

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №1 по дисциплине:
Технологии распознавания образов**

Выполнила:

студент группы ПИЖ-б-о-20-1

Лазарева Дарья Олеговна

Проверил:

доцент кафедры инфокоммуникаций

Романкин Р.А.

Ставрополь, 2022 г.

1. Выполнение примеров из методических указаний

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

```
Out[1]: 5
```

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

```
12
```

```
In [3]: n = 7  
for i in range(n):  
    print(i*10)
```

```
0  
10  
20  
30  
40  
50  
60
```

```
In [4]: i = 0  
while True:  
    i += 1  
    if i > 5:  
        break  
    print("Test while")
```

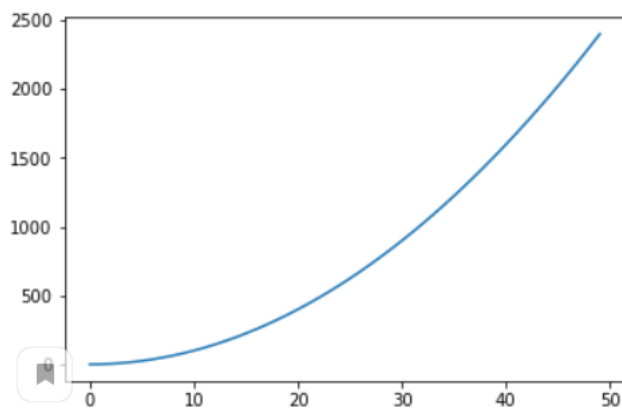
```
Test while  
Test while  
Test while  
Test while  
Test while
```

2. Рисование графиков

```
In [5]: from matplotlib import pylab as plt  
%matplotlib inline
```

```
In [6]: x = [i for i in range(50)]  
y = [i**2 for i in range(50)]  
plt.plot(x,y)
```

```
Out[6]: [<matplotlib.lines.Line2D at 0x216a4750850>]
```



3. Выполнение команды %lsmagic

```
In [7]: %lsmagic
```

```
Out[7]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bo
okmark %cd %clear %cls %colors %conda %config %connect_info %co
py %ddir %debug %dhist %dirs %doctest_mode %echo %ed %edit %en
v %gui %hist %history %killbgscripts %ldir %less %load %load_ex
t %loadpy %logoff %logon %logstart %logstate %logstop %ls %lsma
gic %macro %magic %matplotlib %mkdir %more %notebook %page %pas
tebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %pip %popd %pprin
t %precision %prun %psearch %psource %pushd %pwd %pycat %pylab
%qtconsole %quickref %recall %rehashx %reload_ext %ren %rep %rer
un %reset %reset_selective %rmdir %run %save %sc %set_env %stor
e %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 %%pyth
on %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system
%%time %%timeit %%writefile
```

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

4. Выполнение команды %%time и %timeit

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

Wall time: 5.28 s

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

2.8 μ s \pm 329 ns per loop (mean \pm std. dev. of 7 runs, 100000 loops each)

Задание 1. Счастливый билетик

```
In [1]: ticket_number = list(map(int, (input("Введите число: "))))
```

Введите число: 456375

```
In [2]: if sum(ticket_number[:3]) == sum(ticket_number[3:]):
print('Счастливый')
else:
    print('Обычный')
```

Счастливый

Задание 2. Пароль

```
In [6]: password = 'Qwerty123'
```

11111

```
In [7]: k = password.lower().count("andrei")
if k == 0:
    uq = set(password)
    lower_flag = False
    upper_flag = False
    digit_flag = False
    for char in uq:
        if char.islower():
            lower_flag = True
        if char.isupper():
            upper_flag = True
        if char.isdigit():
            digit_flag = True
    if len(uq) >= 4 and lower_flag and upper_flag and digit_flag:
        print('strong')
    else:
        print('weak')
else:
    print('weak')
```

weak

Задание 3. Числа Фибоначчи

```
In [9]: f1 = f2 = 1
amount = int(input())
```

3

```
In [10]: print(f1, f2, end=' ')
for i in range(2, amount):
    f1, f2 = f2, f1 + f2
    print(f2, end=' ')
```

1 1 2

Задание 4. Время исследований

```
In [48]: import csv
from math import sqrt

with open('cancer_b.csv', 'r', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=',')
    radius_pol = []
    area_pol = []
    for row in data:
        radius_pol.append(int(row[3]))
        area_pol.append(row[6])

-----
ValueError                                Traceback (most recent call last)
<ipython-input-48-abad44938ad2> in <module>
      8     area_pol = []
      9     for row in data:
--> 10         radius_pol.append(int(row[3]))
     11         area_pol.append(row[6])

ValueError: invalid literal for int() with base 10: 'Texture (mean)'
```

Следующий этап - нахождение среднего значения коэффициентов из списка:

```
In [47]: mean_radius = sum(radius_pol) / len(radius_pol)
mean_area = sum(area_pol) / len(area_pol)
print(f"Среднее значение радиуса: {mean_radius} ")
print(f"Среднее значение площади: {mean_area} ")

-----
TypeError                                Traceback (most recent call last)
<ipython-input-47-d6b7fb425c99> in <module>
----> 1 mean_radius = sum(radius_pol) / len(radius_pol)
      2 mean_area = sum(area_pol) / len(area_pol)
      3 print(f"Среднее значение радиуса: {mean_radius} ")
      4 print(f"Среднее значение площади: {mean_area} ")

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Контрольные вопросы:

1. Как осуществляется запуск Jupyter notebook?

Для запуска Jupyter notebook необходимо перейти в папку Scripts и в командной строке набрать команду: `ipython notebook`.

2. Какие существуют типы ячеек в Jupyter notebook?

Существует два вида ячеек:

- 1) Ячейка кода содержит код, который должен быть выполнен в ядре, и отображает его вывод ниже;
- 2) Ячейка Markdown содержит текст, отформатированный с использованием Markdown, и отображает его вывод на месте при запуске.
3. Как осуществляется работа с ячейками в Jupyter notebook?

После выбора ячейки «Code», можно записать код на языке Python, а затем нажать Ctrl+Enter или Shift+Enter. В первом случае введенный код будет выполнен интерпретатором Python, а во втором – будет создана новая ячейка, которая расположится уровнем ниже.

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

Важной частью функционала Jupyter notebook является поддержка магии. Под магией понимаются дополнительные команды, выполняемые в рамках оболочки, которые облегчают процесс разработки и расширяют возможности. Команда: `%lsmagic`

Для работы с переменными окружениями используется команда `%env`.

Запуск кода с расширением `.ipynb` осуществляется с помощью команды `%run`.

Для измерения времени работы необходимо использовать команды `%%time` и `%timeit`.

`%matplotlib` используется для отображения объектов графиков на экране, ключ после него указывает каким способ отображать график.

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

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

Шаги работы:

- 1) Создать новый проект.
- 2) Создать файл `.ipynb` (File => New => Jupyter Notebook).
- 3) В случае, если пакет не установлен, появится сообщение об ошибке. Рядом с этим сообщением будет опция «Установить пакет jupyter»
- 4) После установки необходимо создать ячейки кода и выполнить их.
- 5) После запуска можно посмотреть сервер над окном кода, а рядом с ним находится ядро, созданное как «Python2» или «Python 3»
- 6) Получен доступ к вкладке переменных в PyCharm. В ней можно увидеть, как значения переменных меняются при выполнении ячеек кода. Так же возможно установить точки останова в строках кода (команда Выполнить => Debug Cell)

Visual Studio Code:

Чтобы создать новый Jupyter Notebook, необходимо выполнить команду Ctrl+Shift+P.

После создания блокнота, его необходимо сохранить в рабочем пространстве (команда Save).

Теперь можно данный блокнот экспортировать как скрипт Python или файл HTML/PDF.

По умолчанию в новом блокноте появится пустая ячейка. Необходимо добавить в нее код и выполнить его с помощью команды Ctrl+Enter. Эта команда запустит выделенную ячейку.

Иконка + добавляет новую ячейку для кода, а bin удаляет ее. Чтобы перемещать фрагменты вверх/вниз, необходимо воспользоваться стрелками.

Чтобы изменить тип ячейки на Markdown, необходимо нажать на иконку M, расположенную над кодом. А чтобы снова установить значение code, необходимо выбрать значок {}.