

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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «БКИТ»

Отчет по ДЗ

Выполнил:

студент группы ИУ5-35Б

Крылов Дмитрий

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

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.

Код программы

main.py

```
def fibonacci(n):  
    if n >= 10000000:  
        yield "Параметр слишком большой! (> 10000000)"  
    else:  
        a, b = 0, 1  
        for i in range(n):  
            yield a  
            a, b = b, a + b  
  
if __name__ == '__main__':  
    print(*fibonacci(10))
```

test_main.py

```
import unittest
from main import fibonacci
import unittest
import main

class TestEquation(unittest.TestCase):

    def test_get_roots(self):
        self.assertEqual(list(fibonacci(5)), [0, 1, 1, 2, 3])

    def test_value(self):
        self.assertEqual(list(fibonacci(7)), [0, 1, 1, 2, 3, 5, 8])

    def test_name(self):
        self.assertEqual(list(fibonacci(10000000)), ['Параметр слишком большой! (> 1000000)'])

if __name__ == '__main__':
    assert (list(main.fibonacci(3)) == [0, 1, 1])
    assert (list(fibonacci(4)) == [0, 1, 1, 2])
    assert (list(fibonacci(5)) == [0, 1, 1, 2, 3])
    unittest.main()
```

app.py

```
from flask import Flask
import main

app = Flask('fibonacci')

@app.route('/')
def start():
    return "use /n where n - int"

@app.route('/<int:n>')
def fib(n):
    return str(list(main.fibonacci(n)))

@app.errorhandler(404)
def page_not_found(e):
    return "use /n where n - int"

if __name__=="__main__":
    app.run(debug=True)
```

Результаты работы

```
/Users/dmitrykrylov/PycharmProjects/lab6/venv/bin/python /Users/dmitrykrylov/PycharmProjects/lab6/dz/app.py
* Serving Flask app 'fibonacci'
* Debug mode: on
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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 228-388-987
```



[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

```
In [16]: import requests
import matplotlib.pyplot as plt
```

```
In [17]: r = requests.get('http://127.0.0.1:5000/40')
print(r.text)

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765,
10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309,
3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986]
```

```
In [18]: plt.plot(r.json())
plt.ylabel('some numbers')
plt.show()
```

