



**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 курсу
«Базовые компоненты интернет технологий»

Выполнила:
студент группы № ИУ5-33Б
Балюк А.В

Проверил:
Преподаватель
Гапанюк Ю.Е

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.

Листинг :

```
from flask import Flask, request, render_template, redirect
app = Flask(__name__)

def fib(n):
    if n <= 0:
        pass
    elif n == 1:
        yield 0
    else:
        prev, cur = 0, 1
        yield prev
        for i in range(n-1):
            yield cur
            prev, cur = cur, prev+cur

@app.route('/first')
def fib_1():
    # http://127.0.0.1:5000/first
    n = 9
    return f'<h1>{n} первых чисел Фибоначчи: {str(list(fib(n)))[1:-1]}<h1>'

@app.route('/second')
def fib_2():
    # http://127.0.0.1:5000/second?count=14
    n = int(request.args.get('count'))
    return f'<h1>{n} первых чисел Фибоначчи: {str(list(fib(n)))[1:-1]}<h1>'

if __name__ == '__main__':
    app.run()
```

Тесты :

```
from main import fib
import unittest
import types
```

```

class Test_fib(unittest.TestCase):
    def test_not(self):
        self.n = -1
        self.assertEqual(list(fib(self.n)), [])
    def test1(self):
        self.n = 5
        correct = [0, 1, 1, 2, 3]
        self.assertEqual(list(fib(self.n)), correct)
    def test2(self):
        self.n = 10
        correct = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
        self.assertEqual(list(fib(self.n)), correct)
    def test3(self):
        self.n = 29
        correct = [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]
        self.assertEqual(list(fib(self.n)), correct)
    def test_lazy(self):
        self.n = 5
        correct = [0, 1, 1, 2, 3]
        self.assertEqual(type(fib(self.n)), types.GeneratorType)

if __name__ == '__main__':
    unittest.main()

```

Результаты тестов:

```

Ran 5 tests in 0.028s

OK

Process finished with exit code 0

```

Результаты программы:

```


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

```

14 первых чисел Фибоначчи: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

```
(Envs) C:\Users\Андрей\Desktop\dz\Envs\Scripts>jupyter-notebook
[I 16:36:11.479 NotebookApp] Serving notebooks from local directory: C:\Users\Андрей\Desktop\dz\Envs\Scripts
[I 16:36:11.479 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 16:36:11.479 NotebookApp] http://localhost:8888/?token=0e8284b6520c4bb9e1ec00854bac138ee12ac9590d169edc
[I 16:36:11.479 NotebookApp] or http://127.0.0.1:8888/?token=0e8284b6520c4bb9e1ec00854bac138ee12ac9590d169edc
[I 16:36:11.479 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:36:11.537 NotebookApp]
```

To access the notebook, open this file in a browser:
<file:///C:/Users/%D0%90%D0%BD%D0%B4%D1%80%D0%B5%D0%B9/AppData/Roaming/jupyter/runtime/nbserver-28652-open.html>
Or copy and paste one of these URLs:
<http://localhost:8888/?token=0e8284b6520c4bb9e1ec00854bac138ee12ac9590d169edc>
or <http://127.0.0.1:8888/?token=0e8284b6520c4bb9e1ec00854bac138ee12ac9590d169edc>

 jupyter

Quit

Logout

FilesRunningClusters

Select items to perform actions on them.

0

/

	Name	Last Modified	File size
<input type="checkbox"/>	Untitled.ipynb	17 дней назад	86.1 kB
<input type="checkbox"/>	activate	18 дней назад	2.16 kB
<input type="checkbox"/>	activate.bat	18 дней назад	1 kB
<input type="checkbox"/>	activate.fish	18 дней назад	3.04 kB
<input type="checkbox"/>	activate.nu	18 дней назад	1.32 kB
<input type="checkbox"/>	activate.ps1	18 дней назад	1.76 kB
<input type="checkbox"/>	activate_this.py	18 дней назад	1.19 kB
<input type="checkbox"/>	behave.exe	18 дней назад	106 kB
<input type="checkbox"/>	deactivate.bat	18 дней назад	510 B
<input type="checkbox"/>	deactivate.nu	18 дней назад	333 B
<input type="checkbox"/>	tzpy.exe	18 дней назад	106 kB
<input type="checkbox"/>	flask.exe	18 дней назад	106 kB
<input type="checkbox"/>	fonttools.exe	17 дней назад	106 kB
<input type="checkbox"/>	lpython.exe	17 дней назад	106 kB
<input type="checkbox"/>	lpython3.10.exe	17 дней назад	106 kB
<input type="checkbox"/>	lpython3.exe	17 дней назад	106 kB
<input type="checkbox"/>	jsonpointer	17 дней назад	1.88 kB
<input type="checkbox"/>	jsonschema.exe	17 дней назад	106 kB
<input type="checkbox"/>	jupyter-bundlerextension.exe	17 дней назад	106 kB
<input type="checkbox"/>	jupyter-console.exe	17 дней назад	106 kB
<input type="checkbox"/>	jupyter-dejavu.exe	17 дней назад	106 kB
<input type="checkbox"/>	jupyter-events.exe	17 дней назад	106 kB
<input type="checkbox"/>	jupyter-execute.exe	17 дней назад	106 kB
<input type="checkbox"/>	jupyter-kernel.exe	17 дней назад	106 kB



Обращение к веб-сервису Flask и визуализация результатов

Получение данных ¶

Устанавливаем библиотеку requests и затем импортируем её

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

```
Requirement already satisfied: requests in c:\users\андрей\desktop\dz\envs\lib\site-packages (2.28.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from requests) (1.26.13)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from requests) (2022.12.7)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from requests) (2.1.1)
Note: you may need to restart the kernel to use updated packages.
```

Найти в Яндексе

Копировать

В заметки

pip version 21.3.1; however, version 22.3.1 is available.

