

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Визуализация данных с помощью matplotlib»**

**ОТЧЕТ**  
**по лабораторной работе №5**  
**дисциплины**  
**«Технологии распознавания образов»**

Выполнила:  
Кувшин Ирина Анатольевна  
2 курс, группа ПИЖ-б-о-21-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

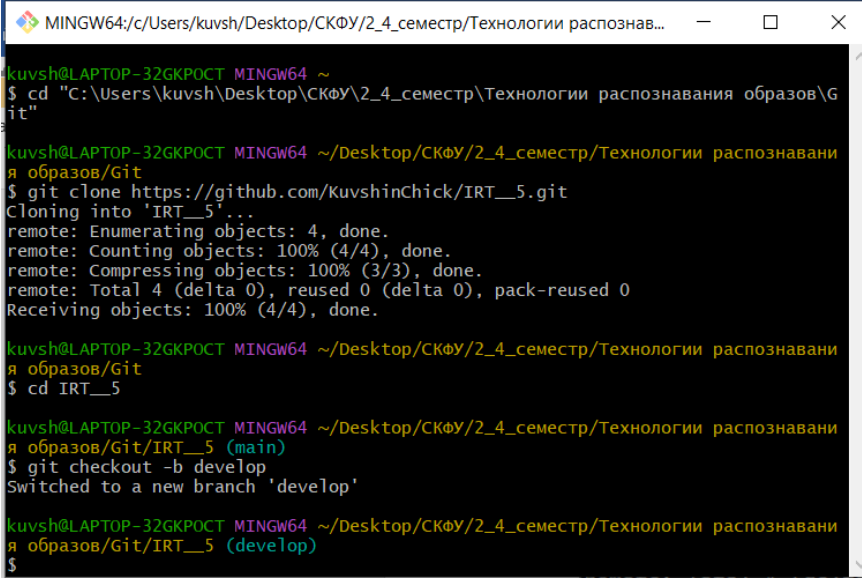
(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Цель работы:** исследовать базовые возможности визуализации данных на плоскости средствами библиотеки matplotlib языка программирования Python.

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).
3. Выполните клонирование созданного репозитория на рабочий компьютер.
4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.



```
MINGW64:~/Desktop/CKФУ/2_4_семестр/Технологии распознаван...
kuvsh@LAPTOP-32GKPOCT MINGW64 ~
$ cd "C:\Users\kuvsh\Desktop\CKФУ\2_4_семестр\Технологии распознавания образов\Git"
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Технологии распознавани
я образов/Git
$ git clone https://github.com/KuvshinChick/IRT__5.git
Cloning into 'IRT__5'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Технологии распознавани
я образов/Git/IRT__5
$ cd IRT__5
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Технологии распознавани
я образов/Git/IRT__5 (main)
$ git checkout -b develop
Switched to a new branch 'develop'
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/CKФУ/2_4_семестр/Технологии распознавани
я образов/Git/IRT__5 (develop)
$
```

Рисунок 5.1 – Клонирование репозитория и создание ветки develop

```
MINGW64/c/Users/kuvsh/Desktop/СКФУ/2_4_семестр/Технологии распознав...
kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Технологии распознавания
образов/Git/IRT__5 (develop)
$ git add .

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Технологии распознавания
образов/Git/IRT__5 (develop)
$ git commit -m "modified .gitignore & readme"
[develop 2bfadfd] modified .gitignore & readme
2 files changed, 131 insertions(+), 1 deletion(-)
create mode 100644 .gitignore

kuvsh@LAPTOP-32GKPOCT MINGW64 ~/Desktop/СКФУ/2_4_семестр/Технологии распознавания
образов/Git/IRT__5 (develop)
$ git push origin develop
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.43 KiB | 1.43 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/KuvshinChick/IRT__5/pull/new/develop
remote:
To https://github.com/KuvshinChick/IRT__5.git
 * [new branch]      develop -> develop
```

Рисунок 5.2 – Обновление .gitignore и readme

6. Проработать примеры лабораторной работы в отдельном ноутбуке.

