

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

**Кафедра инфокоммуникаций**

**Анализ данных**

**Отчет по лабораторной работе №3.1**

Тема: «Работа с  
IPython и Jupyter Notebook»

Выполнил студент группы

ИВТ-б-о-21-1

Богадунов В.И. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь 2023

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

## Ход работы:

**1. Создал репозиторий в GitHub,** дополнил правила в .gitignore для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

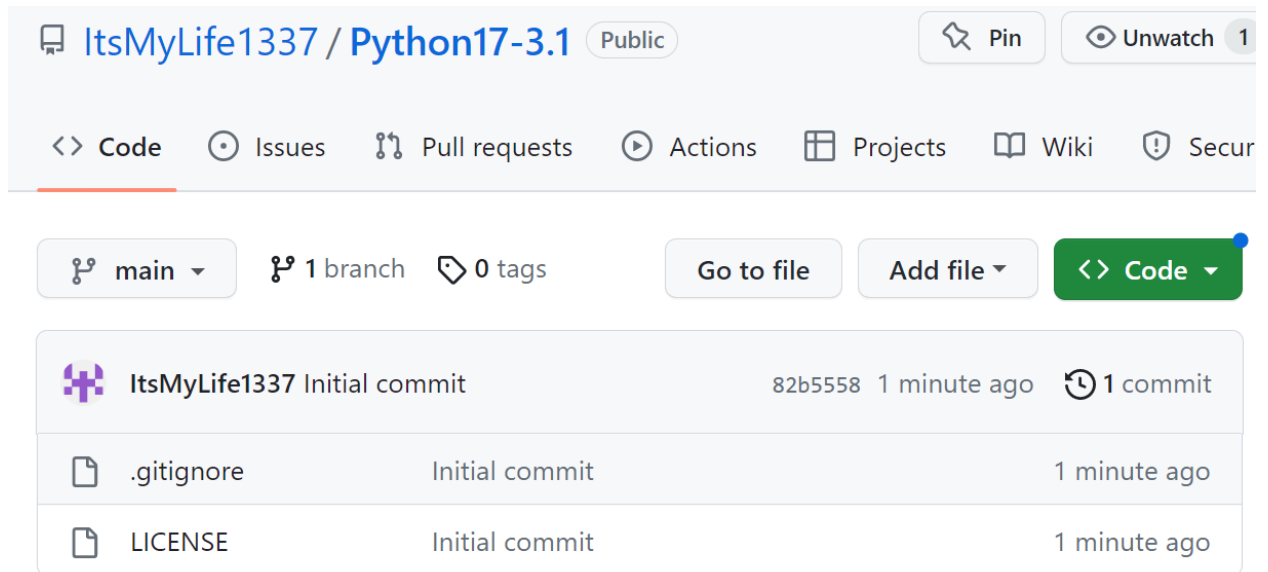


Рисунок 1.1 – Созданный репозиторий

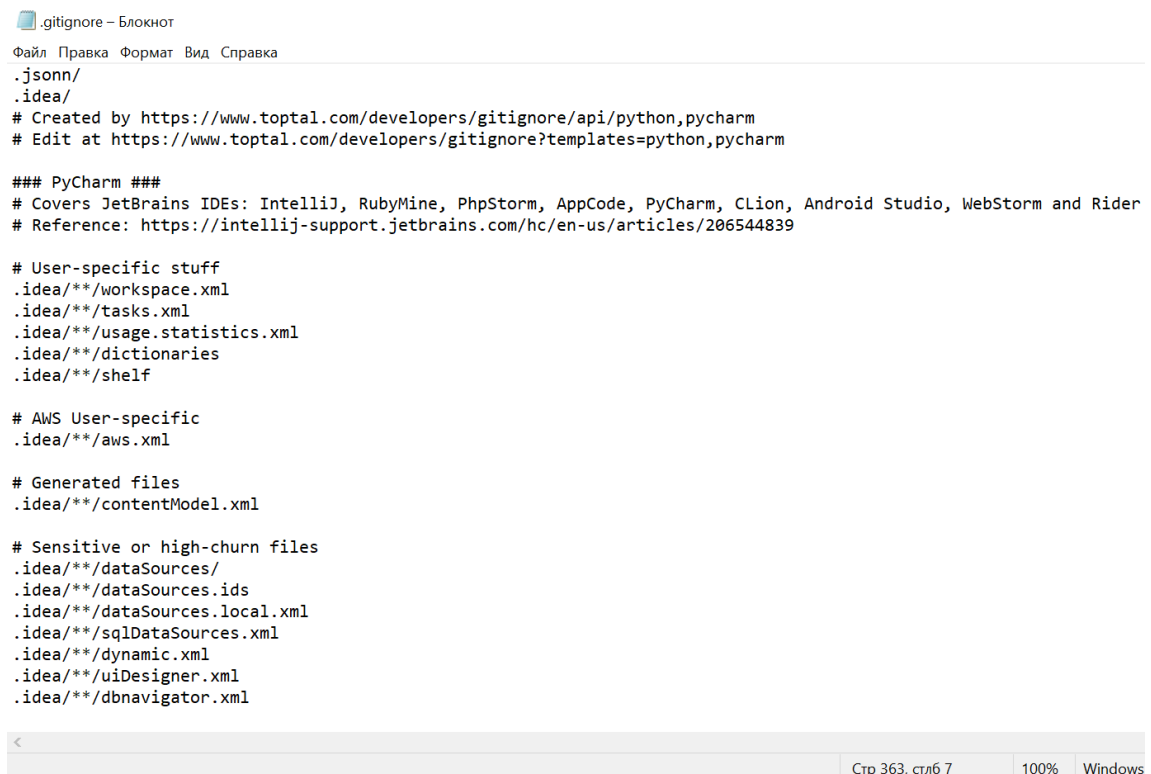


Рисунок 1.2 – Изменения в .gitignore

```

C:\Users\Admin\Desktop\git\Python17-3.1>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? [C:/Users/Admin/Desktop/git/Python17-3.1/.git/hooks]

C:\Users\Admin\Desktop\git\Python17-3.1>

```

Рисунок 1.3 – Организация репозитория в соответствии с моделью ветвления git-flow