You should consider upgrading via the 'C:\Users\Андрей\Desktop\dz\Env\Scripts\python.exe -m pip install --upgrade pip' command.

```
In [2]: import requests
```

Получаем результат сервиса

```
In [3]: param = {'count': '14'}
# Результат запроса
r = requests.get('http://127.0.0.1:5000/second', params=param)
# Убираем html-тег
buff = r.text[4:-4]
print(buff)
```

14 первых чисел Фибоначчи: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

Преобразуем текст в нужные нам числа

```
In [4]: res = [int(i) for i in buff.split(' ')]
```

Визуализация при помощи matplotlib

```
In [5]: pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\андрей\desktop\dz\envs\lib\site-packages (3.6.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (9.3.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (4.38.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: packaging>=20.0 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (22.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (1.0.6)
Requirement already satisfied: numpy>=1.19 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from matplotlib) (1.23.5)
Requirement already satisfied: six>=1.5 in c:\users\андрей\desktop\dz\envs\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

WARNING: You are using pip version 21.3.1; however, version 22.3.1 is available.

You should consider upgrading via the 'C:\Users\Андрей\Desktop\dz\Env\Scripts\python.exe -m pip install --upgrade pip' command.

```
In [6]: import matplotlib.pyplot as plt
```

Зададим данные для осей

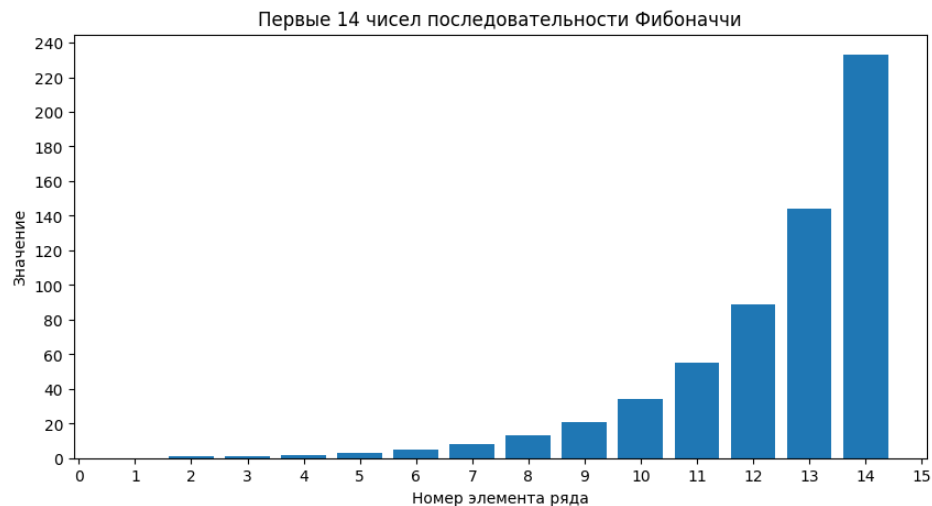
```
In [7]: x = [i+1 for i in range(len(res))]
y = res
```

Построим столбчатую диаграмму

```
In [8]: fig = plt.figure(figsize = (10, 5))
plt.bar(x, y)
plt.xlabel('Номер элемента ряда')
plt.ylabel('Значение')
plt.title('Первые {} чисел последовательности Фибоначчи'.format(len(y)))
#Зададим цену деления
xy = plt.gca()
```

Построим столбчатую диаграмму

```
In [8]: fig = plt.figure(figsize = (10, 5))
plt.bar(x, y)
plt.xlabel('Номер элемента ряда')
plt.ylabel('Значение')
plt.title('Первые {} чисел последовательности Фибоначчи'.format(len(y)))
#Зададим цену деления
xy = plt.gca()
xy.xaxis.set_major_locator(plt.MultipleLocator(1))
xy.yaxis.set_major_locator(plt.MultipleLocator(20))
plt.show()
```



Построим график

Построим график

```
In [9]: fig = plt.figure(figsize = (10, 5)) # размер
plt.plot(x, y, c='red') # график
plt.xlabel('Порядковый номер элемента ряда')
plt.ylabel('Значение')
plt.title('Первые {} чисел последовательности Фибоначчи'.format(len(y)))
# Зададим цену деления
xy = plt.gca()
xy.xaxis.set_major_locator(plt.MultipleLocator(1))
xy.yaxis.set_major_locator(plt.MultipleLocator(20))
plt.show()
```