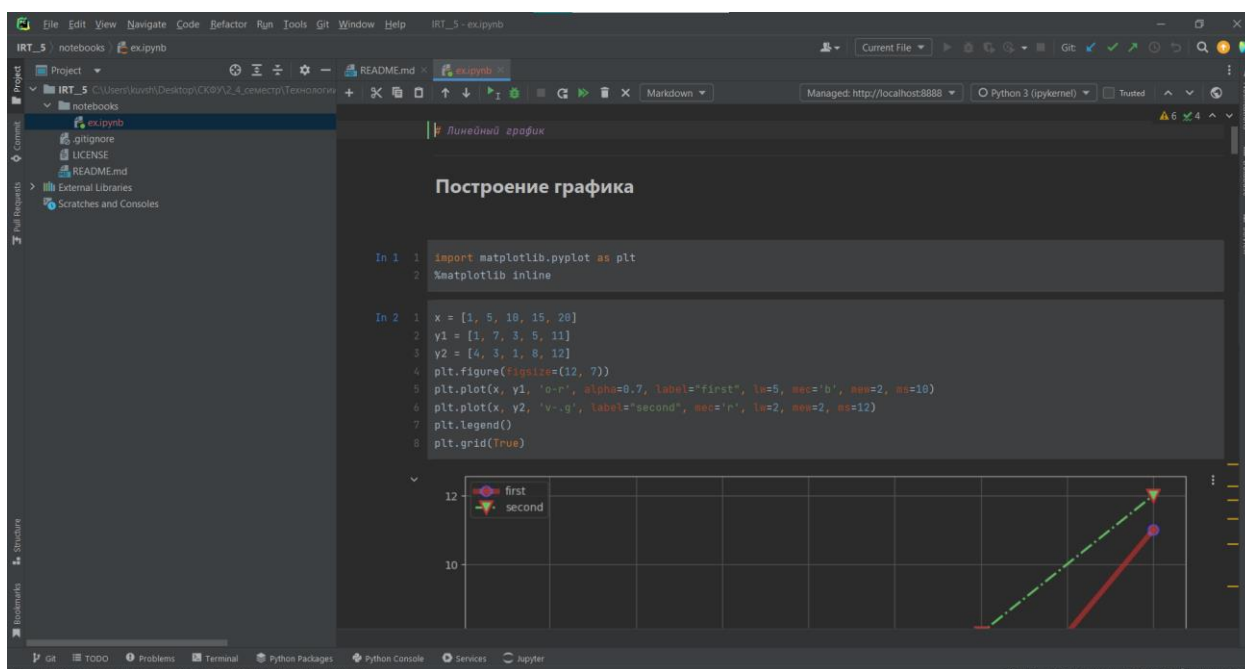


Рисунок 5.3 – Результат проработки примеров (ex.ipynb)

7. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения линейного графика, условие которой предварительно необходимо согласовать с преподавателем.

IRT\_5 - task7\_line.ipynb

Project

- IRT\_5
  - notebooks
    - task7\_line.ipynb

task7\_line.ipynb

Managed: http://localhost:8888

Python 3 (ipykernel)

Trusted

Для этого выражения среднюю арифметическую скорости и среднюю длину свободного пробега молекул:

$$\bar{v} = \sqrt{\frac{8 \times R \times T}{3 \times \pi \times \mu}}$$

$$\bar{\lambda} = \frac{1}{\sqrt{2} \times \pi \times \sigma^2 \times n}$$

Концентрация молекул:

$$n = \frac{p}{k \times T}$$

Тогда:

$$D = \frac{1}{3} \sqrt{\frac{8 \times R \times T}{3 \times \pi \times \mu}} \times \frac{k \times T}{\sqrt{2} \times \pi \times \sigma^2 \times n}$$

