



**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 и системы управления»

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

Домашнее задание

«Функциональные возможности языка Python»

по предмету

«Разработка комплексного приложения»

Выполнил:

студент группы № ИУ5-33Б

Пермяков Дмитрий

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2022 г.

Задание.

С использованием механизма итераторов или генераторов реализуйте с помощью концепции ленивых вычислений одну из последовательностей OEIS. Примером могут являться числа Фибоначчи.

Для реализованной последовательности разработайте 3-5 модульных тестов, которые, в том числе, проверяют то, что последовательность поддерживает ленивые вычисления.

Разработайте веб-сервис с использованием фреймворка Flask, который возвращает N элементов последовательности (параметр N передается в запросе к сервису).

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:

```
# Copyright © 2022 mightyKingRichard <dimapermyakov55@gmail.com>
import requests

def fibonacci(end_cycle):
    """
    Числа Фибоначчи.
    :param end_cycle: количество цифр.
    :return: генератор
    """
    if type(end_cycle) != int:
        raise TypeError('Incorrect type of the end cycle! Waiting: Int')
    old = 0
    res = 1
    for _ in range(end_cycle):
        yield old
        old, res = res, old + res

if __name__ == '__main__':
    url = 'https://127.0.0.1:5000/num/10'
    r = requests.get(url)
```

Libs.fibonacci_test.py:

```
# Copyright © 2022 mightyKingRichard <dimapermyakov55@gmail.com>
import unittest
from src.functions import fibonacci

# python3 -m unittest -v libs/*
class TestFibonacci(unittest.TestCase):
    def test_numbers(self):
        self.assertEqual(len(list(fibonacci(10))), 10)
        self.assertEqual(list(fibonacci(10)), [0, 1, 1, 2, 3, 5, 8, 13, 21,
34])

    def test_iteration(self):
        res = fibonacci(2)
        self.assertEqual(next(res), 0)
        self.assertEqual(next(res), 1)

    def test_type_numbers(self):
        self.assertRaises(TypeError, fibonacci('5'))
```

src.get_numbers:

```
# Copyright © 2022 mightyKlngRichard <dimapermyakov55@gmail.com>

import requests
import matplotlib.pyplot as plt

def make_url(main_url: str, count_numbers: int) -> str:
    base_url = main_url
    res_url = f"{base_url}/num/{count_numbers}"
    return res_url

def get_data(*, main_url: str, count_numbers: int) -> str:
    res_url = make_url(main_url, count_numbers)
    r = requests.get(res_url)
    return r.text

if __name__ == '__main__':
    url = 'http://127.0.0.1:5000'
    y_axis = get_data(main_url=url, count_numbers=int(input('Сколько чисел Фибанначу Вам необходимо? '))).split(', ')
    print(y_axis)
    print(len(y_axis))
    x_axis = list(range(1, len(y_axis) + 1))
    fig = plt.figure(figsize=(10, 5))
    plt.bar(x_axis, y_axis)
    plt.xlabel('Ось абсцисс')
    plt.ylabel('Ось ординат')
    plt.title('Первые {} чисел последовательности Фибоначчи'.format(len(y_axis)))
    plt.show()

    plt.plot(x_axis, y_axis)
    plt.show()
```

app.py:

```
# Copyright © 2022 mightyKlngRichard <dimapermyakov55@gmail.com>

from flask import Flask
from src.functions import fibonacci

app = Flask(__name__)

@app.route('/')
def index():
    return 'Hello World'

@app.route('/num/<number>')
def num(number):
    return str(list(fibonacci(int(number))))[1:-1]
```

Запуск сайта, main.py:

```
# Copyright © 2022 mightyKlngRichard <dimapermyakov55@gmail.com>
from app import app

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

Результат выполнения программы:

