Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию

Выполнил:
студент группы ИУ5-35Б
Коныгина Дарья
Подпись и дата:

Проверил:
Преподаватель каф. ИУ5
Нардид А.Н.
Подпись и дата:

Москва, 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.

Текст программы:

1. Файл main.py с реализацией вычислений последовательности Рекамана:

```
# последовательность Рекамана

def rec():
    n = 0
    prev, cur = 0, 0
    l = []
    while True:
        if n>0 and (prev - n)>0 and (prev - n) not in l:
            cur = prev - n
            yield cur
        else:
            cur = prev + n
            yield cur
        l.append(cur)
        prev = cur
        n += 1

if __name__ == '__main__':
    rec_gen = rec()
    for i in range(10):
        j = next(rec_gen)
        print(j)
```

2. Файл num.py

```
from flask import Flask
from main import rec

app = Flask(__name__)
```

```
@app.route("/")
def hello_world():
    return "<p>Returning the Recamán's sequence!</p>"

@app.route('/num/<int:cnt>')
def get_rec(cnt):
    print(cnt)
    rec_gen = rec()
    res = [next(rec_gen) for _ in range(cnt)]
    return str(res)

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

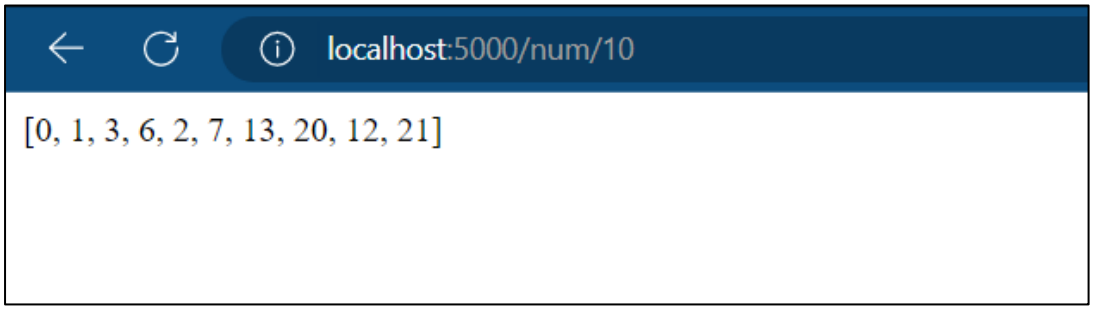
3. Файл test_rec.py

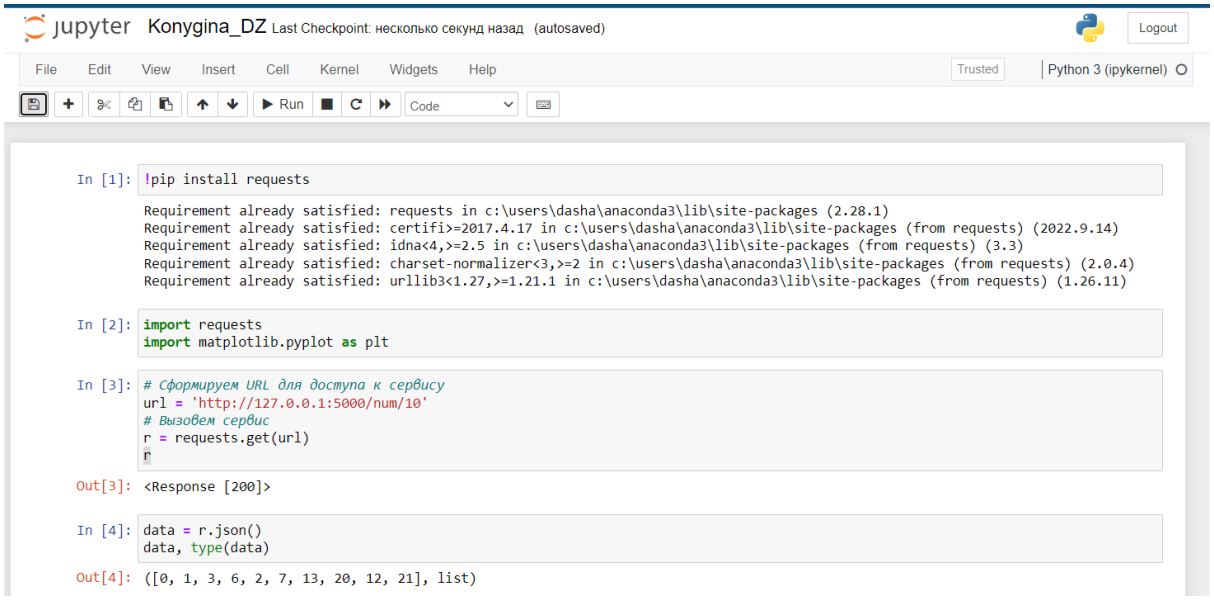
```
import pytest
import unittest
from main import rec
#TDD