Подставив значения в формулу, мы получим:

$$D = 2,01 \times 10^{-8} \times T^{\frac{3}{2}} \frac{\text{м}^2}{\text{с}}$$

Построим график функции:

```
In 2 1 import matplotlib.pyplot as plt
      2 %matplotlib inline
      3
```

IRT\_5 - task7\_line.ipynb

Project

- IRT\_5
  - notebooks
    - task7\_line.ipynb

task7\_line.ipynb

Managed: http://localhost:8888

Python 3 (ipykernel)

Trusted

```
10 # Построение графика
11 plt.title("График зависимости коэффициента диффузии D водорода от температуры T") # заголовок
12 plt.xlabel("T, K") # ось абсцисс
13 plt.ylabel("D, м^2/с") # ось ординат
14 plt.grid() # включение отображения сетки
15 plt.plot(T, D) # построение графика
```

Out 2: []

График зависимости коэффициента диффузии D водорода от температуры T

T, K	D, м <sup>2</sup> /с
100	~0.00002
200	~0.00005
300	~0.00010
400	~0.00016
500	~0.00022
600	~0.00028

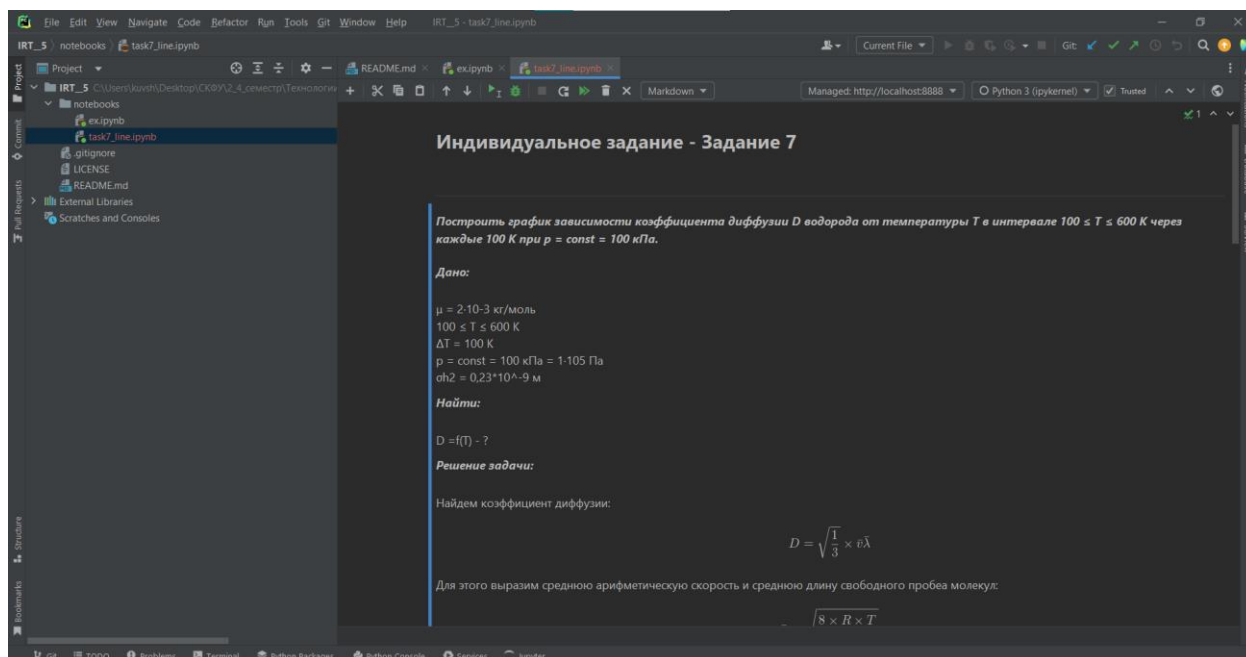


Рисунок 5.4 – Результат выполнения задания

8. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения столбчатой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

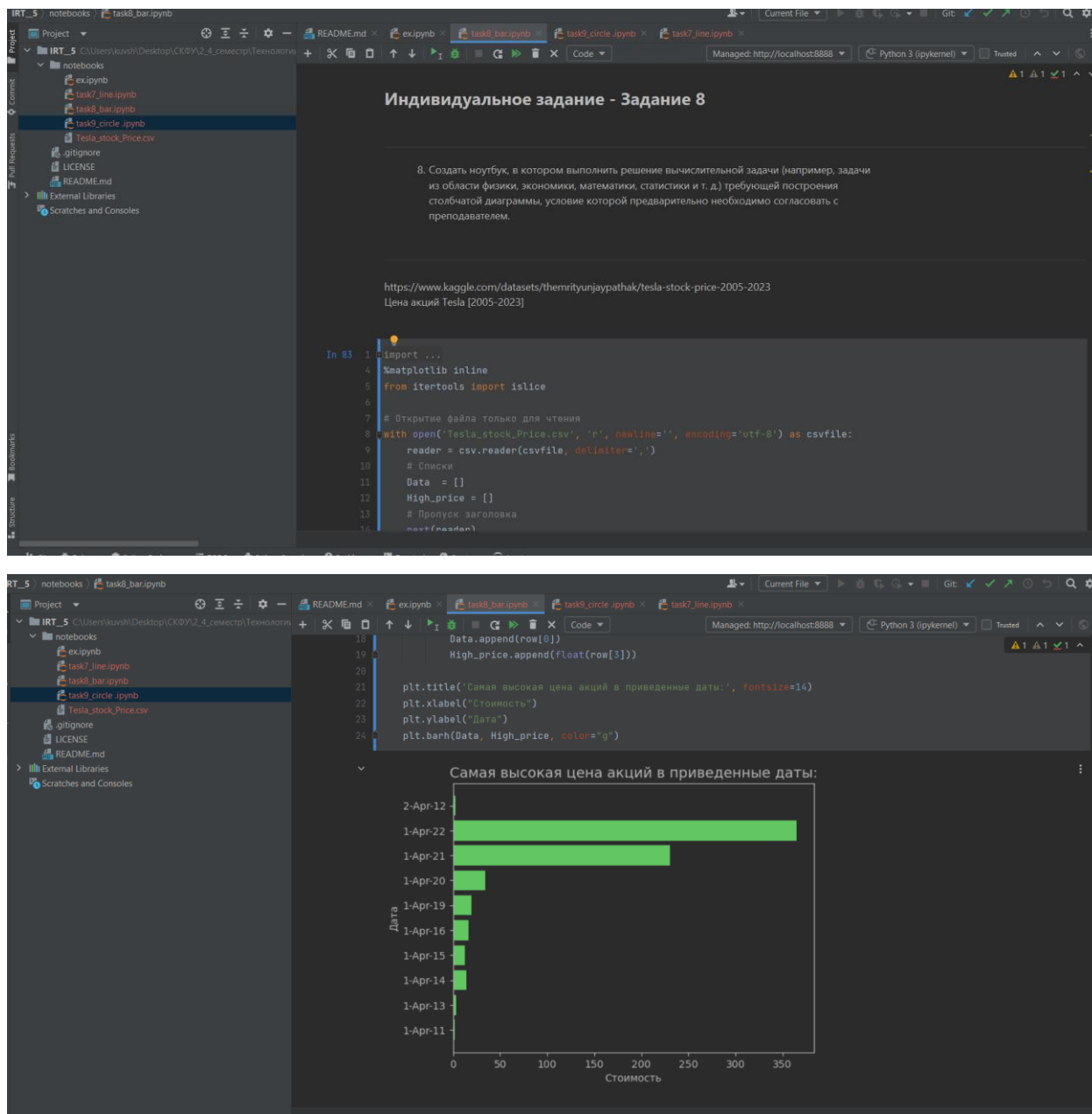


Рисунок 5.5 – Результат выполнения задания

9. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения круговой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

