**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «БКИТ»

Отчет по домашнему заданию

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35Б |  | преподаватель каф. ИУ5 |
| Большаков Георгий |  | Нардид А.Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Описание задания**

### Задание:

1. 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)
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](https://requests.readthedocs.io/en/latest/) и визуализацию полученных от веб-сервиса данных с использованием библиотеки [matplotlib.](https://matplotlib.org/)

**Текст программы**

**fib.py**

**def** fib():  
 prev, cur = 0, 1  
 **while True**:  
 **yield** cur  
 prev, cur = cur, prev+cur  
  
**def** get\_fib\_number\_at\_pos(pos):  
 fib\_gen = fib()  
 number = 0  
 **for** i **in** range(pos):  
 number = next(fib\_gen)  
 **return** number  
  
**def** get\_fib\_seq(n):  
 fib\_gen = fib()  
 arr = []  
 **for** i **in** range(n):  
 arr.append(next(fib\_gen))  
 **return** arr

**test.py**

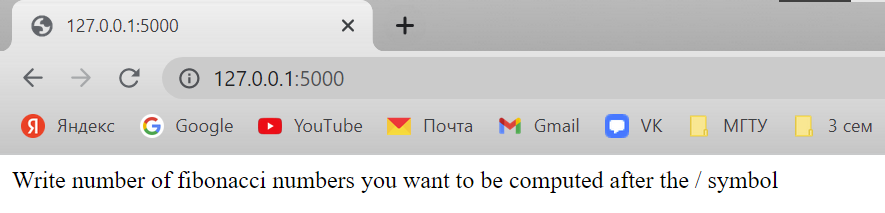
**import** unittest  
**from** fib **import** get\_fib\_number\_at\_pos, get\_fib\_seq, fib  
  
  
**class** TEST(unittest.TestCase):  
 **def** test\_Pos9(self):  
 self.assertEqual(34, get\_fib\_number\_at\_pos(9))  
  
 **def** test\_SeqOf5(self):  
 self.assertEqual([1, 1, 2, 3, 5], get\_fib\_seq(5))  
  
 **def** test\_LenSeqOf10(self):  
 self.assertEqual(10, len(get\_fib\_seq(10)))  
  
**if** \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

**main.py**

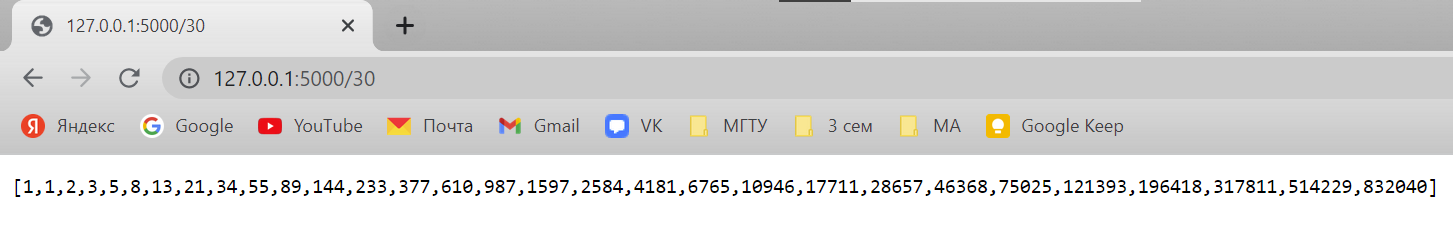
**from** fib **import** fib  
  
**from** flask **import** Flask  
  
app = Flask(\_\_name\_\_)  
  
@app.route("/")  
**def** hello\_world():  
 **return** "<p>Write number of fibonacci numbers you want to be computed after the / symbol</p>"  
  
@app.route("/<int:n>")  
**def** fibonacci\_number(n):  
 fib\_gen = fib()  
 fib\_numbers = []  
 **for** i **in** range(n):  
 fib\_numbers.append(next(fib\_gen))  
 **return** fib\_numbers

