



**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 ИУ5-31Б  
Михалёв Ярослав

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

## fib.py

```
def fibonacci(n):
    if type(n) not in [int]:
        raise TypeError(f"Коэффициент n должен быть натуральным числом!")

    if n <= 0:
        raise ValueError("Коэффициент n должен быть натуральным числом!")

    fib1, fib2 = 0, 1
    for i in range(n):
        fib1, fib2 = fib2, fib1 + fib2
        yield fib1

if __name__ == '__main__':
    print(*fibonacci(10))
```

## tests.py

```
import unittest
import time
from fib import fibonacci

class TestEquation(unittest.TestCase):

    def test_numbers(self):
        self.assertEqual(len(list(fibonacci(10))), 10)
        self.assertEqual(list(fibonacci(5)), [1, 1, 2, 3, 5])

    def test_iteration(self):
        res = fibonacci(2)
        self.assertEqual(next(res), 1)
        self.assertEqual(next(res), 1)

    def test_value(self):
        with self.assertRaises(ValueError) as e:
            list(fibonacci(-10))

    def test_type(self):
        with (self.assertRaises(TypeError)) as e:
            list(fibonacci("B"))

    def test_lazy(self):
        start_time = time.time()
        a = fibonacci(1000000)
        end_time = time.time() - start_time
        self.assertLess(end_time, 1)

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

## start.py

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

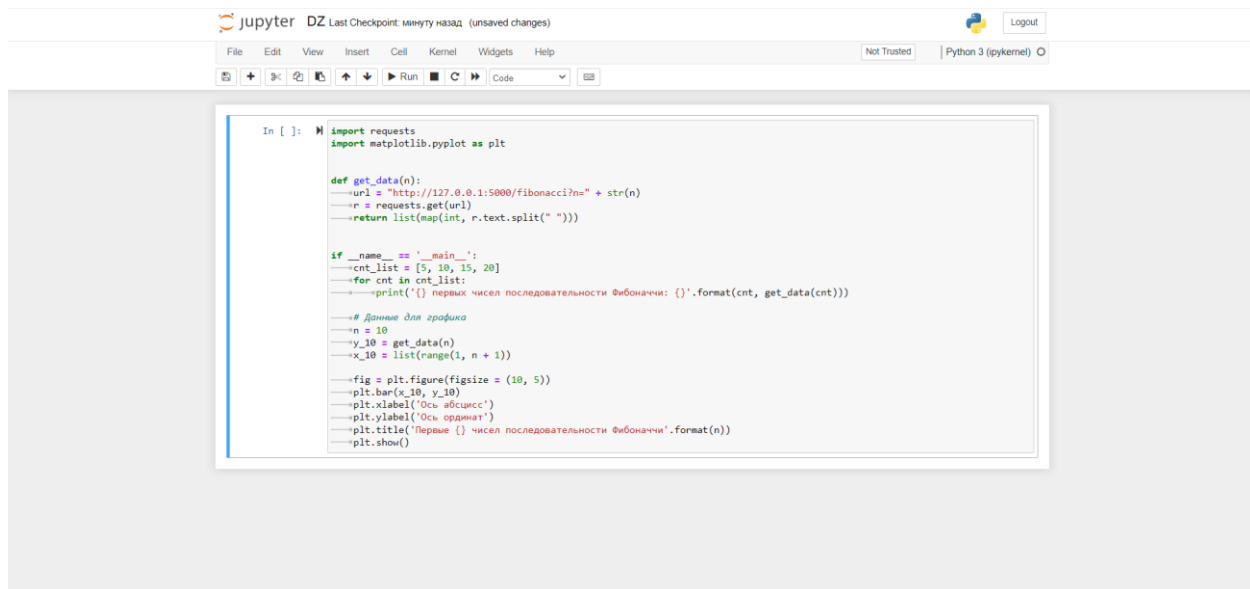
app = Flask(__name__)

@app.route('/')
def hello_world():
    return "Hello, world"

@app.route('/fibonacci')
def print_fibonacci1():
    n = request.args.get("n")
    return " ".join(map(str, fibonacci(int(n))))

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

## DZ.ipynb



The screenshot shows a Jupyter Notebook interface with a single code cell. The code in the cell is as follows:

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

def get_data(n):
    url = "http://127.0.0.1:5000/fibonacci?n=" + str(n)
    r = requests.get(url)
    return list(map(int, r.text.split(" ")))

if __name__ == '__main__':
    cnt_list = [5, 10, 15, 20]
    for cnt in cnt_list:
        print('{} первых чисел последовательности Фибоначчи: {}'.format(cnt, get_data(cnt)))

    # Данные для графика
    n = 10
    y_10 = get_data(n)
    x_10 = list(range(1, n + 1))

    fig = plt.figure(figsize = (10, 5))
    plt.bar(x_10, y_10)
    plt.xlabel('Ось абсцисс')
    plt.ylabel('Ось ординат')
    plt.title('Первые {} чисел последовательности Фибоначчи'.format(n))
    plt.show()
```

# Результаты выполнения

## fib.py

```
Run: fib (1) x
C:\Users\jmiha\AppData\Local\Programs\Python\Python39\python.exe D:/GitHub/IU5_BKIT2022/D3/fib.py
Первые 10 чисел Фибоначчи
1 1 2 3 5 8 13 21 34 55
Process finished with exit code 0
```

## tests.py

```
Run: tests
C:\Users\jmiha\AppData\Local\Programs\Python\Python39\python.exe D:/GitHub/IU5_BKIT2022/D3/tests.py
.....
Ran 5 tests in 0.000s
OK
Process finished with exit code 0
```

## DZ.ipynb

```
In [*]: import requests
import matplotlib.pyplot as plt

def get_data(n):
    url = "http://127.0.0.1:5000/fibonacci?n=" + str(n)
    r = requests.get(url)
    return list(map(int, r.text.split(" ")))

if __name__ == '__main__':
    cnt_list = [5, 10, 15, 20]
    for cnt in cnt_list:
        print('{} первых чисел последовательности Фибоначчи: {}'.format(cnt, get_data(cnt)))

    # Данные для графика
    n = 10
    y_10 = get_data(n)
    x_10 = list(range(1, n + 1))

    fig = plt.figure(figsize = (10, 5))
    plt.bar(x_10, y_10)
    plt.xlabel('Ось абсцисс')
    plt.ylabel('Ось ординат')
    plt.title('Первые {} чисел последовательности Фибоначчи'.format(n))
    plt.show()

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

```
In [ ]:
```