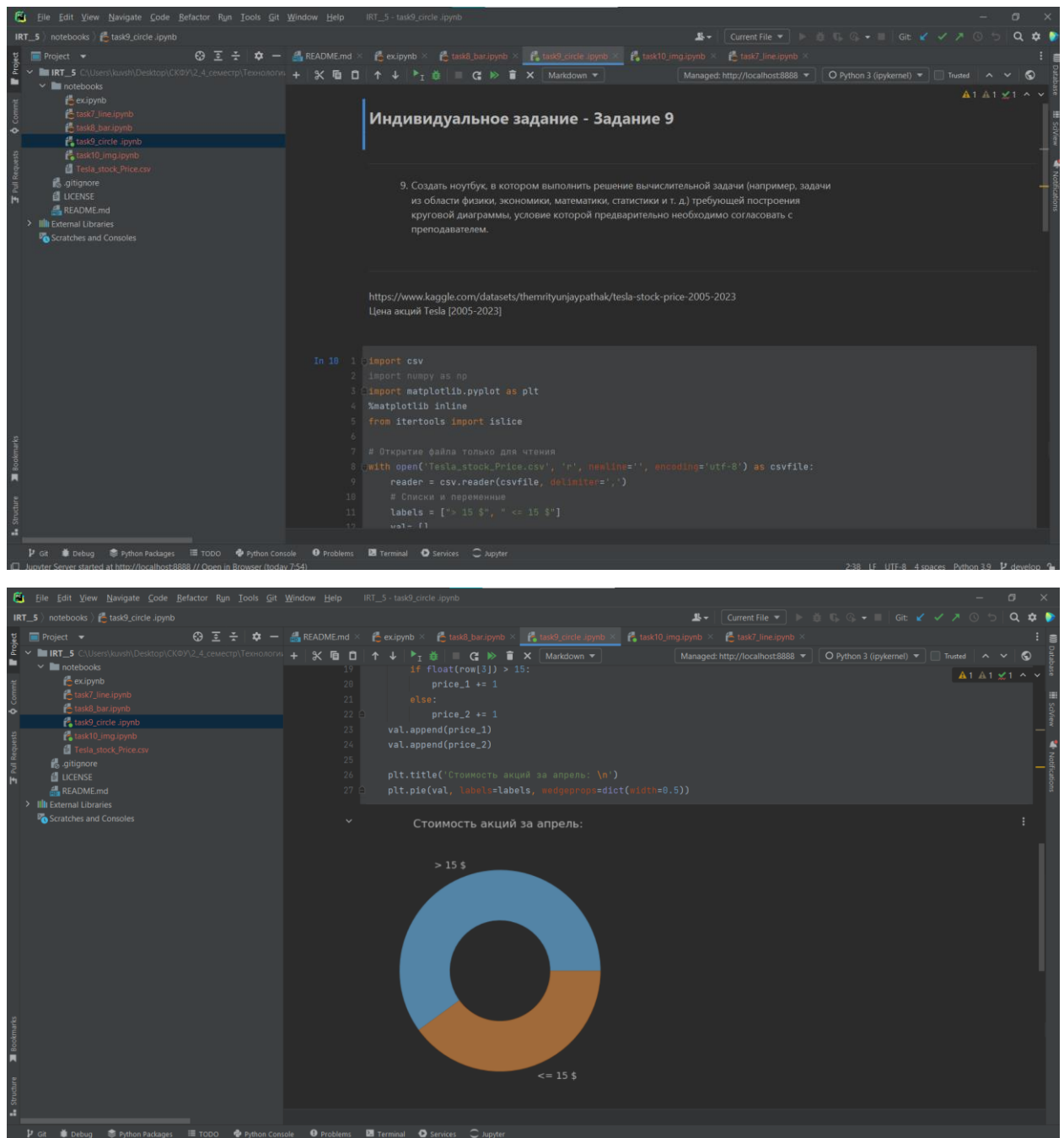


Рисунок 5.6 – Результат выполнения задания

10. Найти какое-либо изображение в сети Интернет. Создать ноутбук, в котором будет отображено выбранное изображение средствами библиотеки matplotlib по URL из сети Интернет.