**Экранные формы**

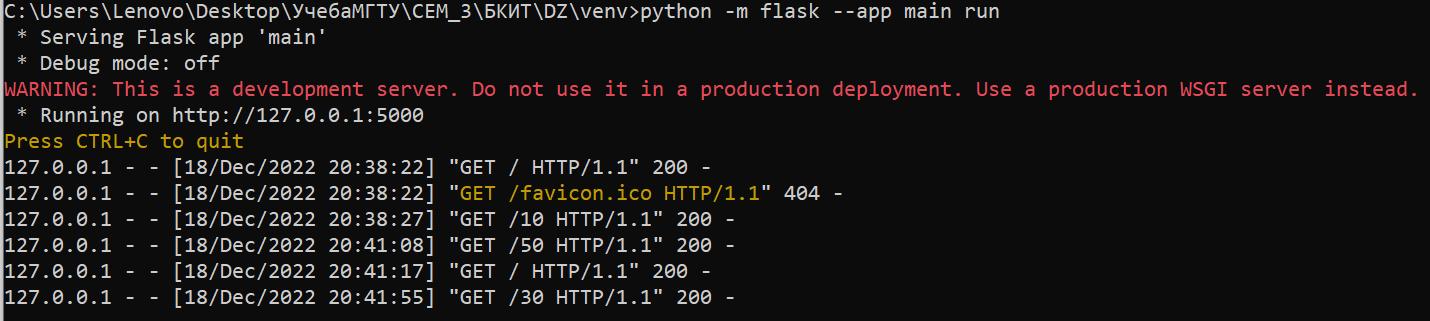
**Главное окно запущенного веб-сервиса:**

****

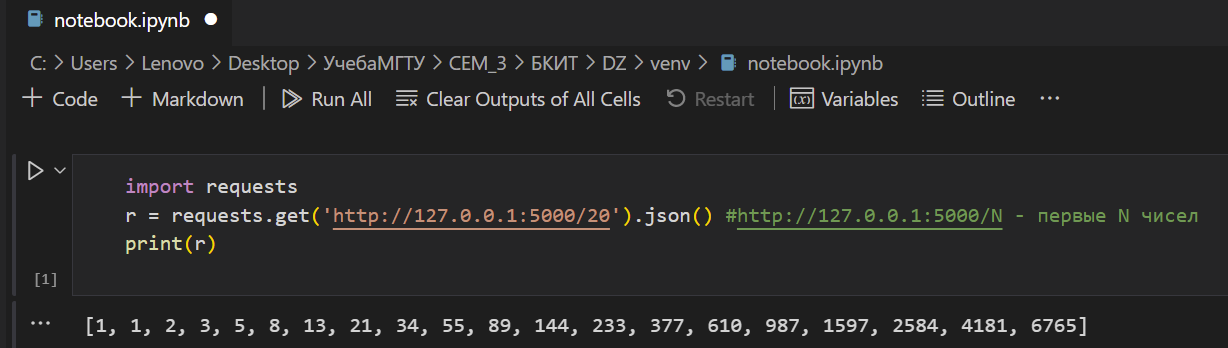
**Окно веб-сервиса с выданными на запрос 30 первыми числами Фиббоначи:**

