



**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 ИУ5-35Б
Сулайманов Роман

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

Программа состоит из нескольких файлов.

fibonacci.py

```
def fibonacci(n):
    if n <= 0 or type(n) not in [int]:
        raise ValueError

    num1, num2 = 0, 1
    for _ in range(n):
        num1, num2 = num2, num2 + num1
        yield num1

if __name__ == '__main__':
    print(*fibonacci(5))
```

test.py

```
import unittest, time
from fibonacci import fibonacci

class TEST(unittest.TestCase):
    def test1(self):
        self.assertEqual(list(fibonacci(5)), [1, 1, 2, 3, 5])

    def test2(self):
        with self.assertRaises(ValueError):
            list(fibonacci(-1))

    def test3(self):
        start_time = time.time()
        fibonacci(10000)
        self.assertLess(time.time() - start_time, 1)

if __name__ == "__main__":
    unittest.main()
```

app.py

```
from flask import Flask, request
from fibonacci import fibonacci

app = Flask(__name__)

@app.route('/')
def app_hello():
    return 'Hello, World!'

@app.route('/fibonacci')
def app_fibonacci():
    n = request.args.get('n')
    return ' '.join(map(str, fibonacci(int(n))))

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

Homework.ipynb

Последовательность Фибоначчи

```
import requests
import matplotlib.pyplot as plt

def get_data(n):
    url = f'http://127.0.0.1:5000/fibonacci?n={str(n)}'
    request = requests.get(url)
    return list(map(int, request.text.split(' ')))
```

Числа Фибоначчи

```
In [16]: num = 10
print(f'Первые 10 чисел последовательности Фибоначчи: {get_data(num)}')
```

Первые 10 чисел последовательности Фибоначчи: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

```
In [15]: num = 20
print(f'Первые 20 чисел последовательности Фибоначчи: {get_data(num)}')
```

Первые 20 чисел последовательности Фибоначчи: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]

```
In [17]: num = 30
print(f'Первые 30 чисел последовательности Фибоначчи: {get_data(num)}')
```

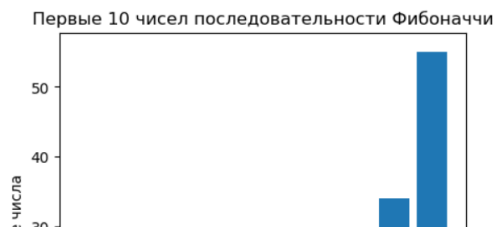
```
In [17]: num = 30
print(f'Первые 30 чисел последовательности Фибоначчи: {get_data(num)}')
```

Первые 30 чисел последовательности Фибоначчи: [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]

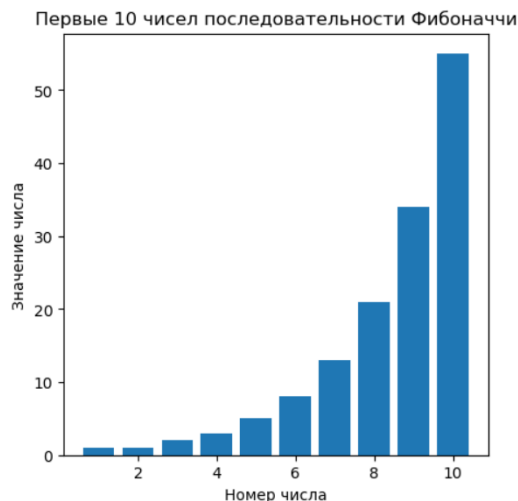
Построение графика

```
In [20]: num = 10
y_axis = get_data(num)
x_axis = [i for i in range(1, num + 1)]

