

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

Институт цифрового развития
Кафедра инфокоммуникаций

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3.1**

Выполнила:
Кузнецова Алена Валерьевна
1 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»,
очная форма обучения

(подпись)

Ставрополь, 2023 г.

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

Выполнение работы:

1. Создали общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

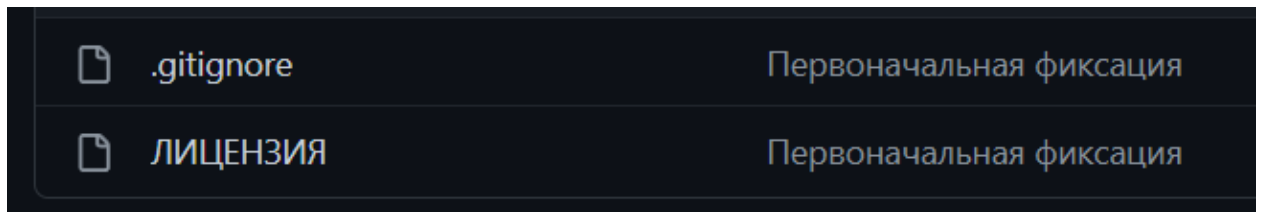


Рисунок 1 – Создание репозитория

2. Выполнили клонирование созданного репозитория.

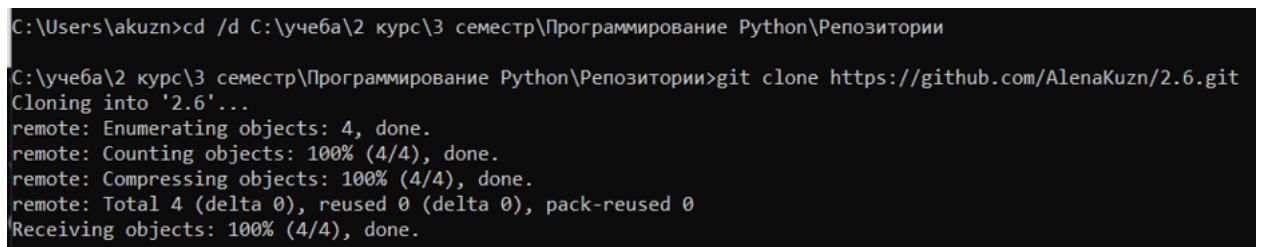


Рисунок 2 – Клонирование репозитория

3. Организовали свой репозиторий в соответствии с моделью git-flow.

