



**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 по «Базовые компоненты интернет-технологий»

Выполнил:

студент группы ИУ5-35Б

Нгуен Зуи Лам

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Нардид А.Н.

Подпись и дата:

Москва

2022

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.

Текст программы.

Проект состоит из трех файлов main.py, fib.py, test.py

fib.py

```
def fib():
    prev, cur = 0, 1
    while True:
        yield cur
        prev, cur = cur, prev+cur
def get_fib_number_at_pos(pos):
    fib_gen = fib()
    number = 0
    for i in range(pos):
        number = next(fib_gen)
    return number
```

main.py

```
from fib import fib

from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Write number of fibonacci numbers you want to be computed after the / symbol</p>"

@app.route("/<int:n>")
```

```
def fibonacci_number(n):
    fib_gen = fib()
    fib_numbers = []
    for i in range(n):
        fib_numbers.append(next(fib_gen))
    return fib_numbers

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

test.py

```
import unittest
from fib import get_fib_number_at_pos

class FibTestCase(unittest.TestCase):
    def test_number_1(self):
        self.assertEqual(1, get_fib_number_at_pos(1))
    def test_number_3(self):
        self.assertEqual(2, get_fib_number_at_pos(3))
    def test_number_7(self):
        self.assertEqual(13, get_fib_number_at_pos(7))
    def test_number_8(self):
        self.assertEqual(21, get_fib_number_at_pos(8))
    def test_number_9(self):
        self.assertEqual(34, get_fib_number_at_pos(9))
```

Jupyter notebook.

Untitled.ipynb

```
In [8]: import requests
import matplotlib.pyplot as plt
def make_url(c):
    st_url = 'http://127.0.0.1:5000/'
    res = st_url + str(c)
    return res

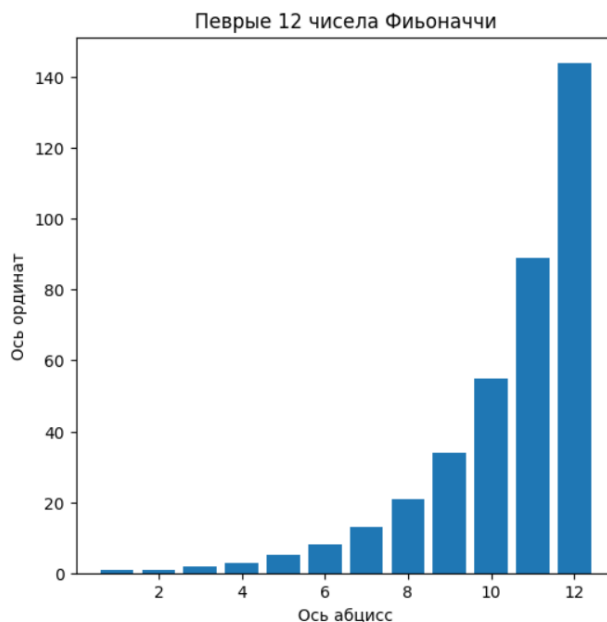
def get_data(c):
    url = make_url(c)
    req = requests.get(url)
    return req.json()
```

```
In [9]: c_list = [5]
for c in c_list:
    print('{} числа Фибоначчи: {}'.format(c, get_data(c)))

5 числа Фибоначчи: [1, 1, 2, 3, 5]
```

```
In [7]: num = 12
os_y = get_data(num)
os_x = list(range(1, len(os_y)+ 1))

fig = plt.figure(figsize = (6, 6))
plt.xlabel('Ось абсцисс')
plt.ylabel('Ось ординат')
plt.title('Первые {} числа Фибоначчи'.format(len(os_y)))
plt.bar(os_x, os_y)
plt.show()
```



Результат выполнения.

Test.py

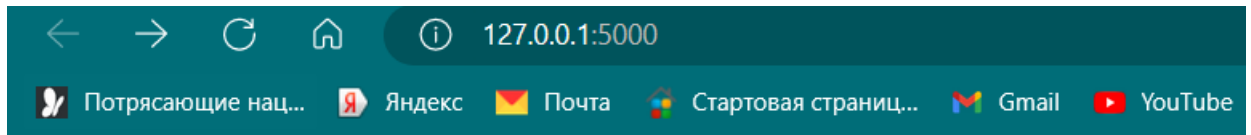
```
D:\PyCharm\DZ\venv\Scripts\python.exe "D:/PyCharm/PyCharm Community Edition 2022.2.2/plugins/python-ce/helpers/pycharm/_jb_unittest_runner.py"
--path D:\PyCharm\DZ\test.py
Testing started at 0:56 ...

Ran 5 tests in 0.003s

OK
```

Main.py

```
D:\PyCharm\02\venv\Scripts\python.exe D:\PyCharm\02\main.py
* Serving Flask app 'main'
* 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: 142-362-580
127.0.0.1 - - [20/Dec/2022 00:57:23] "GET / HTTP/1.1" 200 -
```



Write number of fibonacci numbers you want to be computed after the / symbol