figure = plt.figure(figsize = (5, 5))
plt.bar(x_axis, y_axis)
plt.xlabel('Номер числа')
plt.ylabel('Значение числа')
plt.title('Первые 10 чисел последовательности Фибоначчи')
plt.show()
```



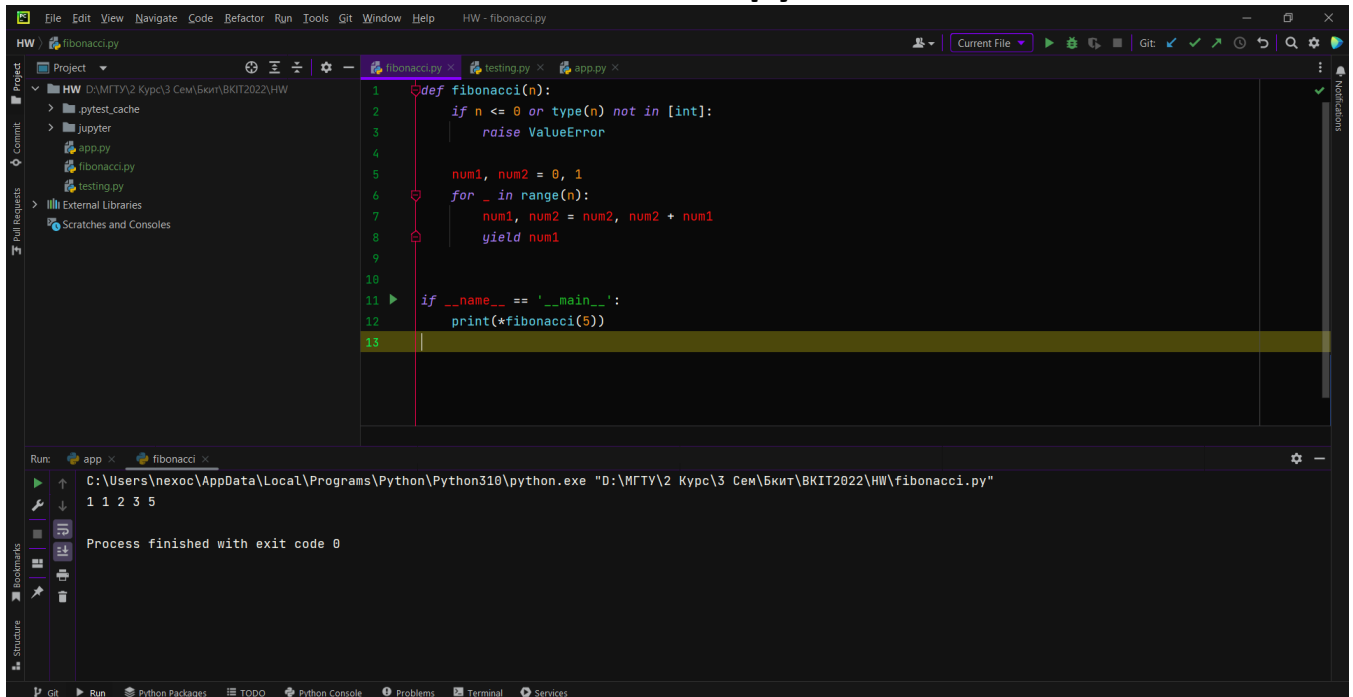
```
plt.xlabel('Номер числа')
plt.ylabel('Значение числа')
plt.title('Первые 10 чисел последовательности Фибоначчи')
plt.show()
```



In []:

Результаты выполнения

fibonacci.py



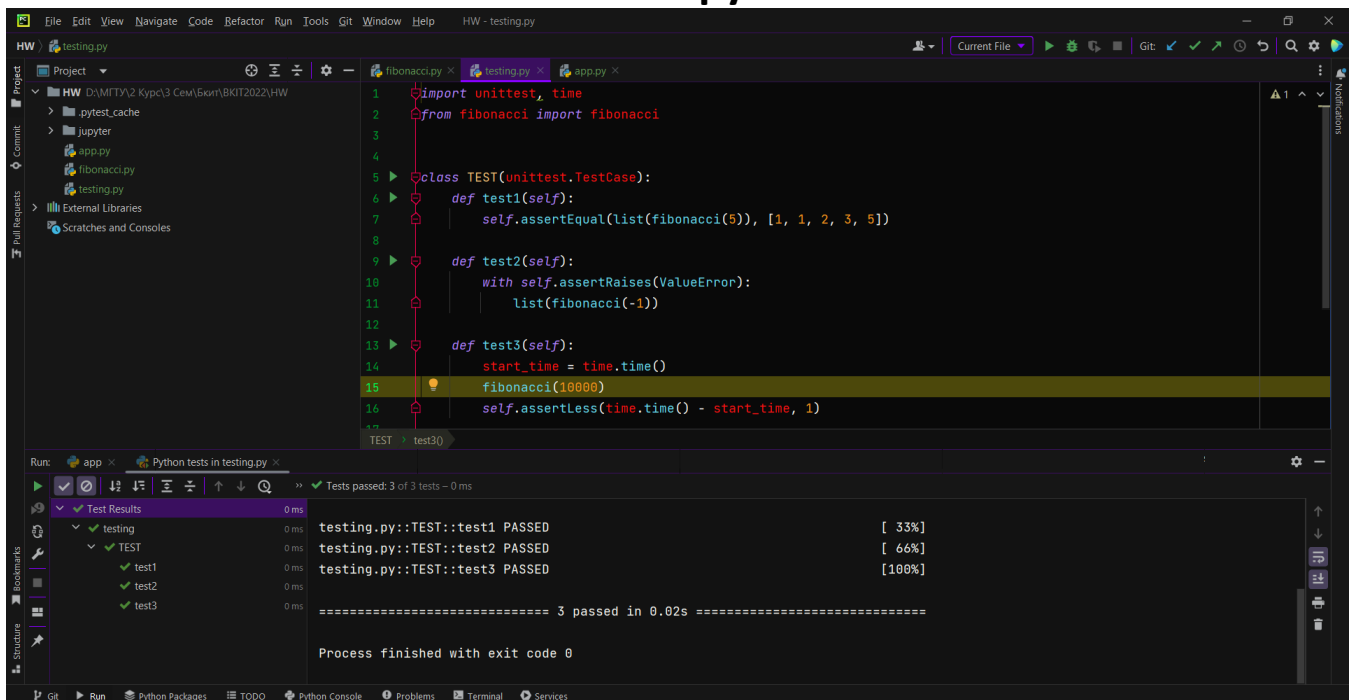
```
1 def fibonacci(n):
2     if n <= 0 or type(n) not in [int]:
3         raise ValueError
4
5     num1, num2 = 0, 1
6     for _ in range(n):
7         num1, num2 = num2, num2 + num1
8         yield num1
9
10
11 if __name__ == '__main__':
12     print(*fibonacci(5))
13
```

Run: C:\Users\nexoc\AppData\Local\Programs\Python\Python310\python.exe "D:\МГТУ\2 Курс\3 Сем\Бкит\BKIT2022\HW\fibonacci.py"

1 1 2 3 5

Process finished with exit code 0

test.py