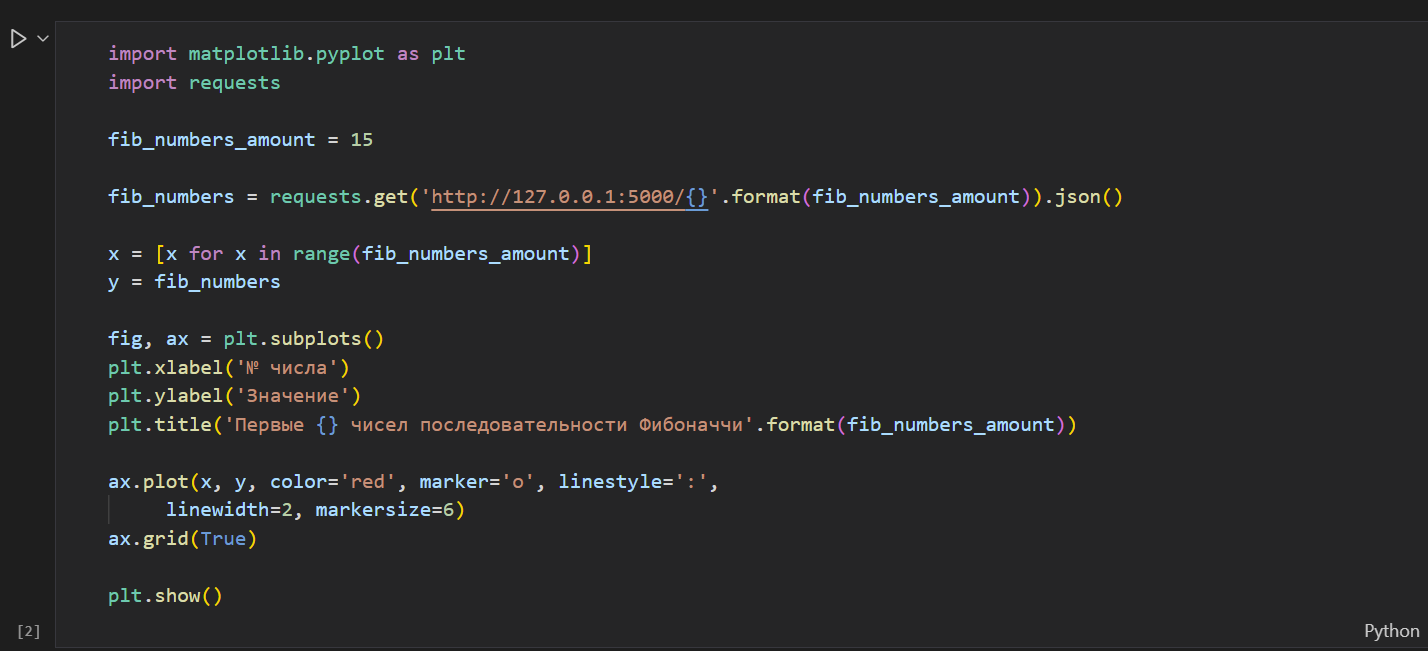
****

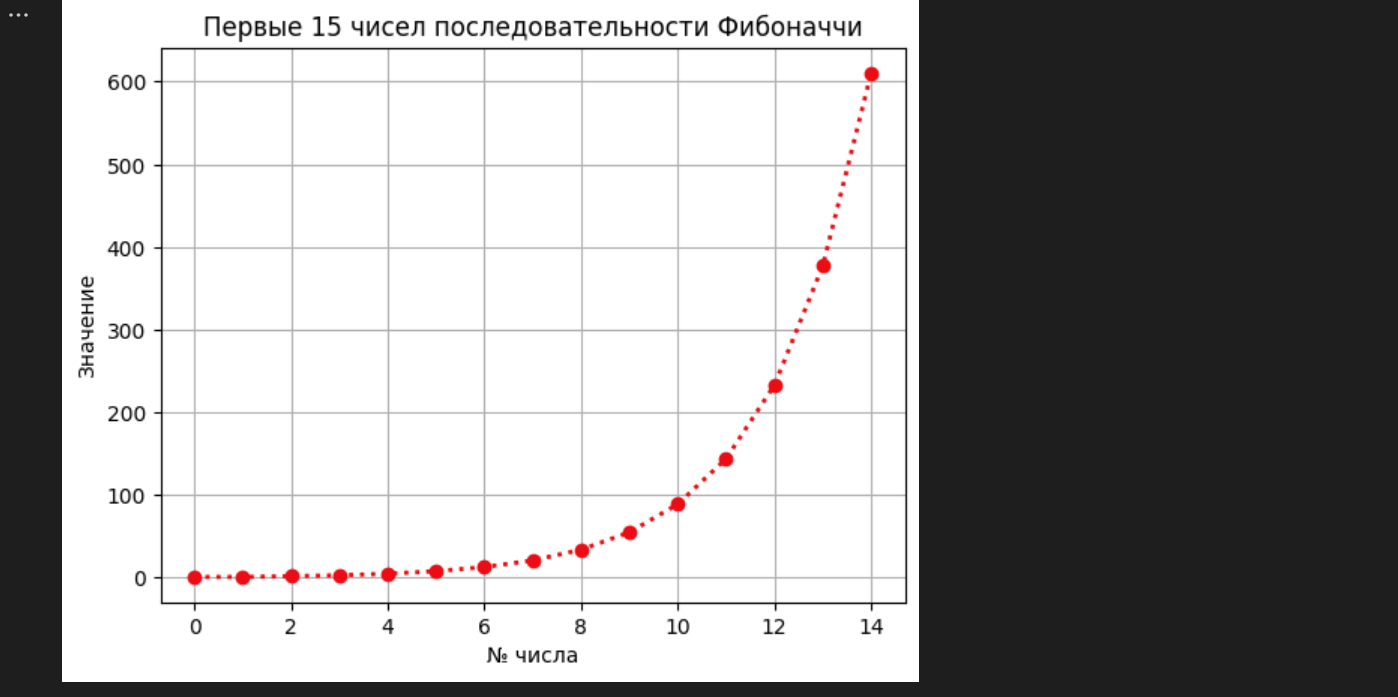
**Запуск веб-сервиса из терминала:**

