

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:  
Мизин Глеб Егорович  
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

```
In [6]: from matplotlib import pylab as plt
        %matplotlib inline

In [7]: x = [i for i in range(50)]
        y = [i**2 for i in range(50)]
        plt.plot(x,y)

Out[7]: [<matplotlib.lines.Line2D at 0x23502438310>]
```

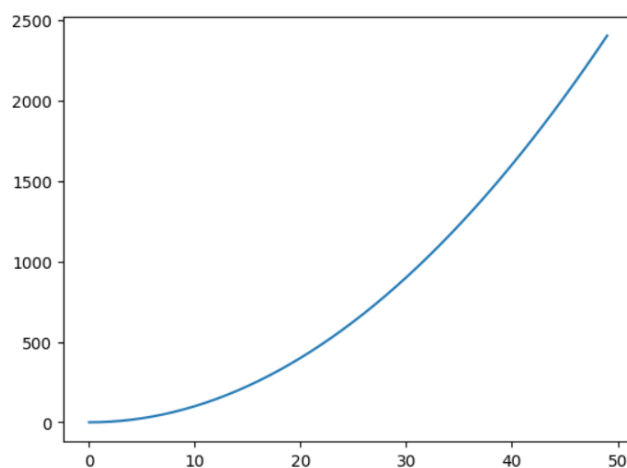


Рисунок 2 – Проработка примеров

```
In [9]: %lsmagic

Out[9]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark %cd %clear %cls %colors %conda %config %co
nnect_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 %matpl
otlib %mkdir %more %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip %popd %pprint %precisio
n %prun %psearch %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 %unl
oad_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.

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 прикрепленного файла lab\_3.1:

#### Задание:

1. Определите число `ticket_number` — шестизначный номер билета;
2. Напишите код, который по шестизначному номеру `ticket_number` билетики 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 = 123321

In [2]: num_1 = ticket_number // 1000
num_2 = ticket_number % 1000
if sum(map(int, str(num_2))) == sum(map(int, str(num_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 [3]: `pw = "Andrew123"`

```
In [4]: name = input("Enter your name:")
if name.lower() == pw.lower() or pw.islower() or pw.isupper() or pw.isdigit() or len(set(pw)) <= 4 or pw.isalpha():
    print("Weak")
else:
    print("Strong")
```

Enter your name:Gleb  
Strong

## 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 [30]: `amount = 10`

```
In [31]: 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

## Рисунок 6 – Условие и решение задания №3

```

In [23]: import csv
import statistics
from matplotlib import pylab as plt
%matplotlib inline

with open('world_prices.csv', 'r', newline='', encoding='utf-8') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    Year = []
    jp = []
    next(reader)
    for k in reader:
        if k[3] != "N/A":
            Year.append(float(k[3]))
            jp.append(float(k[8]))
    sr_year = sum(Year)/len(Year)
    sr_jp = sum(jp)/len(jp)
    print(f"Среднее значение года скидки: {sr_year}")
    print(f"Среднее значение продаж в японии на момент скидки {sr_jp}")

    stat_year = statistics.stdev(Year)
    stat_jp = statistics.stdev(jp)
    print(f"Стандартное отклонение года скидки: {stat_year}")
    print(f"Стандартное отклонение продаж в японии на момент скидки {stat_jp}")
    s_year = sum(Year)
    s_jp = sum(jp)
    sm = 0
    ssy = 0
    for k, elem in enumerate(Year):
        sm += elem * jp[k]
        ssy += elem**2
    a = len(Year)
    b = (a * sm - s_year * s_jp) / (a * ssy - s_year ** 2)
    c = (s_jp - b * s_year) / a
    calc = []
    for elem in Year:
        calc.append(b * elem + c)
    print(f"Уравнение линейной зависимости: y = {b}x + {c}")
    print(f"График функции методом наименьших квадратов: ")

plt.plot(Year, calc)

```

Рисунок 7.1 – Решение задачи №4

Среднее значение года скидки: 2005.2577534532186  
 Среднее значение продаж в японии на момент скидки 0.2311076361740921  
 Стандартное отклонение года скидки: 6.760001701313692  
 Стандартное отклонение продаж в японии на момент скидки 0.6042117540477622  
 Уравнение линейной зависимости:  $y = -0.019219000315882185x + 38.770157033216705$   
 График функции методом наименьших квадратов:

