

МГТУ им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Домашнее задание
«Базовые компоненты интернет-технологий»

Студент группы ИУ5-31Б:
Строганов Георгий Константинович

Преподаватель кафедры ИУ5:
Гапанюк Юрий Евгеньевич

Москва, 2022

Условия ДЗ

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.

Листинг программы

Flask_web.py

```
from flask import Flask
from generator import fib
app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

@app.route("/fibonachi")
def fibonachi_start():
    return "Write after URL '/number!'"

@app.route("/fibonachi/<int:n>")
def fibonachi_number(n):
    return list(fib(n))

@app.errorhandler(404)
def page_not_found(e):
    return "Oops! Try to enter a '/fibonachi/number!'"
```

generator.py

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        yield a
        a, b = b, a + b
```

test_TDD.py

```
# -*- coding: cp1251 -*-
import pytest
from time import time
from generator import fib
class cm_timer:
    def __enter__(self):
        self.__time_begin = time()
    def __exit__(self, type, value, traceback):
```

```

        print(time() - self.__time_begin)

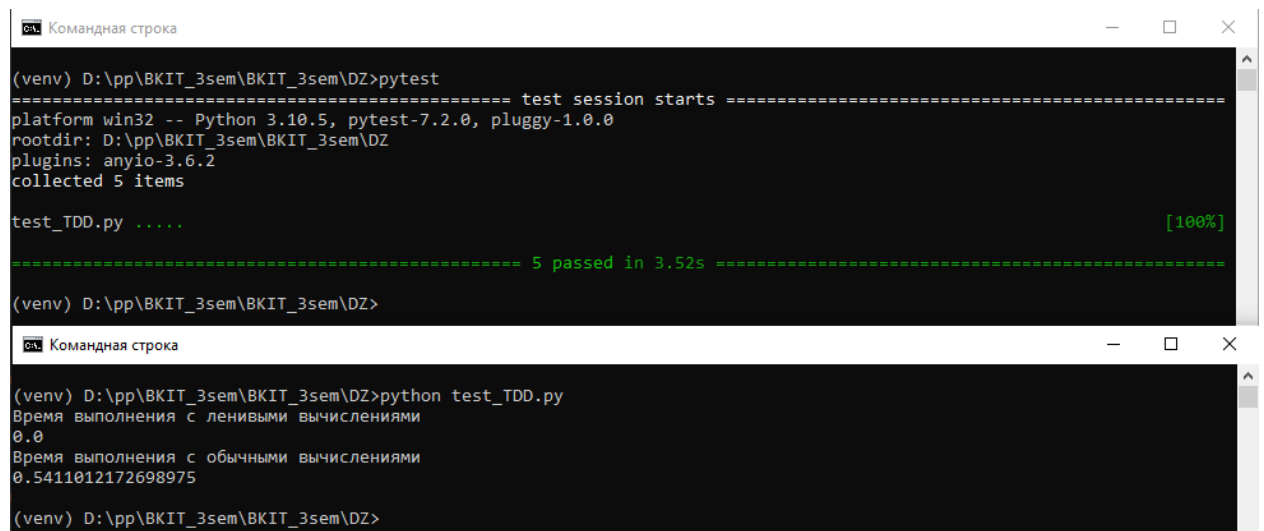
c = 100000
# тестирование результата выполнения
def test_fib_1():
    assert [i for i in fib(5)] == [0, 1, 1, 2, 3]
def test_fib_2():
    assert [i for i in fib(10)] == [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
def test_fib_3():
    assert [i for i in fib(0)] == []
# тестирование на ленивые выражения (запускать не через pytest чтобы увидеть время выполнения)
def test_fib_time_1():
    print("Время выполнения с ленивыми вычислениями")
    with cm_timer():
        temp = fib(c)
    assert list(fib(c)) == [i for i in fib(c)]
def test_fib_time_2():
    print("Время выполнения с обычными вычислениями")
    with cm_timer():
        temp = [i for i in fib(c)]
    assert [i for i in fib(c)] == list(fib(c))

