

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

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

Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию

Выполнила:

студентка группы ИУ5-34Б:

Теряева Ксения Владимировна
15.12.2022г

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Москва, 2022 г

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:

```
def Fibonnachi(n):
    if n>0:
        numb1 = 0
        numb2 = 1
        i = 0
        summ = 0
        yield 0
        while i<n-1:
            summ = numb1+numb2
            numb1 = numb2
            numb2 = summ
            i+=1
            yield numb1
print(list(Fibonnachi(5)))
```

```
import unittest
import sys
from DZ import Fibonnachi

class Test_task(unittest.TestCase):
    def test_numbers(self):
        res_subsequence = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
        self.assertEqual(list(Fibonnachi(10)), res_subsequence)
    def test_empty(self):
        self.assertEqual(list(Fibonnachi(0)), [])
    def test_negative(self):
        self.assertEqual(list(Fibonnachi(-2)), [])
    def lazy_test(self):
        self.assertEqual(sys.getsizeof(Fibonnachi(10)), sys.getsizeof(Fibonnachi(10_000)))
if __name__ == '__main__':
    unittest.main()
```

```
from flask import Flask, render_template, request
from DZ import Fibonnachi

app = Flask(__name__)

@app.route('/form', methods=['post', 'get'])
def form():
    message = ''
    if request.method == 'POST':
        numbers = request.form.get('numbers') #запрос к данным форм
        message = " ".join([str(i) for i in Fibonnachi(int(numbers))])
        #list(Fibonnachi(int(numbers)))
    return render_template('form.html', message=message)

@app.route("/api", methods= ["get"])
def api():
    numbers = int(request.args.get('numbers'))
    return " ".join([str(i) for i in Fibonnachi(int(numbers))])
```

Результат работы программы:

[0, 1, 1, 2, 3]

...

Ran 3 tests in 0.004s

OK

0 1 1 2 3 5 8 13 21 34

Введите число:

Отправить

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

```
In [2]: url = "http://127.0.0.1:5000/api"
num = r.get(url, params={'numbers': input("Введите n: ")}).text
Введите n: 30
```

```
In [3]: list(map(int, num.split()))
```

```
In [5]: plt.plot(list(map(int, num.split())), 'ro')
```

```
Out[5]: [matplotlib.lines.Line2D at 0x22a1b3eeaa0]
```

