



**Министерство науки и высшего образования
Российской Федерации Федеральное государственное
бюджетное образовательное учреждение высшего
образования**

**«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика, искусственный интеллект и
системы управления»**

**Кафедра «Системы обработки информации и
управления»**

**Домашнее задание по курсу
«Базовые компоненты интернет-технологий»**

Выполнил:
студент группы ИУ5-33Б
Иванченко Максим

Проверил:
Доцент кафедры ИУ5
Гапанюк Юрий Евгеньевич

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 fibonacci_func.py (реализация генератора)

```
# Последовательность Фибоначчи
# 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144
def fibonacci(n):
    if n <= 0:
        pass
    elif n == 1:
        yield 0
    else:
        yield 0
        yield 1
        prev_2, prev = 0, 1
        for i in range(n-2):
            yield prev_2 + prev
            prev_2, prev = prev, prev_2+prev
```

Тестирование unittest

```
import types
from fibonacci_func import fibonacci
import unittest

class FibonacciTestCase(unittest.TestCase):
    def test_n_is_not_positive(self):
        n = 0
        self.assertEqual([elem for elem in fibonacci(n)], [])
        n = -5
        self.assertEqual([elem for elem in fibonacci(n)], [])

    def test_n_is_one(self):
        n = 1
        self.assertEqual([elem for elem in fibonacci(n)], [e for e in [0]])

    def test_n_is_two(self):
        n = 2
        self.assertEqual([elem for elem in fibonacci(n)], [e for e in [0, 1]])

    def test_n_is_some(self):
        n = 13
        self.assertEqual([elem for elem in fibonacci(n)], [e for e in [0, 1, 1,
2, 3, 5, 8, 13, 21, 34, 55, 89, 144]])

    def test_lazy_calculations(self):
        for n in -5, 0, 1, 2, 13:
            self.assertEqual(type(fibonacci(n)), types.GeneratorType)

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

Файл flask_app.py (реализация Flask-сервиса)

```
from fibonacci_func import *
from flask import Flask, request, render_template, redirect
app = Flask(__name__)

@app.route('/first_version')
def fib_1():
```

```

    # http://127.0.0.1:5000/first_version
    cnt = 15
    return f'<h1>{cnt} первых чисел Фибоначчи: {str(list(fibonachi(cnt)))[1:-1]}'<h1>'

@app.route('/second_version')
def fib_2():
    # http://127.0.0.1:5000/second_version?count=12
    cnt = int(request.args.get('count'))
    return f'<h1>{cnt} первых чисел Фибоначчи: {str(list(fibonachi(cnt)))[1:-1]}'<h1>'

# Третий вариант с вводом на сервисе
result = ''

@app.route('/third_version', methods=['GET'])
def fib_3():
    # http://127.0.0.1:5000/third_version
    return render_template('get_number.html', sequence=result)

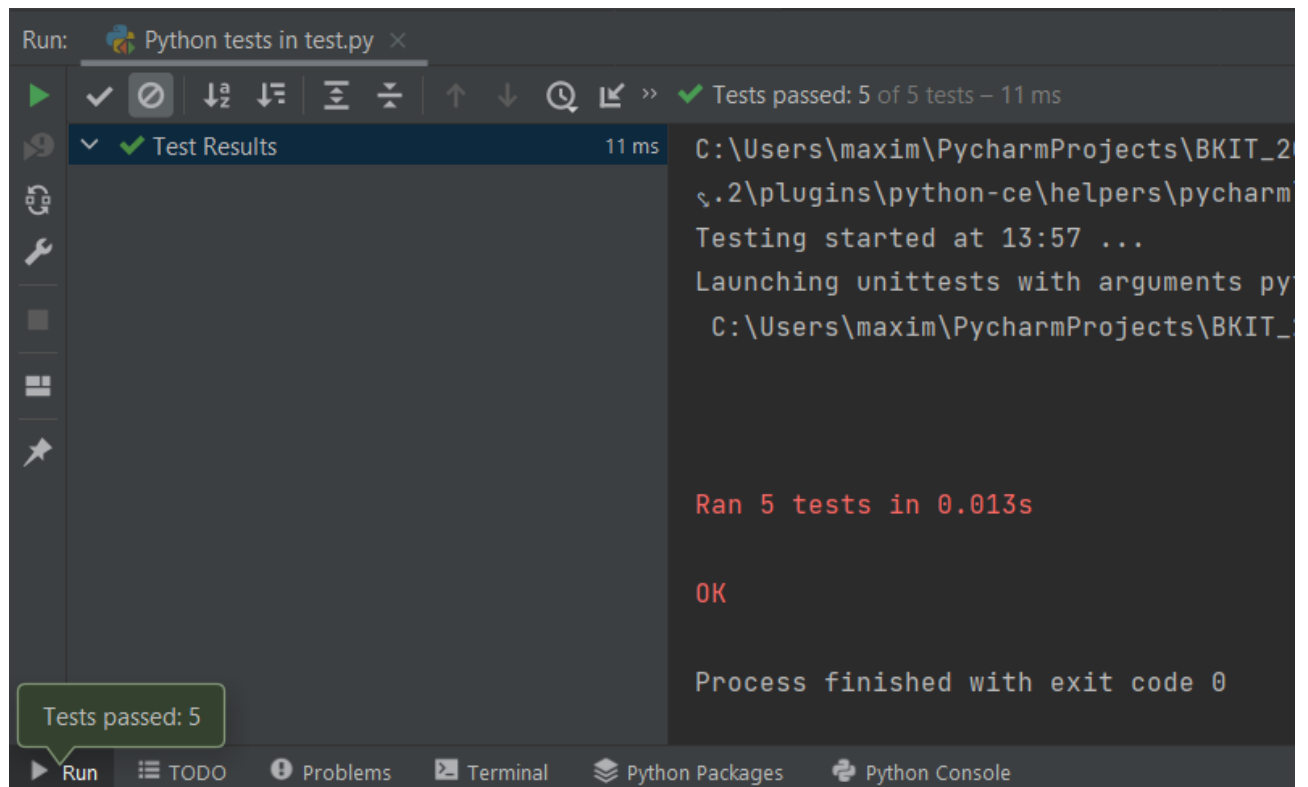
@app.route('/show_sequence', methods=['POST'])
def fib_3_help():
    cnt = int(request.form['Number'])
    global result
    result = str(cnt) + ' первых чисел Фибоначчи: ' +
str(list(fibonachi(cnt)))[1:-1]
    return redirect('/third_version')

if __name__ == '__main__':
    app.run()

```

Результаты выполнения

Тестирование



Flask-сервис



12 первых чисел Фибоначчи: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89

Jupyter-notebook (см. приложение)