

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Работа с IPython и Jupyter Notebook»**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**дисциплины**  
**«Технологии распознавания образов»**

Выполнил:  
Борсуков Владислав Олнгович  
2 курс, группа ПИЖ-б-о-21-1,  
011.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

**Цель работы:** исследовать базовые возможности интерактивных оболочек IPython и Jupyter Notebook для языка программирования Python.

Проработка примеров из лабораторной работы:

```
In [1]: 3 + 2
Out[1]: 5
```

---

```
In [2]: a = 5
        b = 7
        print(a + b)
12
```

---

```
In [4]: n = 7
        for i in range(n):
            print(i*10)
0
10
20
30
40
50
60
```

---

```
In [5]: i = 0
        while True:
            i += 1
            if i > 5:
                break
            print("Test while")
Test while
Test while
Test while
Test while
Test while
```

Рисунок 1 – Проработка примеров

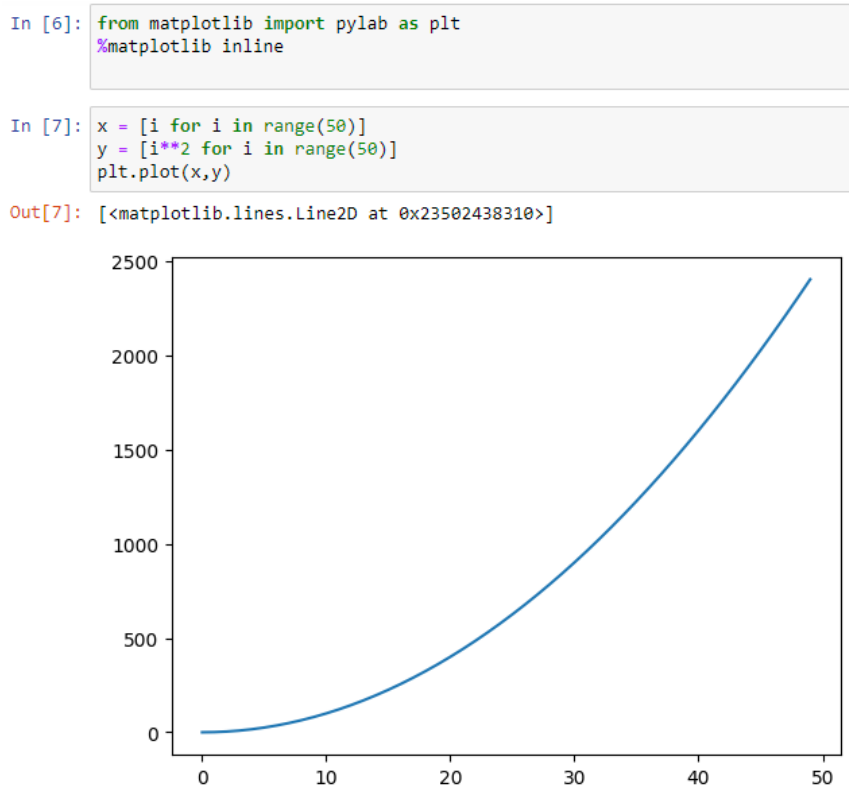


Рисунок 2 – Проработка примеров

```
In [9]: %lsmagic

Out[9]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark %cd %clear %cls %colors %conda %config %connect_info %copy %ddir %debug %dhist %dirs %doctest_mode %echo %ed %edit %env %gui %hist %history %killbgscripts %ldir %less %load %load_ext %loadpy %logoff %logon %logstart %logstate %logstop %ls %lsmagic %macro %magic %matplotlib %mkdir %more %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip %popd %pprint %precision %prun %psource %pushd %pwd %pycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext %ren %rep %rerun %reset %reset_selective %rmdir %run %save %sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos %xdel %xmode

Available cell magics:
%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javascript %%js %%latex %%markdown %%perl %%prun %ppypy %%python %%python2 %%python3 %%ruby %script %%sh %%svg %sx %%system %%time %%timeit %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

```
In [10]: %%time
import time
for i in range(50):
    time.sleep(0.1)

CPU times: total: 31.2 ms
Wall time: 5.14 s
```

```
In [11]: %timeit x = [(i**10) for i in range(10)]

1.45 µs ± 268 ns per loop (mean ± std. dev. of 7 runs, 1,000,000 loops each)
```

## Рисунок 3 – Проработка примеров

### Задания из файла lab\_3.1:

#### Задание:

- 1) Определите число `ticket_number` — шестизначный номер билета;
- 2) Напишите код, который по шестизначному номеру `ticket_number` билетика проверяет, является ли он счастливым;
- 3) Если номер счастливый, выведите строку `Yes`, иначе — `No`.

#### Пример 1:

*Input:* 123456

*Output:* No

#### Пример 2:

*Input:* 123042

*Output:* Yes

