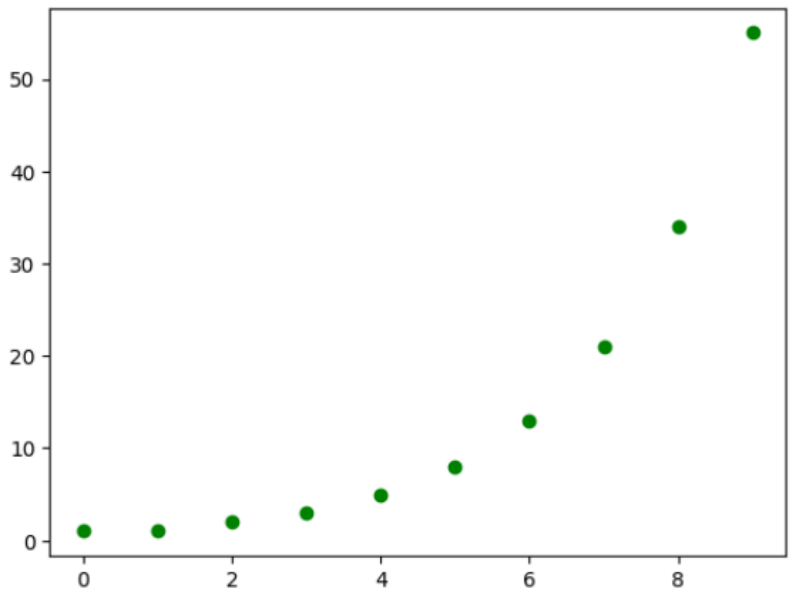
```
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\study\BKIT\dz> python -m flask --app main run
 * Serving Flask app 'main'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [06/Dec/2022 00:59:55] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Dec/2022 00:59:58] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [06/Dec/2022 01:00:32] "GET /50 HTTP/1.1" 200 -
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\study\BKIT\dz>
```

Созданный Jupiter-notebook:



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

fib_numbers_amount = 10

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()

ax.plot(x, y, 'go')

plt.show()
```

[1] ✓ 1.6s



TDD-тестирование используемого генератора:

```
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\study\BKIT\dz> python -m unittest -v test.py
test_number_1 (test.FibTestCase) ... ok
test_number_3 (test.FibTestCase) ... ok
test_number_7 (test.FibTestCase) ... ok
test_number_8 (test.FibTestCase) ... ok
test_number_9 (test.FibTestCase) ... ok

----------------------------------------------------------------------
Ran 5 tests in 0.001s

OK
(bkit-knEZUJXe) PS C:\Users\Dmitriy\Documents\study\BKIT\dz>
```