## 2. Выполнение заданий.

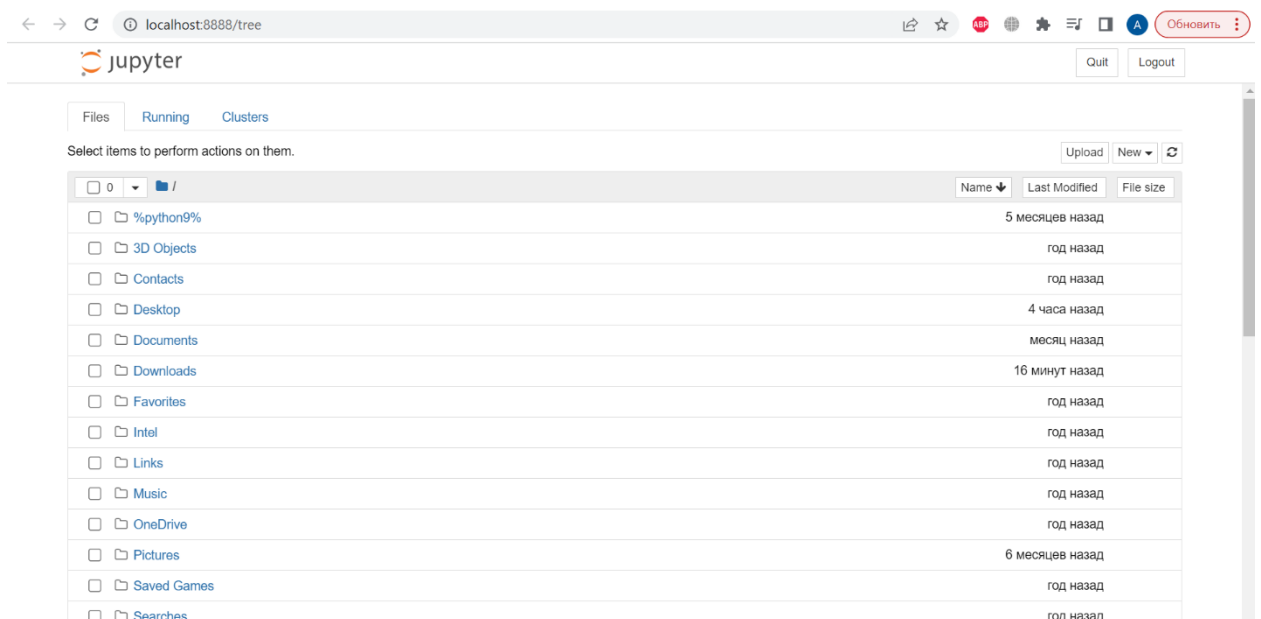


Рисунок 2.1 – Запущенная в браузере оболочка

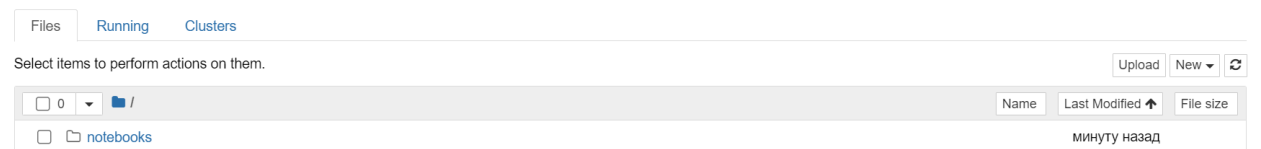


Рисунок 2.2 – Новая папка для примеров ЛР

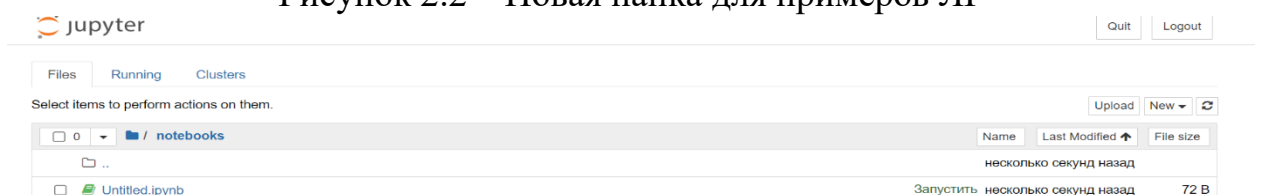


Рисунок 2.3 – Созданный ноутбук

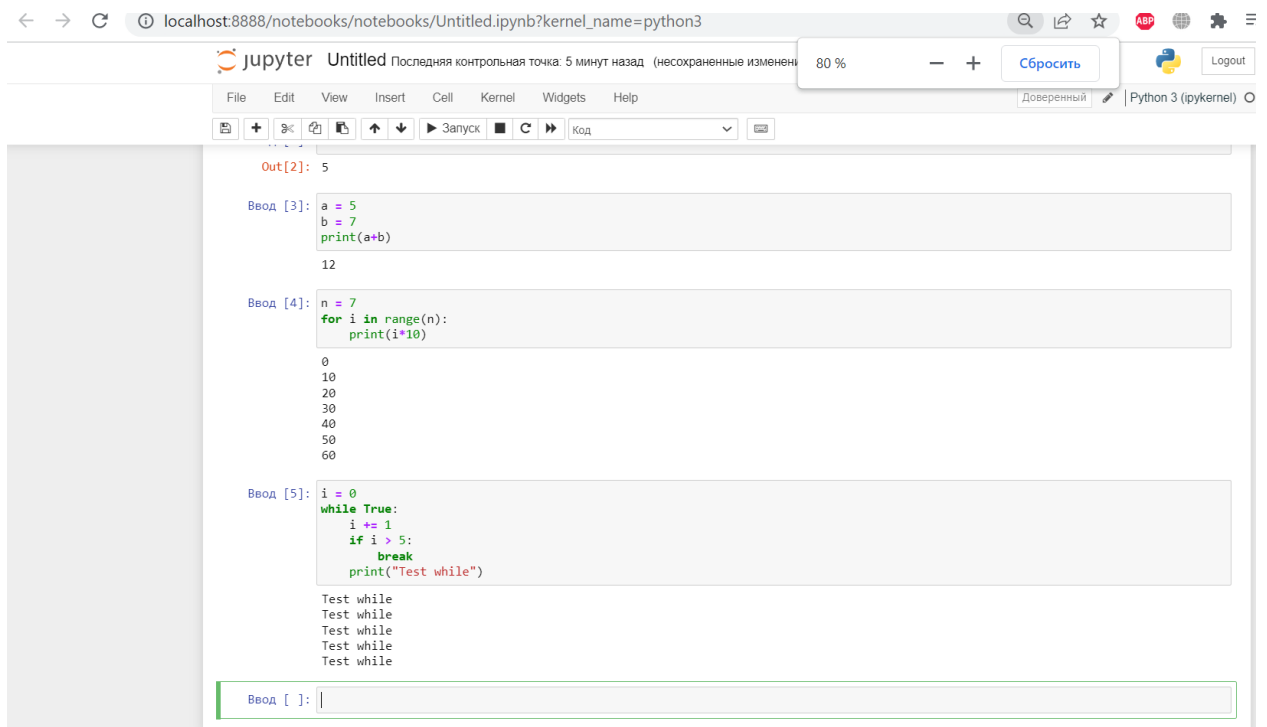


Рисунок 2.4 – Проработка примеров лабораторной работы

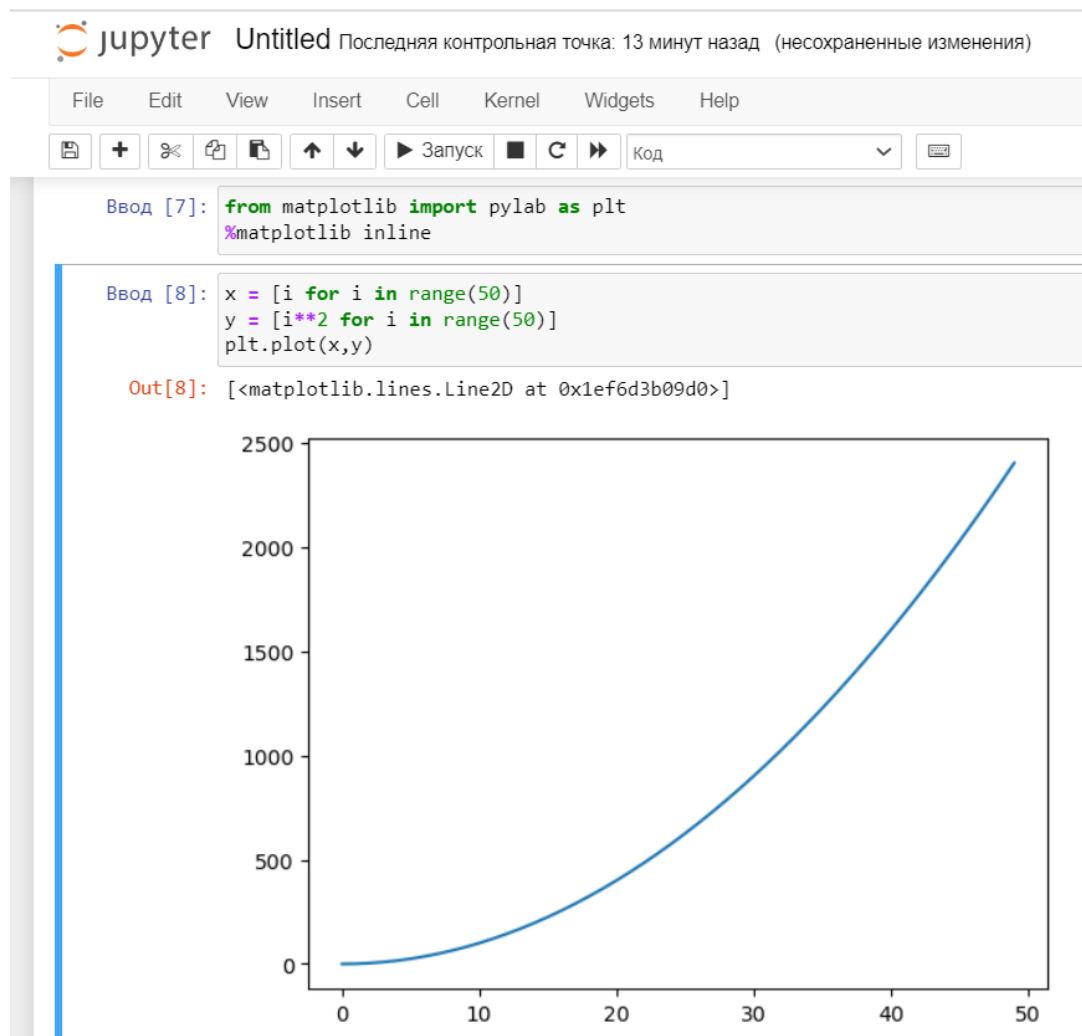


Рисунок 2.5 – Пример построения графика

Ввод [10]: `%lsmagic`

Out[10]: Available line magics:

```
%alias %alias_magic %autoawait %autocall %automagic %autosave
%bookmark %cd %clear %cls %colors %conda %config %connect_info
%copy %ddir %debug %dhist %dirs %doc %test_mode %echo %ed %edit
%env %gui %hist %history %killbgscripts %ldir %less %load %load_ext