Out[23]: [

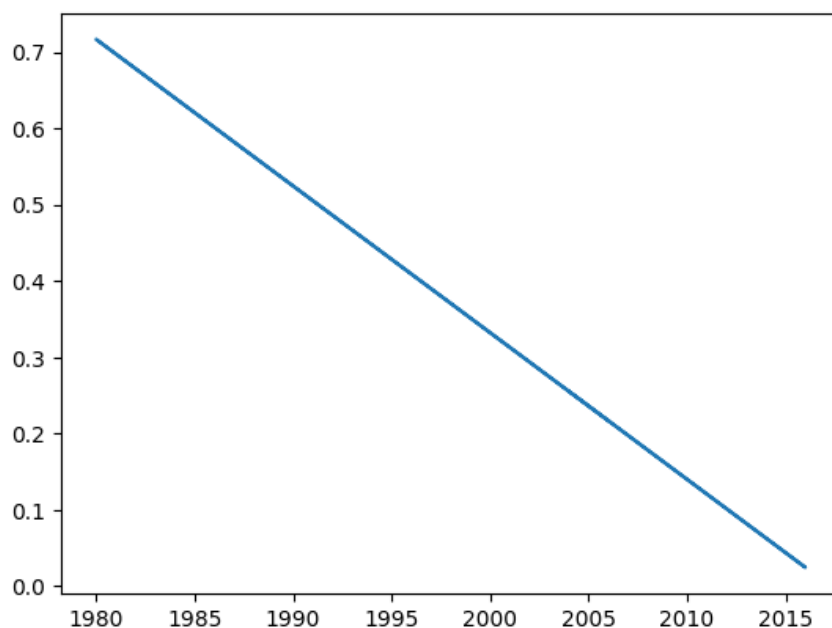


Рисунок 7.2 – Решение задачи №4

## Решение задачи, поставленной в методических указаниях:

Условие задачи: Найти давление воздуха в откачиваемом сосуде как функцию времени откачки  $t$ . Объем сосуда  $V = 100$  л. Процесс считать изотермическим и скорость откачки независимой от давления и равной  $C = 0.01$  л/с. Скоростью откачки называют объем газа, откачиваемый за единицу времени, причем этот объем измеряется при давлении газа в данный момент времени.

Аналитическое решение поставленной задачи: Рассмотрим изотермический процесс при изменении объема на  $dV$  и давления на  $dp$ , тогда  $pV = (p + dp)(V + dV)$ , или  $pV = pV + pdV + Vdp + dpdV$ ; пренебрегая последним слагаемым, мы получаем дифференциальное уравнение первого порядка с разделяющимися переменными  $dp/p = -dV/V$ . Учитывая, что изменение объема  $dV = Cdt$ , уравнение можно проинтегрировать:  $\ln(p) - \ln(p_0) = -\frac{C}{V}t$  или  $\frac{p}{p_0} = \exp(-\frac{C}{V}t)$ . Полученная формула является решением задачи.

## Рисунок 8 – Условие и аналитическое решение выбранной задачи

```
In [1]: import matplotlib.pyplot as plt
import math
N=20; C=0.01; tmax=50.0; dt=tmax/N; V=0.1;
dV=0.0005; qV=V/(V+dV); b1=C/V; dt1=V*(1.0/qV-1.0)/C
p0=1.0
print (" q =",qV," b1=",b1," \Delta t=",dt1)
t=[]; p1=[]; p2=[]
t.append(0); p1.append(p0); p2.append(p0)
for i in range(1,N):
    t1=i*dt; t.append(t1)
    p1.append(math.exp(-b1*t1))
    p2.append(qV**p1/dt1)
plt.plot(t,p1,'k-')
plt.plot(t,p2,'ko:')
plt.xlabel('$t$',fontsize=14)
plt.ylabel('$\eta_1, \eta_2$',fontsize=14)
plt.show()

q = 0.9950248756218906 b1= 0.09999999999999999 \Delta t= 0.04999999999998934
```

