**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию

«**Разработка комплексного приложения на языке Python.**»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б  Казицин Алексей |  | преподаватель кафедры  Гапанюк Ю.Е. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Цель домашнего задания:** разработка комплексного приложения на языке Python.

**Требования к отчету:**

Отчет по домашнему заданию должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

**Задание:**

1. С использованием механизма итераторов или генераторов реализуйте с помощью концепции ленивых вычислений [одну из последовательностей OEIS.](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D1%8D%D0%BD%D1%86%D0%B8%D0%BA%D0%BB%D0%BE%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F_%D1%86%D0%B5%D0%BB%D0%BE%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D1%85_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9) Примером могут являться [числа Фибоначчи.](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8)
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](https://requests.readthedocs.io/en/latest/) и визуализацию полученных от веб-сервиса данных с использованием библиотеки [matplotlib.](https://matplotlib.org/)

Текст программы:

Main.py

from app import app  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run()

Tests.py

import unittest  
from src.fib import fib  
  
class test\_fib(unittest.TestCase):  
 def test\_1(self):  
 fib\_gen = fib()  
 res = [next(fib\_gen) for \_ in range(10)]  
 self.assertEqual(len(res), 10)  
 self.assertEqual(res, [1, 1, 2, 3, 5, 8, 13, 21, 34, 55])  
  
 def test\_2(self):  
 fib\_gen = fib()  
 res = [next(fib\_gen) for \_ in range(3)]  
 self.assertEqual(len(res), 3)  
 self.assertEqual(res, [1, 1, 2])  
  
 def test\_3(self):  
 fib\_gen = fib()  
 res = [next(fib\_gen) for \_ in range(5)]  
 self.assertEqual(len(res), 5)  
 self.assertEqual(res, [1, 1, 2, 3, 5])  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

App.py

from flask import Flask  
from src.fib import fib  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def index():  
 return '<p>This page is returning Fibonacci numbers!</p>'  
  
  
@app.route('/num/<int:cnt>')  
def get\_fib(cnt):  
 fib\_gen = fib()  
 res = [next(fib\_gen) for \_ in range(cnt)]  
 return res

Get\_numbers.py

import requests  
import matplotlib.pyplot as plt  
  
  
def make\_url(cnt):  
 base\_url = 'http://127.0.0.1:5000/num/'  
 res = base\_url + str(cnt)  
 return res  
  
  
def get\_data(cnt):  
 url = make\_url(cnt)  
 r = requests.get(url)  
 return r.json()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 y = get\_data(int(input('Сколько чисел Фибоначчи?')))  
 print(y)  
 print(len(y))  
 x = list(range(1, len(y) + 1))  
 fig = plt.figure(figsize=(10, 5))  
 plt.bar(x, y)  
 plt.xlabel('Ось абсцисс')  
 plt.ylabel('Ось ординат')  
 plt.title('Первые {} чисел последовательности Фибоначчи'.format(len(y)))  
 plt.show()  
  
 plt.plot(x, y)  
 plt.show()

Fib.py

import requests  
  
def fib():  
 *'''  
 Генераторная функция для чисел Факториал  
 '''* prev, cur = 0, 1  
 while True:  
 yield cur  
 prev, cur = cur, prev + cur  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 fib\_gen = fib()  
 res = [next(fib\_gen) for \_ in range(20)]  
 print(res)

Примеры работы программы:











