

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

Курс «Базовые компоненты интернет-технологий»
Домашнее задание

Выполнил:
студент группы ИУ5-34Б:
Даниелян А.А.
Подпись и дата:

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

Москва, 2022 г.

Постановка задачи:

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:

Файл app.py

```
from flask import Flask
from markupsafe import escape
from function.fibonacci import gen_fibonacci, print_fibonacci
app = Flask(__name__)
@app.route("/fibonacci/<int:num>")
def main(num):
    return print_fibonacci(num)
    # stream_with_context(gen_fibonacci(num))
if __name__ == "__main__":
    app.run()
```

Файл Fibonacci.py

```
def gen_fibonacci(n):
    k=0
    n1 = 0
    n2 = 1
    if n ==0:
        yield "0"
        return
    if n == 1:
        yield "1"
        return
    yield 1
    for i in range(n):
        res = n1+n2
        n1, n2 = n2, res
        yield str(res)
    return "end"

def print_fibonacci(n):
    ...
    works only with n>1
```

```

...
f = gen_fibonacci(n)
s=str()
for i in range(n):
    #print (f.__next__())
    s+=str(f.__next__())+', '
s=str(s).rstrip(s[-1])
return s

```

Файл test.py

```

import unittest
from function.fibonacci import print_fibonacci, gen_fibonacci
#Чтобы запустить тест, напиши в терминале "python -m unittest test.py"

```

```

class Test_DZ(unittest.TestCase):
    def test_one (self):
        self.assertEqual (gen_fibonacci(1).__next__(), "1")
    def test_two(self):
        self.assertEqual (print_fibonacci(2), "1,1")
    def test_six (self):
        self.assertEqual(print_fibonacci(6), "1,1,2,3,5,8")

```

Результаты выполнения:

```

In [1]: import requests
import matplotlib.pyplot as plt

```

```

In [2]: r = requests.get('http://127.0.0.1:5000/fibonacci/13')
y=list((r.text).split(sep = ','))
y = [int(x) for x in y]

```

```

In [3]: x = [i for i in range(1,14)]

```

```

In [4]: fig = plt.figure(figsize = (6,4))
ax = fig.add_subplot()
ax.bar(x,y)
ax.grid()
plt.show()

```