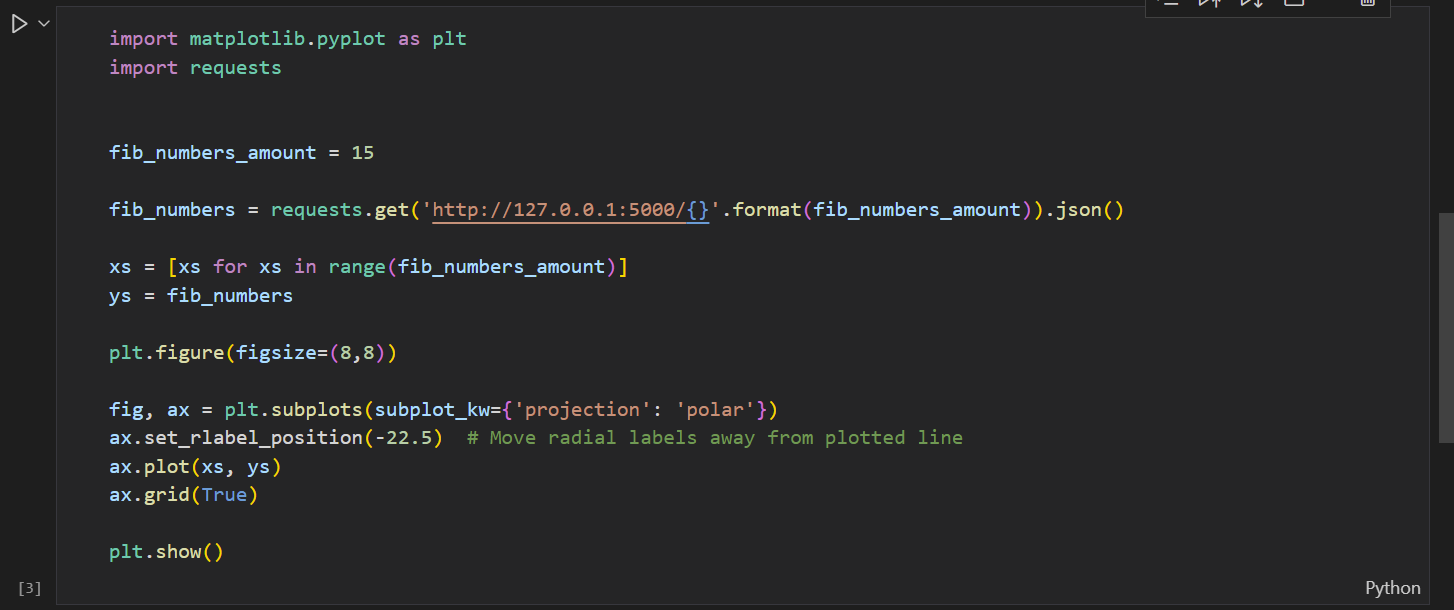
****

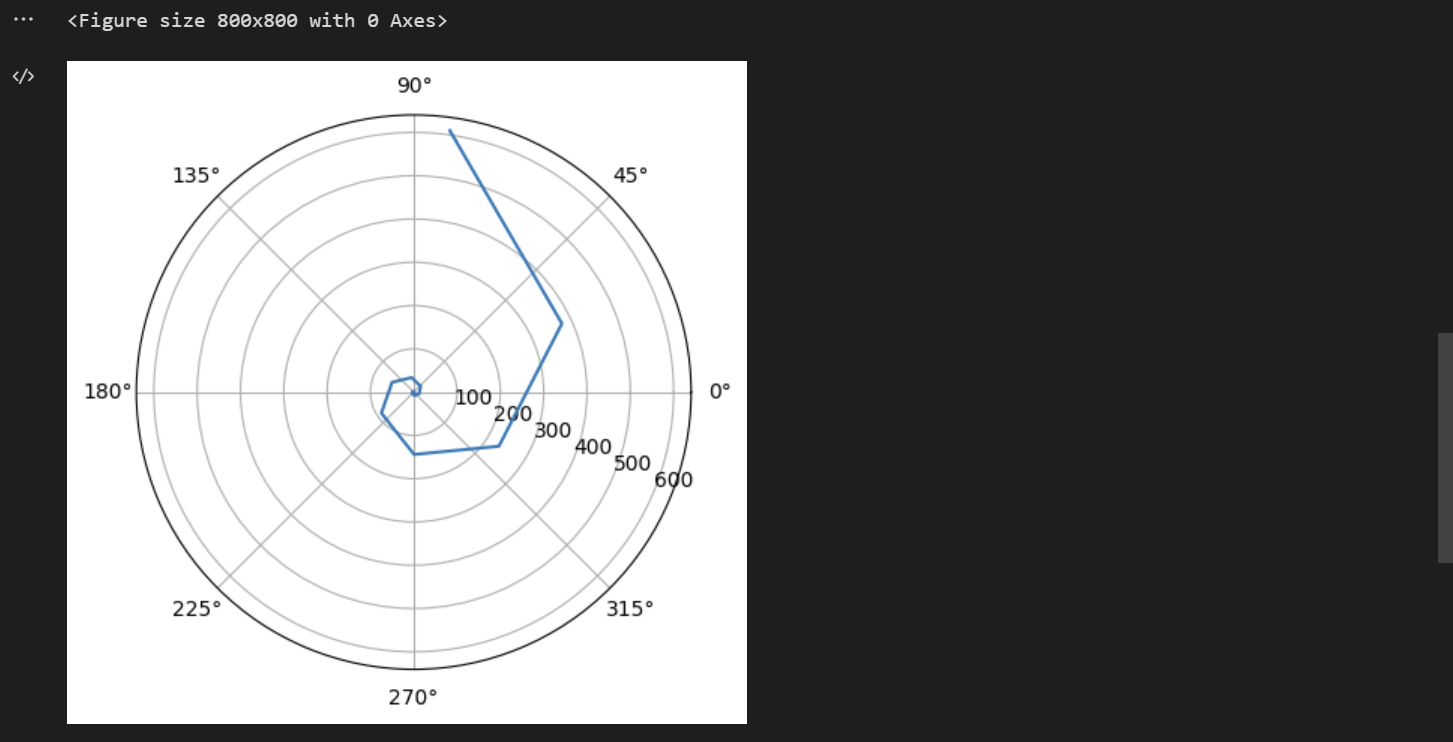
**Созданный Jupiter-notebook:**

****

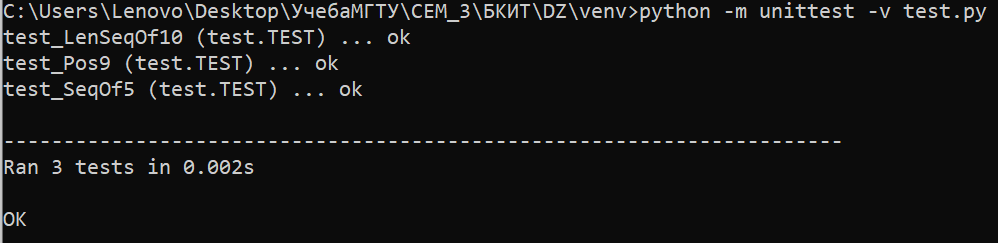
****

****

****

****

**TDD-тестирование используемого генератора:**

****