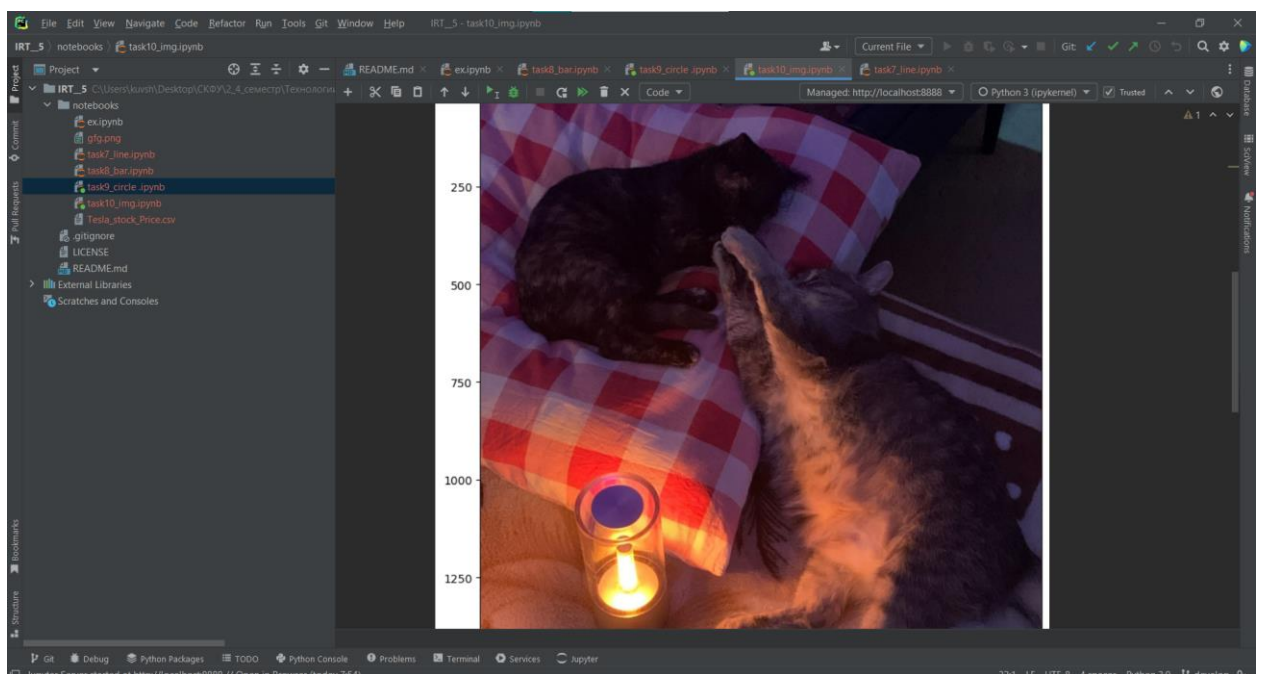
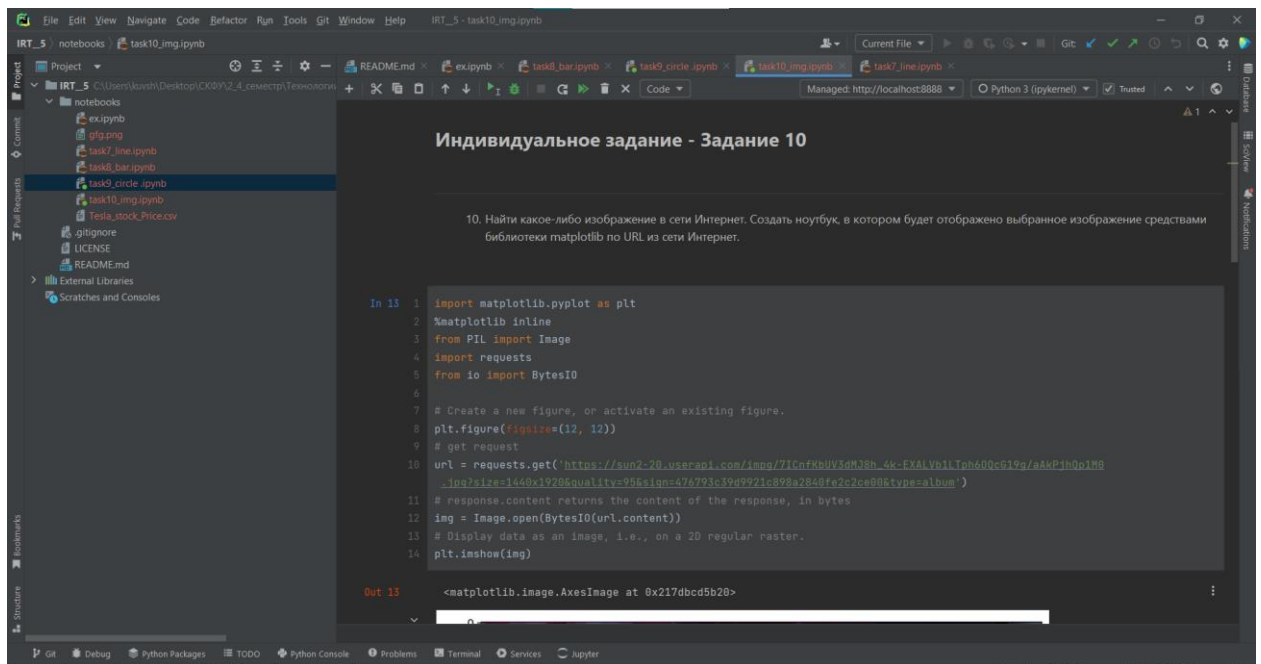


