|  |  |
| --- | --- |
|  | **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 ИУ5-35Б преподаватель каф. ИУ5

Нгуен Зуи Лам Нардид А.Н.

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.
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**

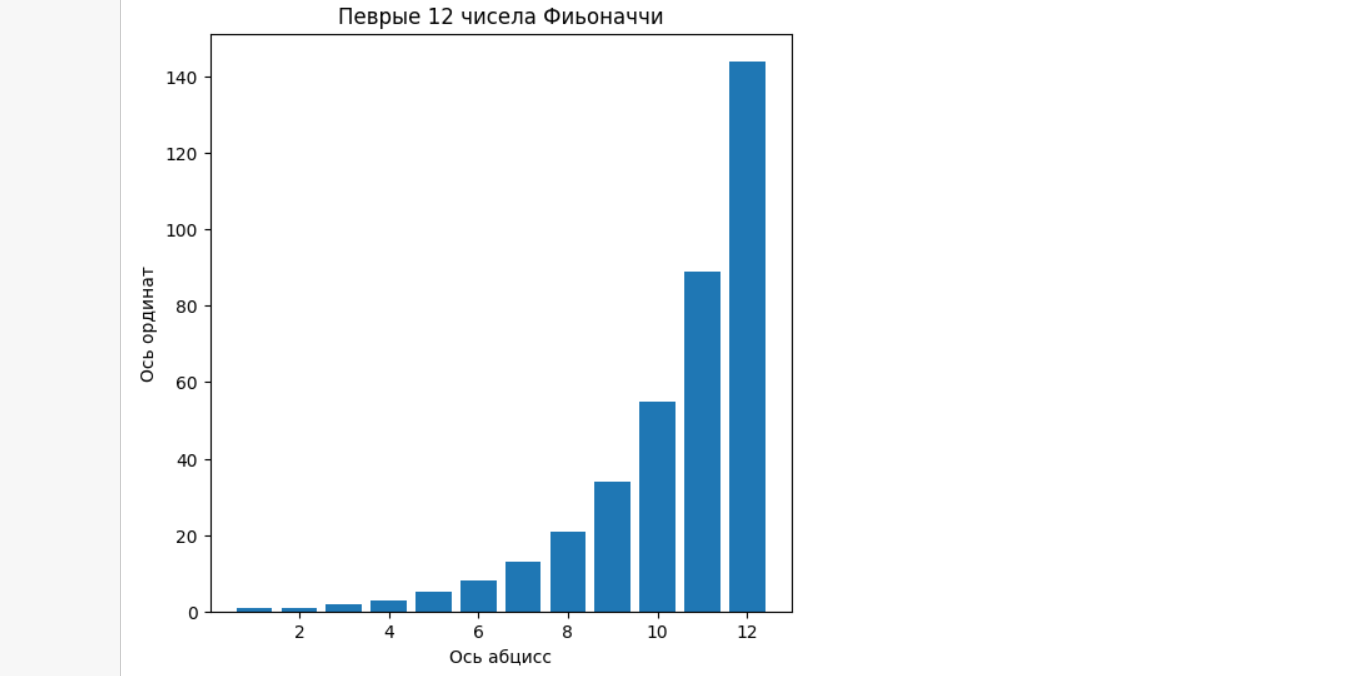
from fib import fib  
  
from flask import Flask  
  
app = Flask(\_\_name\_\_)  
  
@app.route("/")  
def hello\_world():  
 return "<p>Write number of fibonachi numbers you want to be computed after the / symbol</p>"  
  
@app.route("/<int:n>")  
def fibonachi\_number(n):  
 fib\_gen = fib()  
 fib\_numbers = []  
 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))

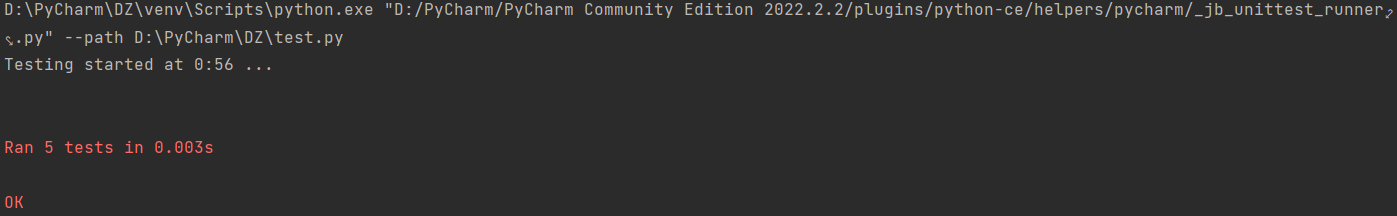
**Jupyter notebook.**

**Untitled.ipynb**

****

**Результат выполнения.**

Test.py



Main.py