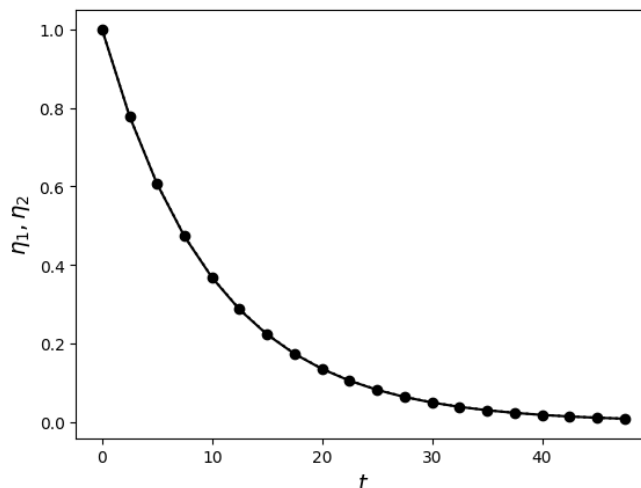


Рисунок 9 – Программа для решения выбранной задачи

**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

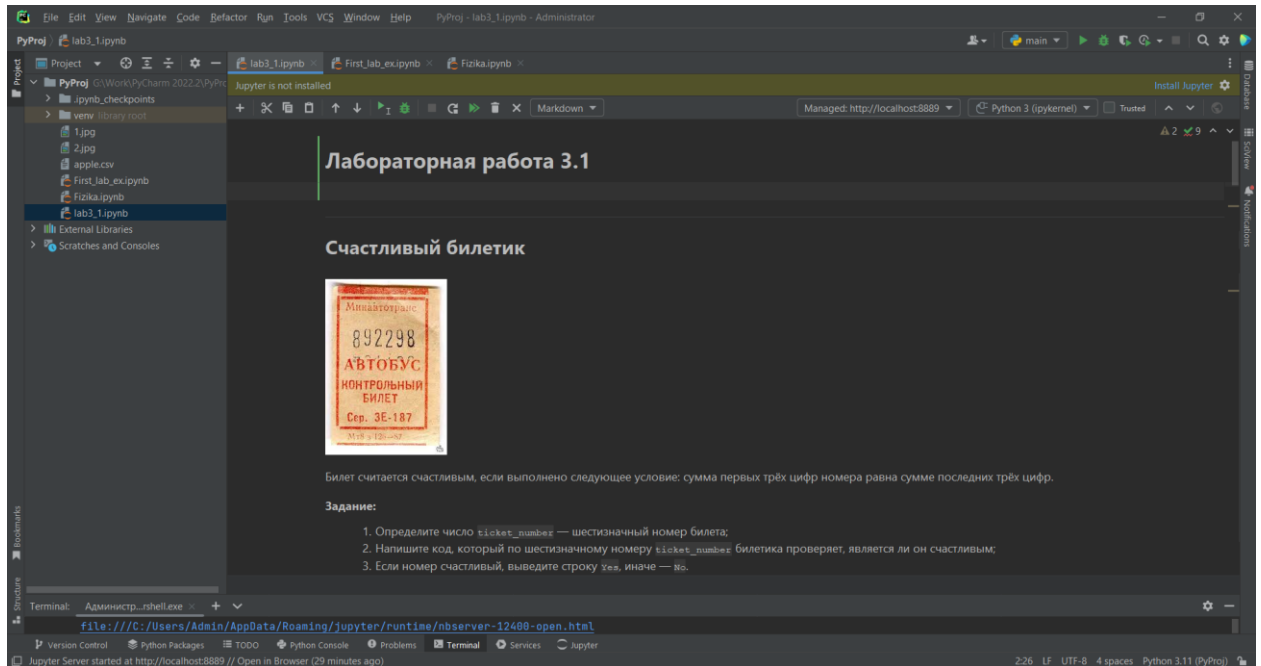
### 4. Что такое "магические" команды Jupyter notebook? Какие "магические" команды Вы знаете?

```
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 %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 Visual Studio 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`