if __name__ == "__main__":
    test_fib_time_1()
    test_fib_time_2()

```

Запуск файла с тестированием

При запуске **test_TDD.py** через **pytest** производится тестирование на правильность работы файла **generator.py**, а при запуске просто как исполняемый файл выводится информация о скорости вычисления с обычными и с ленивыми вычислениями.



The image shows two screenshots of a Windows command prompt window titled "Командная строка".

The top screenshot shows the output of running `pytest` in a virtual environment. The output includes the pytest version (7.2.0), the platform (win32), and the Python version (3.10.5). It shows that 5 items were collected and all tests passed. The summary line indicates "5 passed in 3.52s".

```

(venv) D:\pp\BKIT_3sem\BKIT_3sem\DZ>pytest
===== test session starts =====
platform win32 -- Python 3.10.5, pytest-7.2.0, pluggy-1.0.0
rootdir: D:\pp\BKIT_3sem\BKIT_3sem\DZ
plugins: anyio-3.6.2
collected 5 items

test_TDD.py ..... [100%]

===== 5 passed in 3.52s =====
(venv) D:\pp\BKIT_3sem\BKIT_3sem\DZ>

```

The bottom screenshot shows the output of running `python test_TDD.py` in the same virtual environment. It displays the execution time for the lazy calculations (0.0) and the execution time for the regular calculations (0.5411012172698975).

```

(venv) D:\pp\BKIT_3sem\BKIT_3sem\DZ>python test_TDD.py
Время выполнения с ленивыми вычислениями
0.0
Время выполнения с обычными вычислениями
0.5411012172698975
(venv) D:\pp\BKIT_3sem\BKIT_3sem\DZ>

```

Запуск Flask и Jupyter Notebook

```
Выбрать Командная строка - Flask --app Flask_web run
(venv) D:\pp\BKIT_3sem\BKIT_3sem\DZ>Flask --app Flask_web run
* Serving Flask app 'Flask_web'
* Debug mode: off
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

Командная строка - jupyter notebook
(venv) D:\pp\BKIT_3sem\BKIT_3sem\DZ>jupyter notebook
[I 21:59:16.706 NotebookApp] Serving notebooks from local directory: D:\pp\BKIT_3sem\BKIT_3sem\DZ
[I 21:59:16.706 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 21:59:16.707 NotebookApp] http://localhost:8888/?token=886ca37e3b770ae5bcf1e1f6f50c20b2a1559610f4cb4d38
[I 21:59:16.707 NotebookApp] or http://127.0.0.1:8888/?token=886ca37e3b770ae5bcf1e1f6f50c20b2a1559610f4cb4d38
[I 21:59:16.707 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:59:16.797 NotebookApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/strog/AppData/Roaming/jupyter/runtime/nbserver-9876-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=886ca37e3b770ae5bcf1e1f6f50c20b2a1559610f4cb4d38
    or http://127.0.0.1:8888/?token=886ca37e3b770ae5bcf1e1f6f50c20b2a1559610f4cb4d38
```

Переход по <http://127.0.0.1:5000/> и т.п.

```
< > ↻ 🏠 ! 127.0.0.1:5000
```

```
📄 Hello, World!
```

```
< > ↻ 🏠 ! 127.0.0.1:5000/edfv
```

```
📄 Oops! Try to enter a '/fibonachi/number!'
```

```
< > ↻ 🏠 ! 127.0.0.1:5000/fibonachi
```

```
📄 Write after URL '/number!'
```

```
< > ↻ 🏠 ! 127.0.0.1:5000/fibonachi/10
```

```
📄 [0,1,1,2,3,5,8,13,21,34]
```

```
< > ↻ 🏠 ! 127.0.0.1:5000/fibonachi/30
📄 [0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181,6765,10946,17711,28657,46368,75025,121393,196418,317811,514229]
```

Jupyter Notebook

Смотреть в разделе: **BKIT_3sem/DZ/Pylounge.ipynb**