```
In [1]: ticket_number = 123123

In [2]: sum_1 = ticket_number // 1000
sum_2 = ticket_number % 1000
if sum(map(int, str(sum_2))) == sum(map(int, str(sum_1))):
    print("Yes")
else:
    print("No")
```

Yes

## Рисунок 4 – Условие и решение задания №1

**Задание:**

- 1) Определите строку `password` — придуманный вами пароль;
- 2) Напишите код, который по паролю `password` проверяет, является ли он надёжным;
- 3) Если пароль надёжный, выведите строку `strong`, иначе — `weak`.

Пусть имя пользователя -- Андрей.

**Пример 1:**

*Input:* Aandrei123

*Output:* weak

**Пример 2:**

*Input:* an12dRei

*Output:* strong

```
In [10]: password = "4Marta"
```

```
In [12]: name = input("Введите имя: ")
if (name.lower() in password.lower() or password.islower() or password.isupper() or
    password.isdigit() or len(set(password)) <= 4 or password.isalpha()):
    print("Weak")
else:
    print("Strong")
```

Введите имя: marta  
Weak

## Рисунок 5 – Условие и решение задания №2

### Числа Фибоначчи

Как известно, [числа Фибоначчи](#) — это последовательность чисел, каждое из которых равно сумме двух предыдущих (первые два числа равны 1):  
1, 1, 2, 3, 5, 8, 13, ...

**Задание:**

- 1) Определите число `amount` — количество чисел Фибоначчи, которые надо вывести;
- 2) Напишите код, который выводит первые `amount` чисел Фибоначчи.

**Пример 1:**

*Input:* 3

*Output:* 1 1 2

**Пример 2:**

*Input:* 10

*Output:* 1 1 2 3 5 8 13 21 34 55

```
In [1]: amount = 15
```

```
In [4]: n = 0
tm = 1
for i in range(amount):
    s = n + tm
    n = tm
    print(tm, end = " ")
    tm = s
```

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610

## Рисунок 6 – Условие и решение задания №3

```

In [10]: import csv
import statistics
from matplotlib import pylab as plt
%matplotlib inline

with open('topSubscribed.csv', 'r', newline='', encoding='utf-8') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    sub = []
    year_start = []
    next(reader)
    for k in reader:
        sub.append(int(k[2]))
        year_start.append(int(k[6]))
sr_year = sum(year_start)/len(year_start)
sr_sub = sum(sub)/len(sub)
print(f"Среднее значение подписчиков: {sr_sub}")
print(f"Среднее значение года регистрации канала: {sr_year}\n")
|
otkl_sub = statistics.stdev(sub)
otkl_year = statistics.stdev(year_start)
print(f"Стандартное отклонение количества подписчиков: {otkl_sub}")
print(f"Стандартное отклонение года регистрации канала: {otkl_year}\n")

sum_year = sum(year_start)
sum_sub = sum(sub)
sm = 0
ssy = 0
for k, elem in enumerate(year_start):
    sm += elem * sub[k]
    ssy += elem**2
a = len(year_start)
b = (a * sm - sum_year * sum_sub) / (a * ssy - sum_year ** 2)
c = (sum_sub - b * sum_year) / a
result = []
for elem in year_start:
    result.append(b * elem + c)
print(f"Уравнение линейной зависимости: y = {b}x + {c}")
print(f"График функции методом наименьших квадратов: ")
plt.plot(year_start,result)

```

Рисунок 7.1 – Решение задачи №4

Среднее значение подписчиков: 21581400.0  
Среднее значение года регистрации канала: 2012.594

Стандартное отклонение количества подписчиков: 16625563.551453028  
Стандартное отклонение года регистрации канала: 4.110238199673841

Уравнение линейной зависимости:  $y = -546718.133449435x + 1121903035.0715322$   
График функции методом наименьших квадратов:

Out[10]: [<matplotlib.lines.Line2D at 0x29920be0760>]

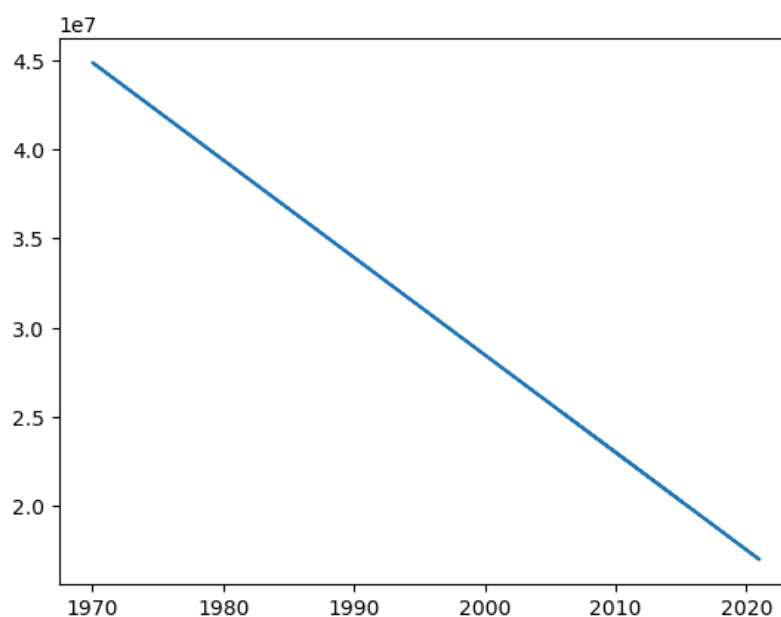


Рисунок 7.2 – Решение задачи №4

Решение задачи, поставленной в методических указаниях:

Найти производную функции  $y = \sqrt{x^2 + 4} \cdot \ln(\sin x)$

