

**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:
студент группы ИУ5-32Б
Казицин Алексей

Проверил:
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](#). 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:

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:

```
In [1]: !pip install requests
```

```
Requirement already satisfied: requests in c:\users\lesha\pycharmprojects\bkitdz\venv\lib\site-packages (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\lesha\pycharmprojects\bkitdz\venv\lib\site-packages (from r
equests) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\lesha\pycharmprojects\bkitdz\venv\lib\site-packages (from requ
ests) (1.26.13)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lesha\pycharmprojects\bkitdz\venv\lib\site-packages (from request
s) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in c:\users\lesha\pycharmprojects\bkitdz\venv\lib\site-packages (from requests) (3.
4)
```

```
In [2]: import requests
import matplotlib.pyplot as plt
```

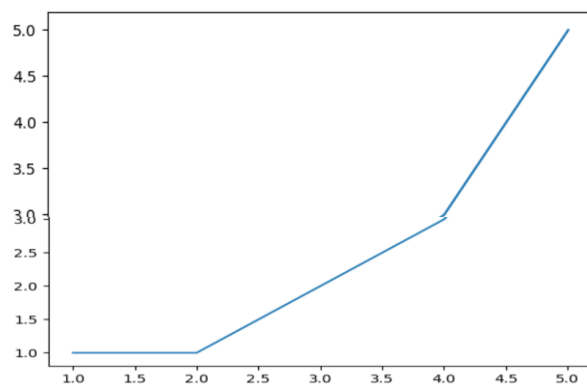
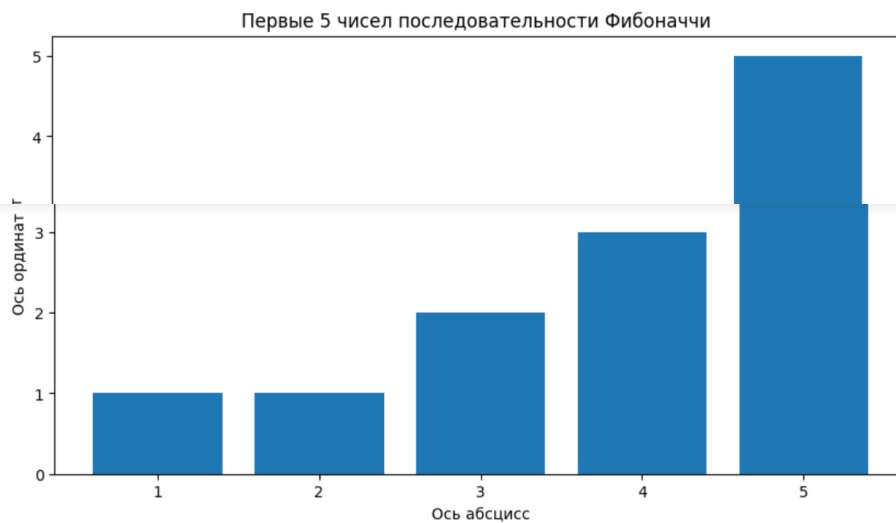
```
def make_url(cnt):
    base_url = 'http://127.0.0.1:5000/num/'
    res = base_url + str(cnt)
    return res
```

```
def get_data(cnt):
    url = make_url(cnt)
    r = requests.get(url)
    return r.json()
```

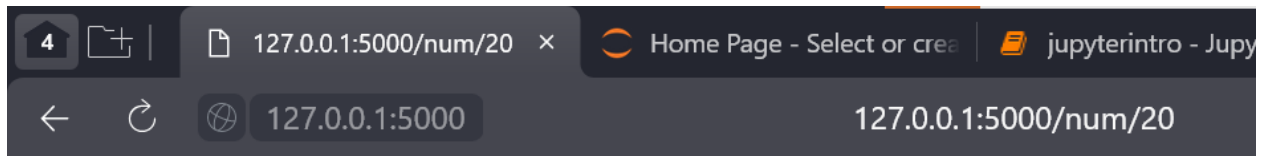
```
if __name__ == '__main__':
    y = get_data(int(input('Сколько чисел Фибоначчи?')))
    print(y)
    print(len(y))
    x = list(range(1, len(y) + 1))
    fig = plt.figure(figsize=(10, 5))
    plt.bar(x, y)
    plt.xlabel('Ось абсцисс')
    plt.ylabel('Ось ординат')
    plt.title('Первые {} чисел последовательности Фибоначчи'.format(len(y)))
    plt.show()

    plt.plot(x, y)
    plt.show()
```

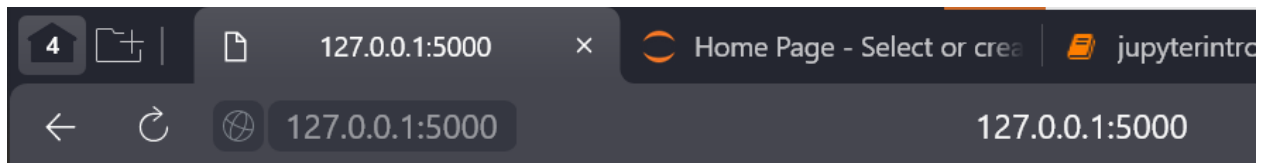
```
Сколько чисел Фибоначчи?5
[1, 1, 2, 3, 5]
5
```



```
In [ ]:
```



[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]



This page is returning Fibonacci numbers!