Рисунок 5.7 – Результат выполнения задания

11. Зафиксируйте сделанные изменения в репозитории.
12. Выполните слияние ветки для разработки с веткой main (master).
13. Отправьте сделанные изменения на сервер GitHub.

### Контрольные вопросы

1. Как выполнить построение линейного графика с помощью matplotlib?



Для построения линейного графика используется функция *plot()*, со следующей сигнатурой:

```
plot([x], y, [fmt], *, data=None, **kwargs)
plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)
```

2. Как выполнить заливку области между графиком и осью? Между двумя графиками?

Для заливки областей используется функция *fill\_between()*. Сигнатура функции:

```
fill_between(x, y1, y2=0, where=None, interpolate=False, step=None, *,
data=None, **kwargs)
```

3. Как выполнить выборочную заливку, которая удовлетворяет некоторому условию?

```
plt.plot(x, y, c="r")
plt.fill_between(x, y, where=(y > 0))
```

4. Как выполнить двухцветную заливку?

Вариант двухцветной заливки:

```
plt.plot(x, y, c="r")
plt.grid()

plt.fill_between(x, y, where=y>=0, color="g", alpha=0.3)
plt.fill_between(x, y, where=y<=0, color="r", alpha=0.3)
```

5. Как выполнить маркировку графиков?

```
plt.plot(x, y, marker="o", c="g")
```

6. Как выполнить обрезку графиков?

