** 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

**«Функциональные возможности языка Python»**

по предмету

«Разработка комплексного приложения»

Выполнил:

студент группы № ИУ5-31Б

Радченко Дмитрий

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2023 г.

**Задание.**

С использованием механизма итераторов или генераторов реализуйте с помощью концепции ленивых вычислений [одну из последовательностей OEIS.](https://ru.wikipedia.org/wiki/%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D1%8D%D0%BD%D1%86%D0%B8%D0%BA%D0%BB%D0%BE%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F_%D1%86%D0%B5%D0%BB%D0%BE%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D1%85_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9) Примером могут являться [числа Фибоначчи.](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8)

Для реализованной последовательности разработайте 3-5 модульных тестов, которые, в том числе, проверяют то, что последовательность поддерживает ленивые вычисления.

Разработайте веб-сервис с использованием фреймворка Flask, который возвращает N элементов последовательности (параметр N передается в запросе к сервису).

Создайте Jupyter-notebook, который реализует обращение к веб-сервису с использованием библиотеки [requests](https://requests.readthedocs.io/en/latest/) и визуализацию полученных от веб-сервиса данных с использованием библиотеки [matplotlib.](https://matplotlib.org/)

**Код программы:**

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>  
*import* requests  
  
  
*def* fibonacci(end\_cycle):  
 *"""  
 Числа Фибоначчи.* ***:param*** *end\_cycle: количество цифр.* ***:return****: генератор  
 """  
 if* type(end\_cycle) != int:  
 *raise* TypeError('Incorrect type of the end cycle! Waiting: Int')  
 old = 0  
 res = 1  
 *for* \_ *in* range(end\_cycle):  
 *yield* old  
 old, res = res, old + res  
  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 url = 'https:/127.0.0.1:5000/num/10'  
 r = requests.get(url)

**Libs.fibonacci\_test.py:**

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>  
*import* unittest  
*from* src.functions *import* fibonacci  
  
  
# python3 -m unittest -v libs/\*  
*class* TestFibonacci(unittest.TestCase):  
 *def* test\_numbers(self):  
 self.assertEqual(len(list(fibonacci(10))), 10)  
 self.assertEqual(list(fibonacci(10)), [0, 1, 1, 2, 3, 5, 8, 13, 21, 34])  
  
 *def* test\_iteration(self):  
 res = fibonacci(2)  
 self.assertEqual(next(res), 0)  
 self.assertEqual(next(res), 1)  
  
 *def* test\_type\_numbers(self):  
 self.assertRaises(TypeError, fibonacci('5'))

**src.get\_numbers:**

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>  
  
*import* requests  
*import* matplotlib.pyplot *as* plt  
  
  
*def* make\_url(main\_url: str, count\_numbers: int) -> str:  
 base\_url = main\_url  
 res\_url = f"{base\_url}/num/{count\_numbers}"  
 *return* res\_url  
  
  
*def* get\_data(\*, main\_url: str, count\_numbers: int) -> str:  
 res\_url = make\_url(main\_url, count\_numbers)  
 r = requests.get(res\_url)  
 *return* r.text  
  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 url = 'http://127.0.0.1:5000'  
 y\_axis = get\_data(main\_url=url, count\_numbers=int(input('Сколько чисел Фибанначу Вам необходимо? '))).split(', ')  
 print(y\_axis)  
 print(len(y\_axis))  
 x\_axis = list(range(1, len(y\_axis) + 1))  
 fig = plt.figure(figsize=(10, 5))  
 plt.bar(x\_axis, y\_axis)  
 plt.xlabel('Ось абсцисс')  
 plt.ylabel('Ось ординат')  
 plt.title('Первые {} чисел последовательности Фибоначчи'.format(len(y\_axis)))  
 plt.show()  
  
 plt.plot(x\_axis, y\_axis)  
 plt.show()

**app.py:**

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>  
  
*from* flask *import* Flask  
*from* src.functions *import* fibonacci  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route('/')  
*def* index():  
 *return* 'Hello World'  
  
  
@app.route('/num/<number>')  
*def* num(number):  
 *return* str(list(fibonacci(int(number))))[1:-1]

**Запуск сайта, main.py:**

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>  
*from* app *import* app  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 app.run()

**Результат выполнения программы:**

 