%loadpy %log %logoff %logon %logstart %logstate %logstop %ls %lsmagic
%macro %magic %matplotlib %mkdir %more %notebook %page %pastebin
%pdb %pdef %pdoc %pfile %pinf %pinf2 %pip %popd %pprint %precision
%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 %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.

Ввод [ ]:

Рисунок 2.6 – Просмотр дополнительных команд

Ввод [14]: `%run ./Untitled1.ipynb`

```
Hello world!
Hello world!
Hello world!
Hello world!
Hello world!
```

Ввод [ ]:

Рисунок 2.7 – Запуск кода из другого ноутбука

Ввод [17]: `%%time`  
`import time`  
`for i in range(50):`  
 `time.sleep(0.1)`

Wall time: 5.51 s

Ввод [18]: `%timeit x = [(i**10) for i in range(10)]`

4.16  $\mu$ s  $\pm$  128 ns per loop (mean  $\pm$  std. dev. of 7 runs, 100000 loops each)

Рисунок 2.8 – Измерение времени с помощью `%%time` и `%timeit`


## Индивидуальное задание.

1.1. Измерив рост всех студентов в группе, получили следующие данные: 1,64 м, 1,86 м, 1,72 м, 1,95 м, 1,76 м, 1,65 м, 1,79 м, 1,82 м, 1,92 м.













Найти средний рост студентов в группе.

Решение данной задачи в ноутбуке:

localhost:8888/notebooks/notebooks/Untitled2.ipynb?kernel\_name=python3

 **jupyter** Untitled2 Последняя контрольная точка: 20 минут назад (несохраненные изменения)

File Edit View Insert Cell Kernel Widgets Help

        Запуск    Код 

```
Ввод [29]: stat = [1.64, 1.86, 1.72, 1.95, 1.76, 1.65, 1.79, 1.82, 1.92]
count = len(stat)
i = 0
co = 0
l = 0
for i, co in enumerate(stat):
    l = l + stat[i]
print("Средний рост в классе: ", l/count)