Для того, чтобы отобразить только часть графика, которая отвечает определенному условию используйте предварительное маскирование данных с помощью функции *masked\_where* из пакета *numpy*.

```
x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)

y_masked = np.ma.masked_where(y < -0.5, y)
plt.ylim(-1, 1)

plt.plot(x, y_masked, linewidth=3)
```

## 7. Как построить ступенчатый график?

```
x = np.arange(0, 7)
y = x

where_set = ['pre', 'post', 'mid']
fig, axs = plt.subplots(1, 3, figsize=(15, 4))

for i, ax in enumerate(axs):
    ax.step(x, y, "g-o", where=where_set[i])
    ax.grid()
```

## 8. Как построить стековый график?

```
x = np.arange(0, 11, 1)

y1 = np.array([(-0.2)*i**2+2*i for i in x])
y2 = np.array([(-0.4)*i**2+4*i for i in x])
y3 = np.array([2*i for i in x])

labels = ["y1", "y2", "y3"]

fig, ax = plt.subplots()

ax.stackplot(x, y1, y2, y3, labels=labels)
ax.legend(loc='upper left')
```

## 9. Как построить stem-график?

```
x = np.arange(0, 10.5, 0.5)
y = np.array([(-0.2)*i**2+2*i for i in x])

plt.stem(x, y)
```

## 10. Как построить точечный график?

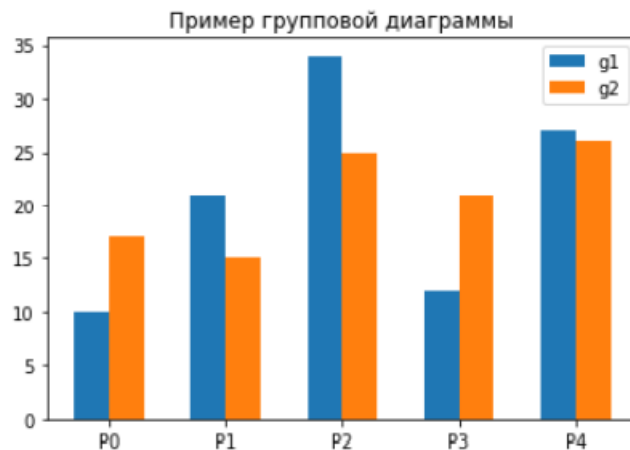
```
x = np.arange(0, 10.5, 0.5)
y = np.cos(x)

plt.scatter(x, y)
```

11. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

*bar()* – для построения вертикальной диаграммы  
*barh()* – для построения горизонтальной диаграммы.

12. Что такое групповая столбчатая диаграмма? Что такое столбчатая диаграмма с *errorbar* элементом?



*Errorbar* элемент позволяет задать величину ошибки для каждого элемента графика. Для этого используются параметры *xerr*, *yerr* и *ecolor* (для задания цвета):

13. Как выполнить построение круговой диаграммы средствами matplotlib?

Круговые диаграммы – это наглядный способ показать доли компонент в наборе. Они идеально подходят для отчетов, презентаций и т.п. Для построения круговых диаграмм в *Matplotlib* используется функция *pie()*.

## 14. Что такое цветовая карта? Как осуществляется работа с цветовыми картами в matplotlib?

Цветовая карта представляет собой подготовленный набор цветов, который хорошо подходит для визуализации того или иного набора данных. Подробное руководство по цветовым картам вы можете найти на официальном сайте *Matplotlib* (<https://matplotlib.org/tutorials/colors/colormaps.html#sphx-glr-tutorials-colors-colormaps-py>). Также отметим, что такие карты можно создавать самостоятельно, если среди существующих нет подходящего решения. Ниже представлены примеры некоторых цветовых схем, из библиотеки *Matplotlib*.

## 15. Как отобразить изображение средствами matplotlib?

```
from PIL import Image
import requests

from io import BytesIO

response = requests.get('https://matplotlib.org/_static/logo2.png')
img = Image.open(BytesIO(response.content))

plt.imshow(img)
```

## 16. Как отобразить тепловую карту средствами matplotlib?

```
np.random.seed(123)

data = np.random.rand(5, 7)
plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')
```