def test_1():
    rec_gen = rec()
    l = []
    for i in range(10):
        l.append(next(rec_gen))
    assert [0, 1, 3, 6, 2, 7, 13, 20, 12, 21] == l

# должна выдаваться ошибка!!!!!!!!!!
def test_2():
    rec_gen = rec()
    assert 1 == rec_gen(1)

def test_3():
    rec_gen = rec()
    l = []
    for i in range(11):
        l.append(next(rec_gen))
    assert 11 == l[-1]


if __name__ == "__main__":

    unittest.main()

        pass
```

Экранные формы с примерами выполнения программы:

```
============================= test session starts ==============================
collecting ... collected 3 items

test_rec.py::test_1 PASSED                                               [ 33%]
test_rec.py::test_2 FAILED                                               [ 66%]
test_rec.py:13 (test_2)
def test_2():
        rec_gen = rec()
>       assert 1 == rec_gen(1)
E       TypeError: 'generator' object is not callable

test_rec.py:16: TypeError

test_rec.py::test_3 PASSED                                               [100%]

======================== 1 failed, 2 passed in 0.14s =========================

Process finished with exit code 1
```

Веб-сервис с использованием фреймворка Flask:





In [5]:
```
# Функции для выполнения запроса к сервису

def make_url(cnt):
    base_url = 'http://127.0.0.1:5000/num/'
    res = base_url + str(cnt)
    return res

def get_data(cnt):
    url = make_url(cnt)
    r = requests.get(url)
    return r.json()
```

In [6]:
```
cnt_list = [5, 10, 15, 20]
for cnt in cnt_list:
    print('{} первых чисел последовательности Рекамена: {}'.format(cnt, get_data(cnt)))
```

