** 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 № ИУ5-35Б

Обухов Антон

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](https://requests.readthedocs.io/en/latest/) и визуализацию полученных от веб-сервиса данных с использованием библиотеки [matplotlib.](https://matplotlib.org/)

**Код программы:**

import requests

def fibonacci(end\_cycle):

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)

**Функции.fiba.py:**

import unittest

from Функции.func 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'))

**Функции.numb.py:**

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()

**f1.py:**

from flask import Flask

from Функции.func 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, fstart.py:**  
from f1 import app

if \_\_name\_\_ == "\_\_main\_\_":

app.run()

**Результат выполнения программы:**

 