```
In [7]: from sympy import *
import numpy as np
import math

x = Symbol('x')
y = sqrt(x**2 + 4) * log(sin(x))

result = y.diff(x)
print(f"y' = {result}")

y' = x*log(sin(x))/sqrt(x**2 + 4) + sqrt(x**2 + 4)*cos(x)/sin(x)
```

Рисунок 8 – Программа для решения выбранной задачи

**Выводы:** исследовали базовые возможности интерактивных оболочек IPython и Jupyter Notebook для языка программирования Python

## Контрольные вопросы

### 1. Как осуществляется запуск Jupyter notebook?

При помощи команды `jupyter-notebook` в терминале IDE

### 2. Какие существуют типы ячеек в Jupyter notebook?

Markdown и Code

### 3. Как осуществляется работа с ячейками в Jupyter notebook?

Ячейки в `jupyter notebook` можно создавать удалять и запускать их работу при помощи комбинаций клавиш

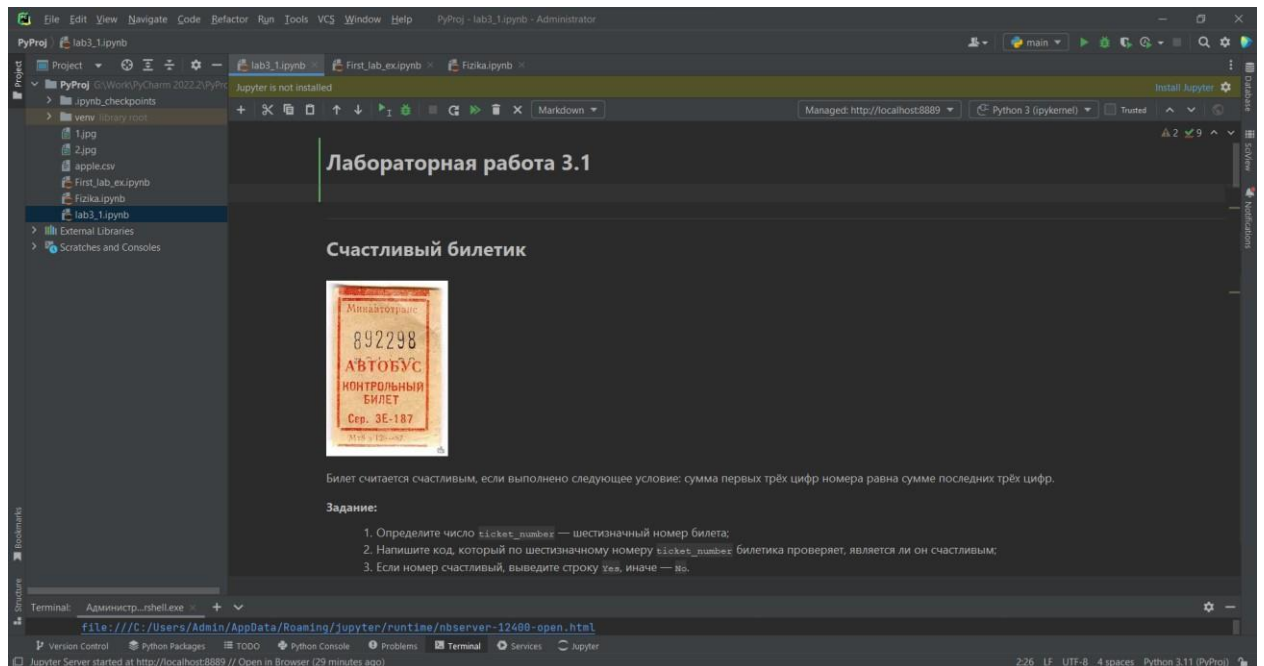
### 4. Что такое "магические" команды Jupyter notebook? Какие "магические" команды Вы знаете?

```
Out[9]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark %cd %clear %cls %colors %conda %config %connect_info %copy %ddir %debug %dhist %dirs %doctest_mode %echo %ed %edit %env %gui %hist %history %killbgscripts %ldir %less %load %load_ext %loadpy %logoff %logon %logstart %logstate %logstop %ls %lsmagic %macro %magic %matplotlib %mkdir %more %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip %popd %pprint %precision %prun %pssearch %psource %pushd %pwd %pycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext %ren %rep %rerun %reset %reset_selective %rmdir %run %save %sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos %xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javascript %%js %%latex %%markdown %%perl %%prun %%pypy %%python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

5. Самостоятельно изучите работу с Jupyter notebook и IDE PyCharm и VisualStudio Code.



6. Приведите основные этапы работы с Jupyter notebook в IDE PyCharm и Visual Studio Code.

Pip install notebook, после чего можно запустить при помощи команды jupyter-notebook