```
1 import unittest, time
2 from fibonacci import fibonacci
3
4
5 class TEST(unittest.TestCase):
6     def test1(self):
7         self.assertEqual(list(fibonacci(5)), [1, 1, 2, 3, 5])
8
9     def test2(self):
10         with self.assertRaises(ValueError):
11             list(fibonacci(-1))
12
13     def test3(self):
14         start_time = time.time()
15         fibonacci(10000)
16         self.assertLess(time.time() - start_time, 1)
17
```

Run: Python tests in testing.py

Tests passed: 3 of 3 tests - 0 ms

Test Results

Test	Duration	Result
testing.py::TEST::test1	0 ms	PASSED [33%]
testing.py::TEST::test2	0 ms	PASSED [66%]
testing.py::TEST::test3	0 ms	PASSED [100%]

===== 3 passed in 0.02s =====

Process finished with exit code 0

Homework.ipynb

Последовательность Фибоначчи

```
import requests
import matplotlib.pyplot as plt

def get_data(n):
    url = f'http://127.0.0.1:5000/fibonacci?n={str(n)}'
    request = requests.get(url)
    return list(map(int, request.text.split(' ')))
```

Числа Фибоначчи

```
In [16]: num = 10
print(f'Первые 10 чисел последовательности Фибоначчи: {get_data(num)}')
```

Первые 10 чисел последовательности Фибоначчи: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

```
In [15]: num = 20
print(f'Первые 20 чисел последовательности Фибоначчи: {get_data(num)}')
```

Первые 20 чисел последовательности Фибоначчи: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]

```
In [17]: num = 30
print(f'Первые 30 чисел последовательности Фибоначчи: {get_data(num)}')
```

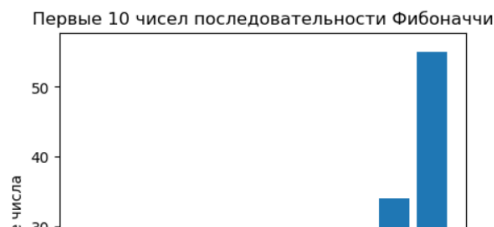
```
In [17]: num = 30
print(f'Первые 30 чисел последовательности Фибоначчи: {get_data(num)}')
```

Первые 30 чисел последовательности Фибоначчи: [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]

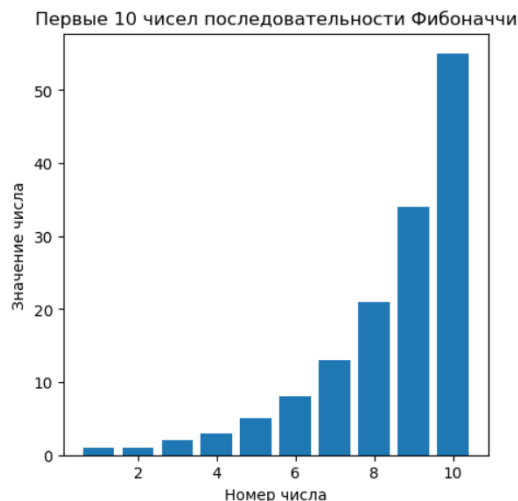
Построение графика

```
In [20]: num = 10
y_axis = get_data(num)
x_axis = [i for i in range(1, num + 1)]

figure = plt.figure(figsize = (5, 5))
plt.bar(x_axis, y_axis)
plt.xlabel('Номер числа')
plt.ylabel('Значение числа')
plt.title('Первые 10 чисел последовательности Фибоначчи')
plt.show()
```



```
plt.xlabel('Номер числа')
plt.ylabel('Значение числа')
plt.title('Первые 10 чисел последовательности Фибоначчи')
plt.show()
```



In []: