**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 предмету

«Базовые компоненты интернет-технологий»

Выполнил:

студент группы ИУ5-31Б

Изибаев Андрей

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2022 г.

**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.

**Код программы**

**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 результатов 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 выражение  
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()

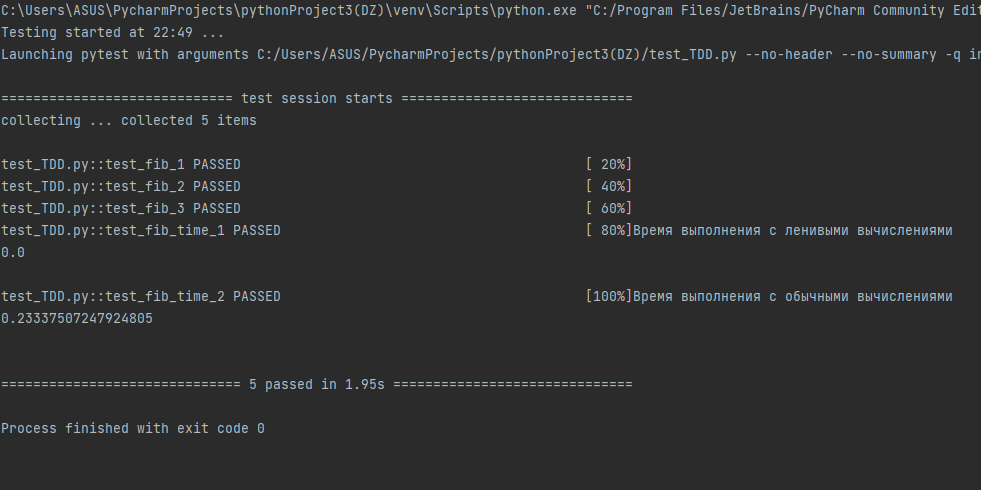
**Flask228.py**

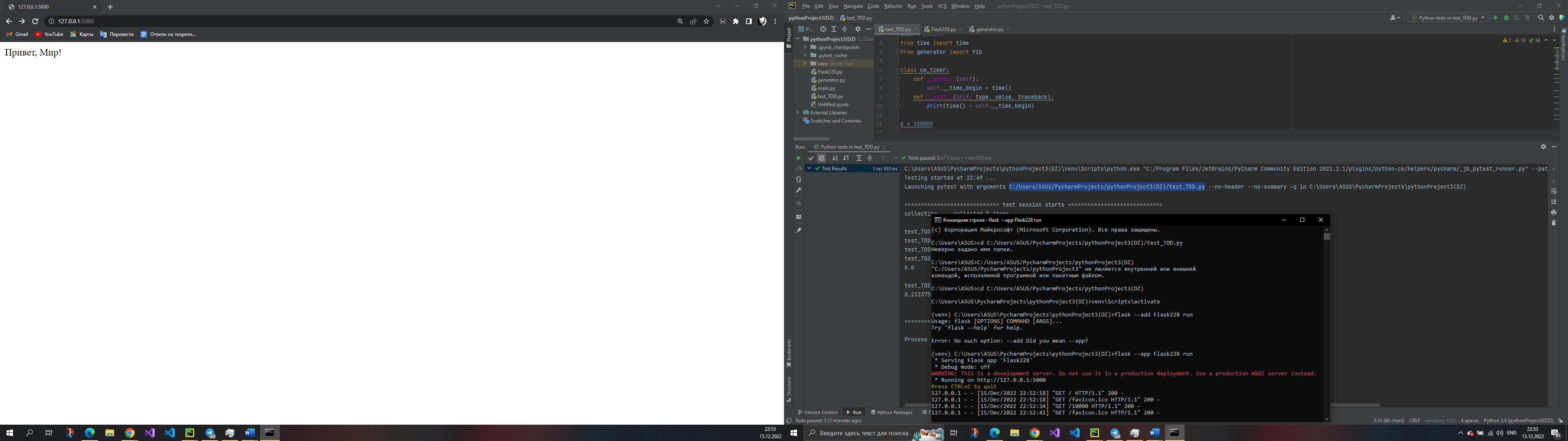
from flask import Flask  
from generator import fib  
app = Flask(\_\_name\_\_)  
  
@app.route("/")  
def hello\_world():  
 return "<p>Привет, Мир!</p>"  
  
@app.route("/<int:n>")  
def fibonachi\_number(n):  
 return list(fib(n))  
  
@app.errorhandler(404)  
def page\_not\_found(e):  
 return "Ошибка! Введите '/число!'"

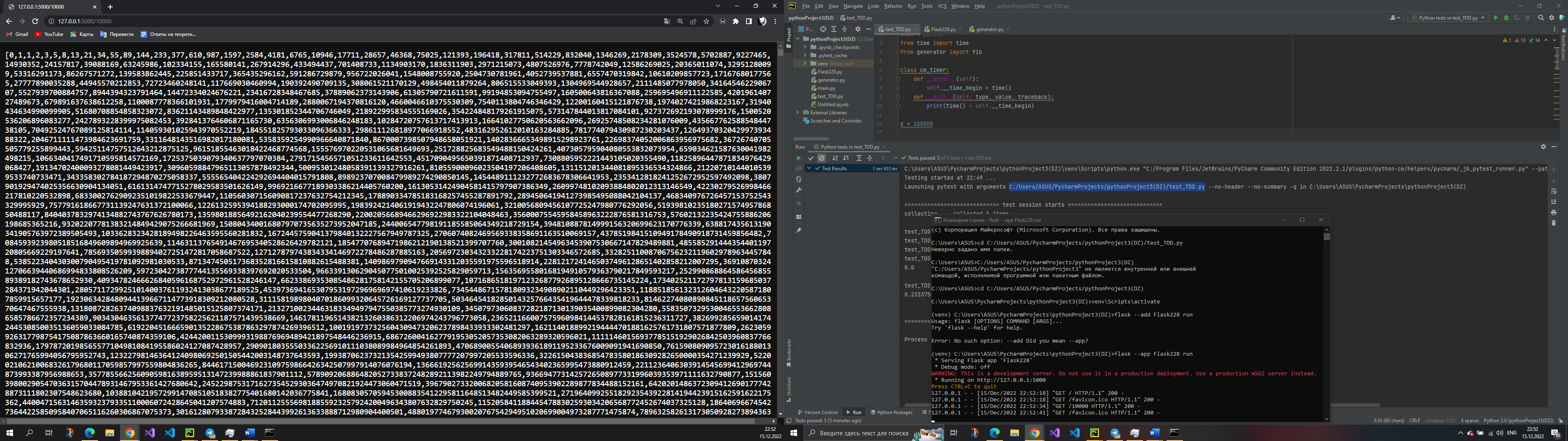
**generator.py**

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

**Результаты выполнения**







Juputer Notebook смотреть в разделе ДЗ “untitled.jpynb”