```
5 первых чисел последовательности Рекамена: [0, 1, 3, 6, 2]
10 первых чисел последовательности Рекамена: [0, 1, 3, 6, 2, 7, 13, 20, 12, 21]
15 первых чисел последовательности Рекамена: [0, 1, 3, 6, 2, 7, 13, 20, 12, 21, 11, 22, 10, 23, 9]
20 первых чисел последовательности Рекамена: [0, 1, 3, 6, 2, 7, 13, 20, 12, 21, 11, 22, 10, 23, 9, 24, 8, 25, 43, 62]
```

jupyter Konygina_DZ Last Checkpoint: несколько секунд назад (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

Построение графика

In [7]:
```
# Данные для графика
y_10 = get_data(10)
x_10 = list(range(1, len(y_10)+1))
```

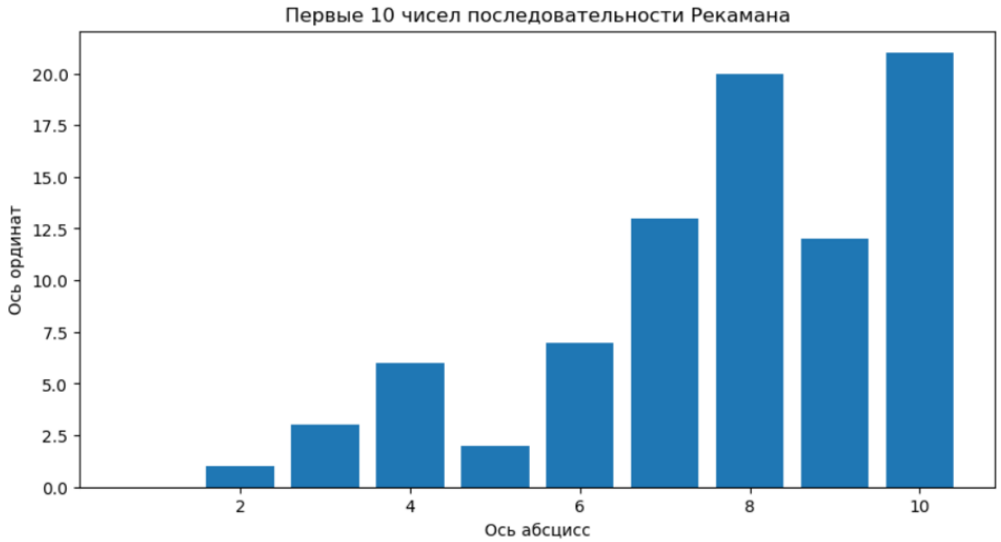
In [8]:
```
# Ось абсцисс - порядковые номера чисел
x_10
```

Out[8]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

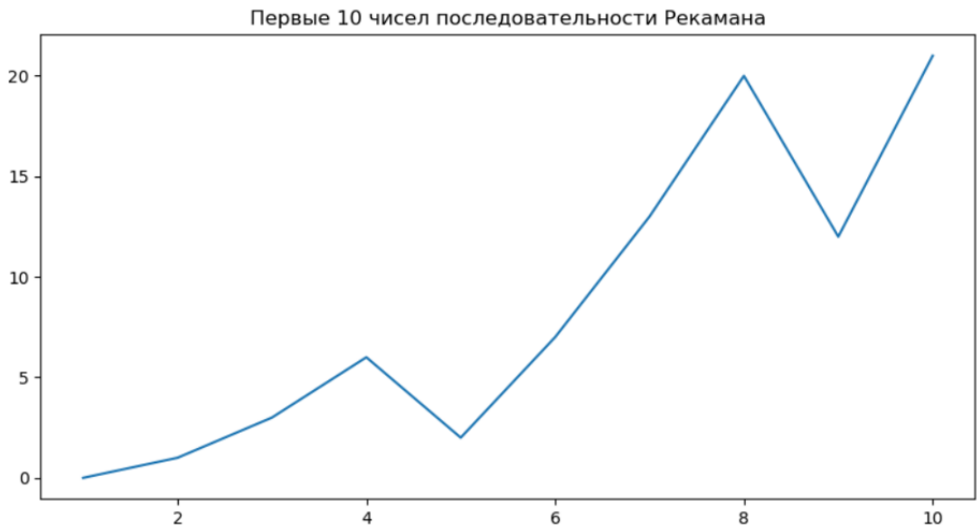
In [9]:
```
# Ось ординат - числа
y_10
```

Out[9]: [0, 1, 3, 6, 2, 7, 13, 20, 12, 21]

In [10]:
```
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 Рекамана'.format(len(y_10)))
plt.show()
```



In [12]:
```
fig = plt.figure(figsize = (10, 5))
plt.plot(x_10, y_10)
plt.title('Первые {} чисел последовательности Рекамана'.format(len(y_10)))
plt.show()
```


Первые 10 чисел последовательности Рекамана
20
15
10
5
0
2
4
6
8
10

In [ ]: