

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций
«Работа с IPython и Jupyter Notebook»

Отчет по лабораторной работе № 3.1
по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1

Кучеренко С. Ю. « » 2023 г.

Подпись студента _____

Работа защищена « » _____ 2023 г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

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

Выполнение работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.

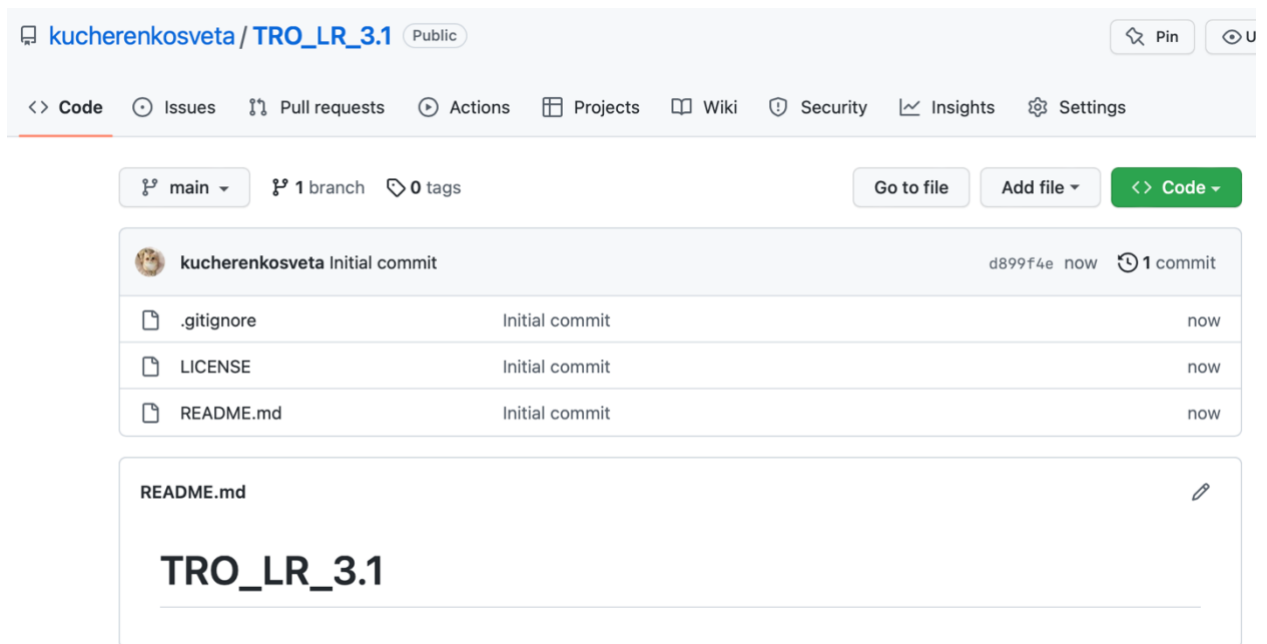


Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория.

```
LR_1 — -zsh — 80x24
Last login: Sun Feb 12 14:15:38 on ttys000
[(base) svetik@MacBook-Air-Svetik ~ % cd ~/Desktop/TPO
[(base) svetik@MacBook-Air-Svetik TPO % cd ~/LR_1
cd: no such file or directory: /Users/svetik/LR_1
[(base) svetik@MacBook-Air-Svetik TPO % cd LR_1
[(base) svetik@MacBook-Air-Svetik LR_1 % git clone https://github.com/kucherenkosveta/TRO_LR_3.1.git
Cloning into 'TRO_LR_3.1'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
(base) svetik@MacBook-Air-Svetik LR_1 %
```

Рисунок 2 – Клонирование репозитория

4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```

((base) svetik@MacBook-Air-Svetik TRO_LR_3.1 % git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/Users/svetik/Desktop/TPO/TRO_LR_3.1/.git/hooks]
(base) svetik@MacBook-Air-Svetik TRO_LR_3.1 %

```

Рисунок 3 – Организация репозитория в соответствии с моделью git-flow

5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

6. Проработать примеры лабораторной работы.

```

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

In [3]: 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 4 – Примеры из лабораторной работы

7. Решить задания в ноутбуках, выполненных преподавателем.

```
In [ ]: ticket_number = # any 6-digit number
```

```
In [12]: ticket_number = list(map(int, list(input())))  
123456
```

```
In [13]: if sum(ticket_number[:3]) == sum(ticket_number[3:]):  
          print("Yes")  
          else:  
            print("No")  
No
```

Рисунок 5 – Решение задания 1

```
In [ ]: password = # any password
```

```
In [3]: password = input('Enter password: ')  
name = input('Enter your name: ')  
Enter password: an12dRei  
Enter your name: andrei
```

```
In [5]: if password.isalpha() or password.isdigit():  
          print("weak")  
          exit(-1)  
if password.islower() or password.isupper():  
    print("weak")  
    exit(-1)
```

```
In [9]: un_symb = set(password)  
if len(un_symb) < 4:  
    print('weak')  
    exit(-1)
```

```
In [10]: if name.lower() in password.lower():  
           print("weak")  
           exit(-1)  
          else:  
            print("strong")  
strong
```

Рисунок 6 – Решение задания 2

```
In [ ]: amount = # any number
```

```
In [15]: amount = int(input()) - 1
```

10

```
In [18]: a, b = 0, 1
for i in range(amount):
    a, b = b, a + b
    print(b)
```

1
2
3
5
8
13
21
34
55

Рисунок 7 – Решение задания 3

```
In [35]: import csv
from math import sqrt

with open('organizations_gdp_hist.csv', 'r', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=',')
    total_gdp = []
    gdp_var = []
    for row in data:
        if row[4] == "Ingreso alto":
            total_gdp.append(float(row[6]))
            gdp_var.append(float(row[8]))
```

```
In [38]: av_total = sum(total_gdp) / len(total_gdp)
av_var = sum(gdp_var) / len(gdp_var)
print('Среднее значение ВВП страны: '
      f"{av_total}")
print('Среднее значение изменения ВВП в прошлом году: '
      f"{av_var}")
```

Среднее значение ВВП страны: : 1249197923.1747534
Среднее значение изменения ВВП в прошлом году: : 0.2521313144269093

```
In [42]: v1 = sum((elem - av_total) ** 2 for elem in total_gdp) / len(total_gdp)
st_v1 = sqrt(v1)
v2 = sum((elem - av_var) ** 2 for elem in gdp_var) / len(gdp_var)
st_v2 = sqrt(v2)
print("Стандартное отклонение ВВП страны: "
      f"{st_v1}")
print("Стандартное отклонение изменения ВВП в прошлом году: "
      f"{st_v2}")
```

Стандартное отклонение ВВП страны: 2933781476.848383
Стандартное отклонение изменения ВВП в прошлом году: 1.3912502466006698

Задача:

В июле 2016 года планируется взять кредит в банке на пять лет в размере S тыс. рублей. Условия его возврата таковы:

- каждый январь долг возрастает на 25% по сравнению с концом предыдущего года;
- с февраля по июнь каждого года необходимо выплатить часть долга;
- в июле 2017, 2018 и 2019 долг остаётся равным S тыс. рублей;
- выплаты в 2020 и 2021 годах равны по 625 тыс. рублей;
- к июлю 2021 долг будет выплачен полностью.

Найдите общую сумму выплат за пять лет.

Решение:

Введем переменные: $k = 1 + 25/100 = 1.25$, $Y = 625\ 000$ рублей.

Общая сумма выплат: $x = 3(kS - S) + 2Y = 3S(k - 1) + 2Y$. Долг был полностью погашен последней выплатой Y .

Значит, $k(kS - Y) = Y$, а отсюда:

$$S = (k+1)Y / k^{**2} = (3Y(k+1) / k^{**2}) + 2Y = 3Y(k^{**2} - 1) / k^{**2} + 2Y$$

```
In [8]: Y = 625000
k = 1.25
S = ((3*Y * (k**2 - 1)) / k**2) + 2*Y
print(S)
1925000.0
```

Рисунок 8 - Решение экономической задачи

9. Зафиксируйте сделанные изменения в репозитории.

Вопросы для защиты работы:

1. Как осуществляется запуск Jupyter notebook?

Jupyter Notebook входит в состав Anaconda. Для запуска Jupyter Notebook перейдите в папку Scripts (она находится внутри каталога, в котором установлена Anaconda) и в командной строке наберите: «ipython notebook»

В результате будет запущена оболочка в браузере

2. Какие существуют типы ячеек в Jupyter notebook?

Ячейки в блокноте Jupyter бывают четырех типов – Code, Markdown и Raw и Headings.

Содержимое в ячейке Code обрабатывается как инструкции на языке программирования, по умолчанию используется Python.

Ячейки Markdown содержат текст, отформатированный с использованием языка markdown. Доступны все виды функций форматирования, такие как выделение текста жирным шрифтом и курсивом, отображение упорядоченного или неупорядоченного списка, отображение табличного содержимого и т.д.

Содержимое Raw ячейки не оценивается ядром notebook.

Headings-ячейка может использоваться для разбивки блокнота на разделы.

3. Как осуществляется работа с ячейками в Jupyter notebook?

Для запуска ячейки используете команды из меню Cell, либо следующие сочетания клавиш: Ctrl+Enter – выполнить содержимое ячейки. Shift+Enter – выполнить содержимое ячейки и перейти на ячейку ниже. Alt+Enter – выполнить содержимое ячейки и вставить новую ячейку ниже.

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

Magic—это отличные команды, упрощающие нашу жизнь при решении определенных задач. Часто похожи на команды Unix, но реализованы на Python. Магических команд в Python великое множество!

Существует 2 типа магических команд: строчные (применяются к одной строке) и ячейечные (применимы ко всей ячейке). Строчные команды начинаются с символа %, а ячейечные—с двух %%. Просмотреть все доступные команды можно через: %lsmagic

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.