Средний рост в классе: 1.79
```

Рисунок 2.9 – Решённая задача

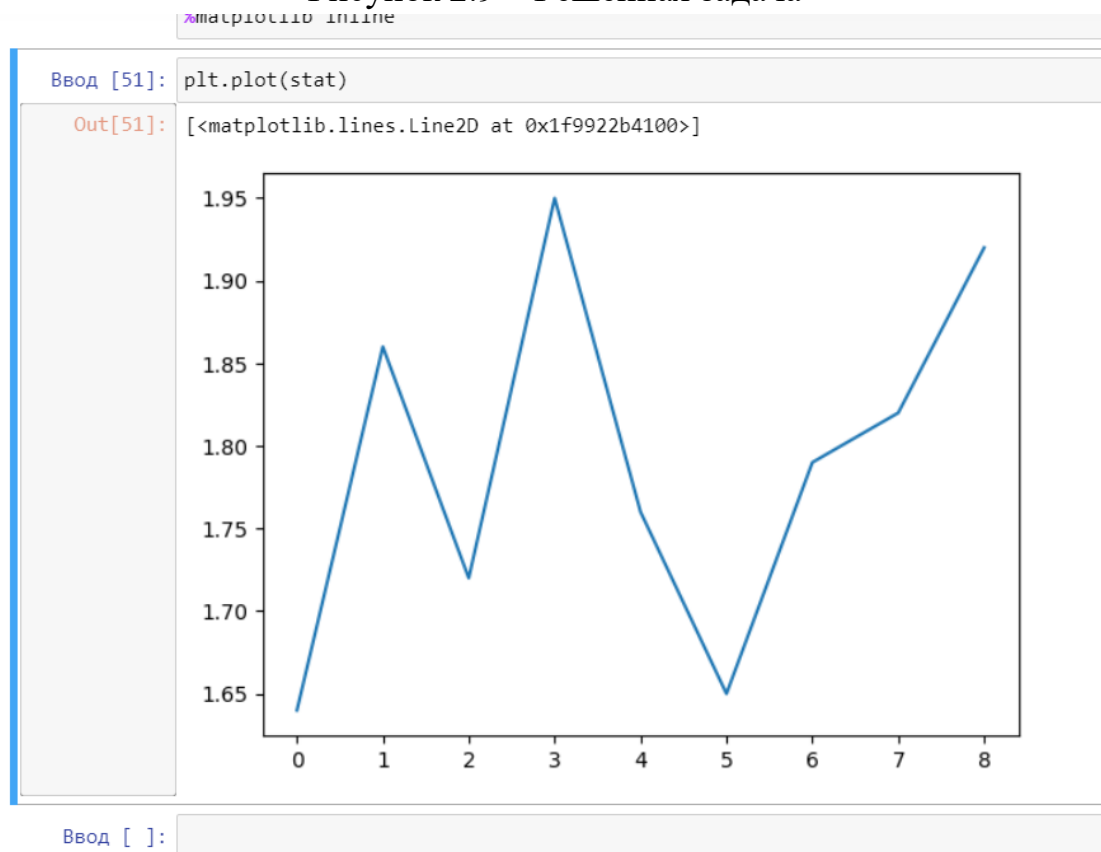
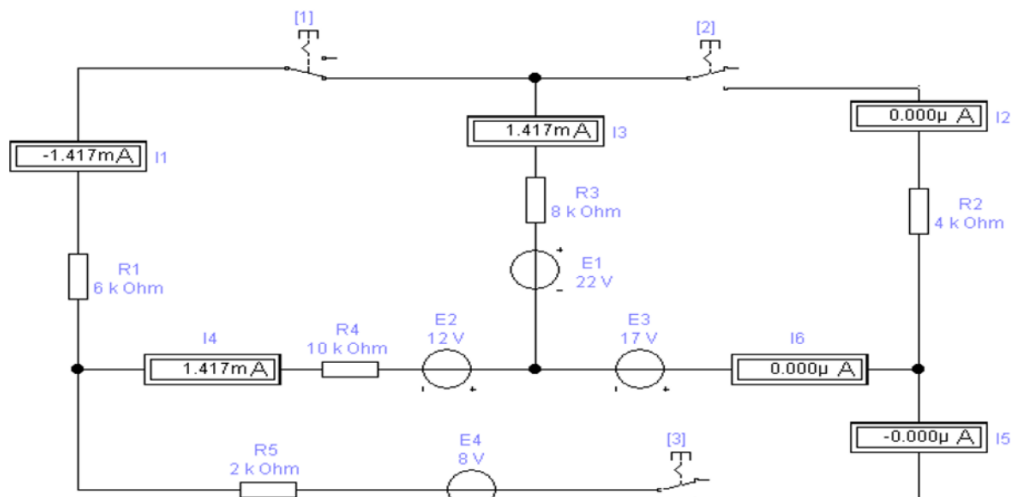


Рисунок 2.10 – График роста учеников в классе

Проверка выполнения второго закона Кирхгофа для заданного контура. Вид:



Рассчитать силу тока при разных замкнутых ключах. И проверить выполнение второго закона Кирхгофа.

Рисунок 2.11 – Индивидуальное задание №2

