**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: Проверил:

студент группы ИУ5Ц-52Б преподаватель каф. ИУ5

Павлов Сергей Алексеевич Гапанюк Юрий Евгеньевич

Подпись и дата: 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.

1. Для реализованной последовательности разработайте 3-5 модульных тестов, которые, в том числе, проверяют то, что последовательность

поддерживает ленивые вычисления.

1. Разработайте веб-сервис с использованием фреймворка Flask, который возвращает N элементов последовательности (параметр N передается в запросе к сервису).
2. 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\_seq.py, fib\_seq\_test.py

**Fib\_seq.py**

*# функция-генератор позволяющая создать бесконечную последовательность Фибоначчи*def fibonacci\_seq():  
 x, y = 0, 1  
 while True:  
 yield x  
 x, y = y, x + y

**main.py**

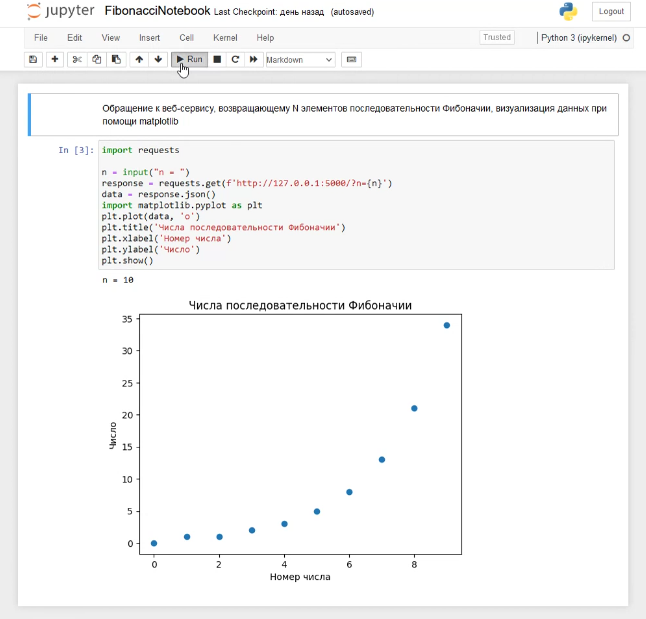
from flask import Flask, request  
  
from fib.fib\_seq import fibonacci\_seq  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route('/')  
def fib():  
 n = int(request.args.get('n', default=0))  
 fib\_gen = fibonacci\_seq()  
 return [next(fib\_gen) for \_ in range(n)]  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run()

**fib\_seq\_test.py**

from fib.fib\_seq import fibonacci\_seq  
import types  
  
  
def test\_first():  
 fib\_gen = fibonacci\_seq()  
 assert next(fib\_gen) == 0  
  
  
def test\_first\_5():  
 fib\_gen = fibonacci\_seq()  
 assert [next(fib\_gen) for \_ in range(5)] == [0, 1, 1, 2, 3]  
  
  
def test\_first\_25():  
 fib\_gen = fibonacci\_seq()  
 assert [next(fib\_gen) for \_ in range(25)] == [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]  
  
  
def test\_first\_100():  
 fib\_gen = fibonacci\_seq()  
 assert [next(fib\_gen) for \_ in range(100)] == [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, 832040, 1346269, 2178309, 3524578, 5702887,  
 9227465, 14930352, 24157817, 39088169, 63245986, 102334155,  
 165580141, 267914296, 433494437, 701408733, 1134903170, 1836311903,  
 2971215073, 4807526976, 7778742049, 12586269025, 20365011074,  
 32951280099, 53316291173, 86267571272, 139583862445, 225851433717,  
 365435296162, 591286729879, 956722026041, 1548008755920,  
 2504730781961, 4052739537881, 6557470319842, 10610209857723,  
 17167680177565, 27777890035288, 44945570212853, 72723460248141,  
 117669030460994, 190392490709135, 308061521170129, 498454011879264,  
 806515533049393, 1304969544928657, 2111485077978050,  
 3416454622906707, 5527939700884757, 8944394323791464,  
 14472334024676221, 23416728348467685, 37889062373143906,  
 61305790721611591, 99194853094755497, 160500643816367088,  
 259695496911122585, 420196140727489673, 679891637638612258,  
 1100087778366101931, 1779979416004714189, 2880067194370816120,  
 4660046610375530309, 7540113804746346429, 12200160415121876738,  
 19740274219868223167, 31940434634990099905, 51680708854858323072,  
 83621143489848422977, 135301852344706746049, 218922995834555169026]  
  
  
def test\_lazy():  
 assert isinstance(fibonacci\_seq(), types.GeneratorType)

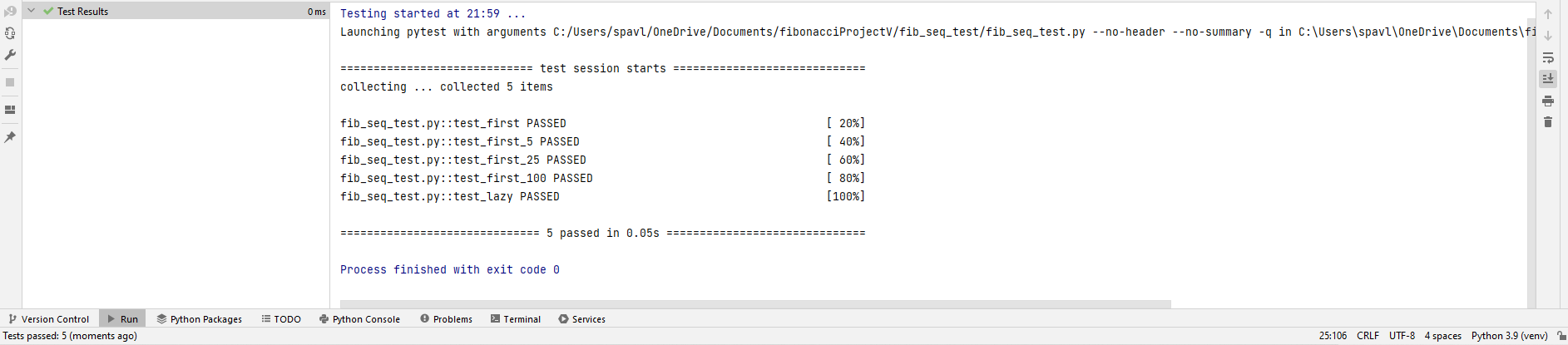
**Jupyter notebook.**

**FibonacciNotebook.ipynb**



# Результат выполнения.

fib\_seq\_test.py



Main.py