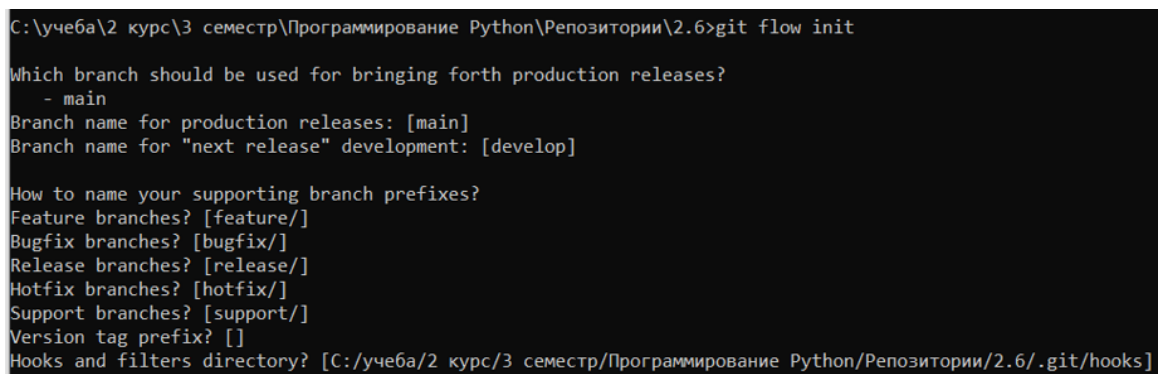


Рисунок 3 – Моделью ветвления git-flow

4. Проработали примеры лабораторной работы.

```
Ввод [1]: 2 + 3
Out[1]: 5

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

Ввод [3]: n = 7
          for i in range(n):
              print(i * 10)
0
10
20
30
40
50
60

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

Рисунок 4 – Примеры

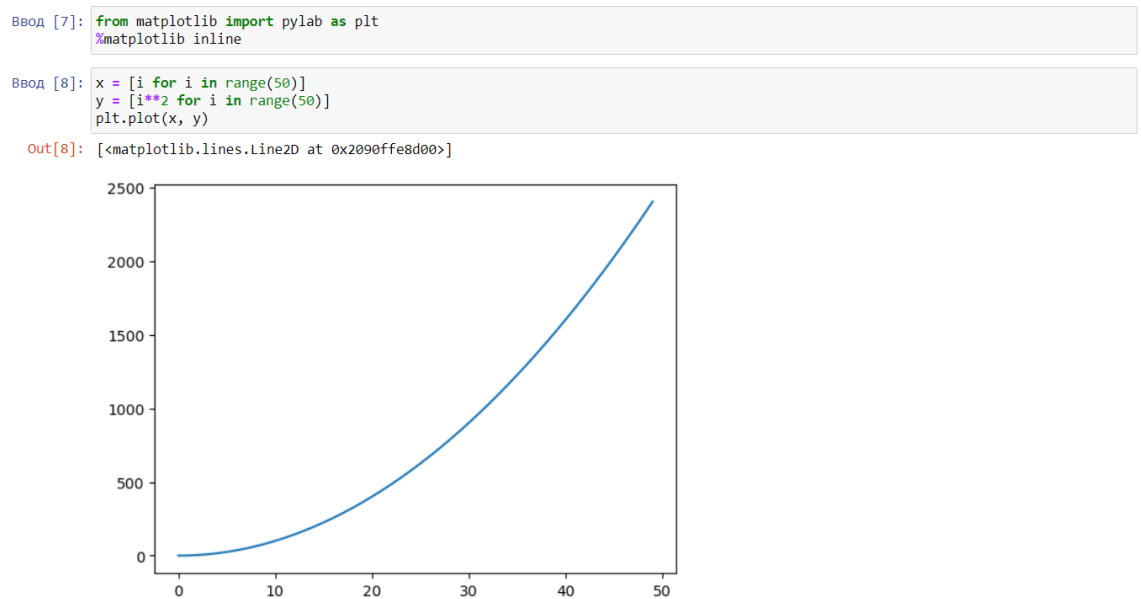


Рисунок 5 – Пример построения графика

5. Решили задания в ноутбуках.

Задание:

- 1) Определите число `ticket_number` — шестизначный номер билета;
- 2) Напишите код, который по шестизначному номеру `ticket_number` билета 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

```
Ввод [2]: t = int(input())

summa1 = t // 100000 + t // 10000 % 10 + t // 1000 % 10
summa2 = t % 1000 // 100 + t % 100 // 10 + t % 10

if (summa1 == summa2):
    print("YES")
else:
    print("NO")

123060
YES
```

Рисунок 6 – Задания

6. Создали ноутбук, в котором выполнили решение вычислительной задачи.

```
Ввод [2]: import numpy as np
import matplotlib.pyplot as plt

#Функция скорости
def v(t):
    return 18*t - 9*t**2

#Функция перемещения
def s(t):
    return [np.trapz(v(t)[t <= i], t[t <= i]) for i in t]

t = np.linspace(0, 15)
s_values = s(t)

#Построение графика
plt.plot(t, s_values)
plt.xlabel('Time')
plt.ylabel('Pyt')
plt.show()
```

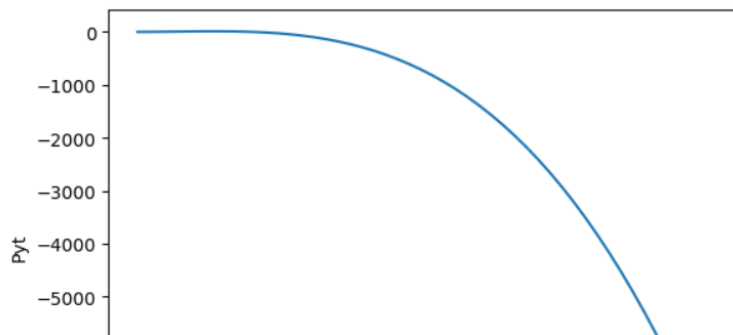


Рисунок 7 – Индивидуальное задание

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?

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

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.