



**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  
Клементьев Артем

Проверил:  
Преподаватель кафедры ИУ-5  
Гапанюк Юрий

## 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.

## main.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))
```

## testing.py

```
import unittest
from main import fibonacci

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

    def test2(self):
        self.assertEqual(list(fibonacci(5)), [1, 1, 2, 3, 5])

    def test3(self):
        self.assertEqual(list(fibonacci(7)), [1, 1, 2, 3, 5, 8, 13])

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

## app.py

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

app = Flask(__name__)

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

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

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

# Homework.ipynb

Jupyter Untitled Last Checkpoint: 6 часов назад (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3 (ipykernel)

```
In [9]: 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(' ')))
```

```
In [10]: 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 [11]: num = 20
print(f'Первые 20 чисел последовательности Фибоначчи: {get_data(num)}')
```

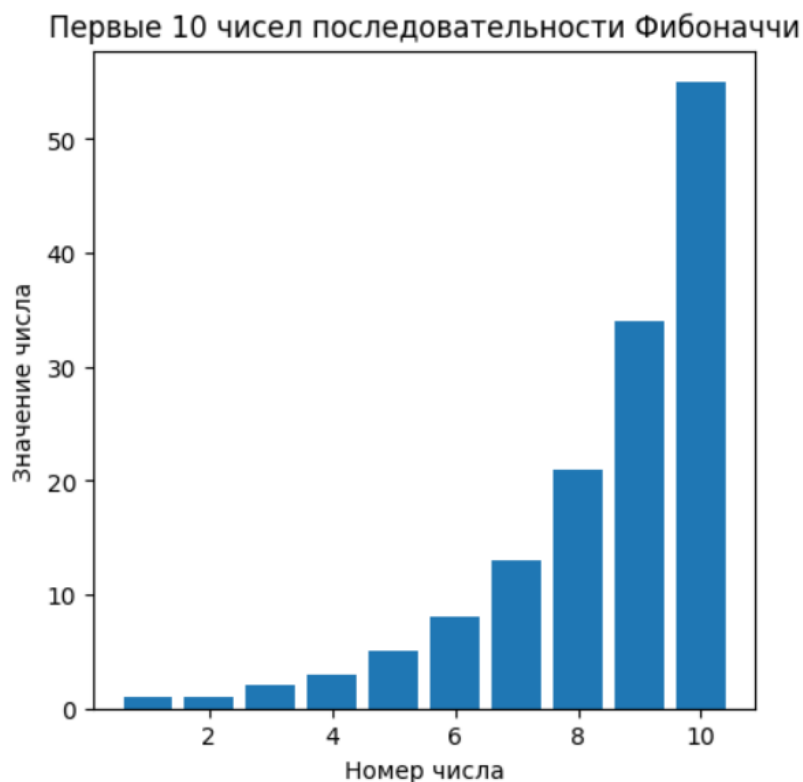
Первые 20 чисел последовательности Фибоначчи: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]

```
In [12]: 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]

```
In [13]: 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()
```



# Результаты выполнения

## main.py

```
C:\Users\jrsyt\PycharmProjects\DZ\venv
1 1 2 3 5

Process finished with exit code 0
```

## testing.py

✓

⊘

↓<sup>a</sup>

↓<sub>≡</sub>

≡

÷

↑

↓

🔍

»

✓ Tests passed: 3 of 3 tests – 1 ms

✓ Test Results1 ms

✓ testing1 ms

✓ TEST1 ms

✓ test11 ms

✓ test20 ms

✓ test30 ms

Ran 3 tests in 0.003s

OK

Process finished with exit code 0