```
Ввод [59]: print('Проверим выполнение второго закона Кирхгофа:')
U1=I1*(R1+R3+R4)
U11=E1+E2
print(U1, ' ', U11)
print('При разомкнутых ключах 1 и 3.')
U2=I2*(R2+R3)
U22=E1-E3
print(abs(U2), ' ', U22)
print('При разомкнутых ключах 1 и 2.')
U3=I4*(R2+R3)
U33=E2+E3-E4
print(U3, ' ', U33)
print('Вывод: второй закон Кирхгофа выполняется!')
```

```
Проверим выполнение второго закона Кирхгофа:
34.0 34
При разомкнутых ключах 1 и 3.
5.0 5
При разомкнутых ключах 1 и 2.
21.0 21
Вывод: второй закон Кирхгофа выполняется!
```

Рисунок 2.12 – Результат

**Вывод:** в результате выполнения лабораторной работы были получены базовые навыки для работы с интерактивными оболочками IPython и Jupyter Notebook для языка программирования Python.

### Ответы на контрольные вопросы:

#### 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?**

Code – для кода, Markdown – для текста

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

### **Запуск и прерывание выполнения кода**

Если ваша программа зависла, то можно прервать ее выполнение выбрав на панели меню пункт

Kernel -> Interrupt.

Для добавления новой ячейки используйте Insert->Insert Cell Above и Insert->Insert Cell Below.

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?**



Важной частью функционала Jupyter Notebook является поддержка магии. Под магией в IPython понимаются дополнительные команды, выполняемые в рамках оболочки, которые облегчают процесс разработки и расширяют ваши возможности. Список доступных магических команд можно получить с помощью команды: `%lsmagic`

Ввод [10]: `%lsmagic`

Out[10]: 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 %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 %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.

Ввод [ ]: |

## 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.

PyCharm

1. Сначала вы должны создать новый проект.

2. В этом проекте создайте новый файл `ipynb`, выбрав `File> New...> Jupyter Notebook`. Это должно открыть новый файл записной книжки.

3. Если у вас не установлен пакет Jupyter Notebook, над вновь открытым файлом `ipynb` появится сообщение об ошибке. Сообщение об ошибке гласит: «Пакет Jupyter не установлен», и у вас будет опция

«Установить пакет jupyter» рядом с ним.

4. Нажмите «Установить пакет jupyter». Это запустит процесс установки, который вы можете просмотреть, щелкнув запущенные процессы в правом нижнем углу окна PyCharm.

5. Чтобы начать изучение Jupyter Notebook в PyCharm, создайте ячейки кода и выполните их.

6. Выполните ячейку кода, чтобы запустить сервер Jupyter. По умолчанию сервер Jupyter использует порт 8888 по умолчанию на локальном хосте. Эти конфигурации доступны в окне инструментов сервера. После запуска вы можете просмотреть сервер над окном исходного кода, а рядом с ним вы можете просмотреть ядро, созданное как «Python 2» или «Python 3».

7. Теперь вы можете получить доступ к вкладке переменных в PyCharm, чтобы увидеть, как значения ваших переменных меняются при выполнении ячеек кода. Это помогает при отладке. Вы также можете установить точки останова в строках кода, а затем щелкнуть значок

«Выполнить» и выбрать «Debug Cell» (или использовать сочетание клавиш `Alt+Shift+Enter`), чтобы начать отладку.

Visual Studio Code:

- Если у вас еще нет существующего файла Jupyter Notebook, откройте VS Code Command Palette с помощью сочетания клавиш

CTRL+SHIFT+P (Windows) или Command+SHIFT+P (macOS) и запустите команду «Python: Create Blank New Jupyter Notebook».

- Если у вас уже есть файл Jupyter Notebook, это так же просто, как просто открыть этот файл в VS Code. Он автоматически откроется с новым нативным редактором Jupyter.