

## Дисциплина «Искусственный интеллект»

### Рабочая тетрадь № 1

Цель изучения дисциплины – формирование компетенций в области современных информационных технологий с применением инструментария искусственного интеллекта и машинного обучения. Для реализации методов и алгоритмов будем использовать язык программирования Python.

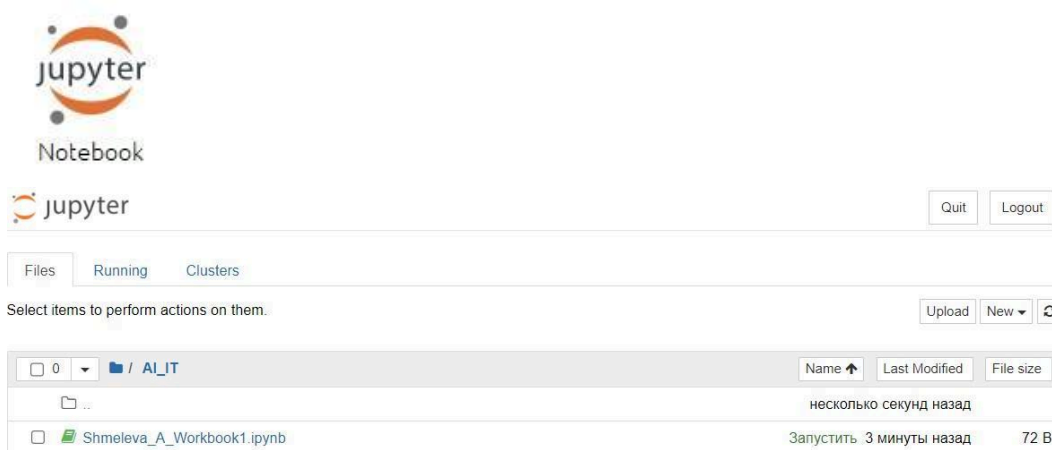
Python является популярным высокоуровневым языком программирования, который применяется для решения прикладных задач принятия решений с использованием технологий искусственного интеллекта и машинного обучения.


Для реализации методов и алгоритмов, изучаемых в данной дисциплине, рекомендуется использовать среду разработок Anaconda, содержащую библиотеки языка программирования Python с открытым исходным кодом.

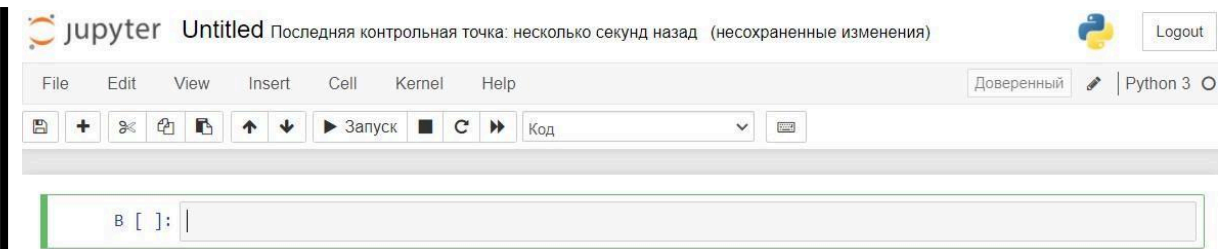
<https://www.anaconda.com/products/individual>



Для выполнения заданий из рабочих тетрадей рекомендуется использовать Jupyter Notebook.



Название рабочих тетрадей должно содержать фамилию и имя студента, номер рабочей тетради. Например,  Shmeleva\_A\_Workbook1.ipynb  
Для создания файла необходимо выбрать вкладку **New/Notebook**:



Далее **File/Rename** и указать название файла.

## 1.1. Теоретический материал – Типы данных

### Типы данных

Все типы данных в Python относятся к одной из 2-х категорий: изменяемые (mutable) и неизменяемые (immutable).

Неизменяемые объекты:

- числовые данные (int, float),
- bool,
- None,
- символьные строки (class 'str'),
- кортежи (tuple).

Изменяемые объекты:

- списки (list),
- множества (set),
- словари (dict).

## 1.2. Пример

### Задача:

Выведите на печать и определите тип переменной.

```
x = 3+5.2*7
y = None
z = 'a', 5, 12.345, (2, 'b')
df = [['Антонова Антонина', 34, 'ж'], ['Борисов Борис', 26, 'м']]
A = {1, 'title', 2, 'content'}
```

### Решение:

```
print(x, '|', type(x), '\n', y, '|', type(y), '\n',
      z, '|', type(z), '\n', df, '|', type(df), '\n', A, '|', type(A))
```

### Ответ:

```
39.4 | <class 'float'>
None | <class 'NoneType'>
('a', 5, 12.345, (2, 'b')) | <class 'tuple'>
[['Антонова Антонина', 34, 'ж'], ['Борисов Борис', 26, 'м']] | <class 'list'>
{1, 2, 'content', 'title'} | <class 'set'>
```

### 1.3. Задание

#### Задача:

Выведите на печать и определите тип переменной.

```
x = 5 >= 2
A = {1, 3, 7, 8}
B = {2, 4, 5, 10, 'apple'}
C = A & B
df = 'АНТОНОВА АНТОНИНА', 34, 'ж'
z = 'type'
D = [1, 'title', 2, 'content']
```

#### Решение:

```
data = [x,A,B,C,df,z,D]
for i in range(len(data)):
    print(data[i], '| ', type(data[i]), '\n')
```

#### Ответ:

```
True | <class 'bool'>

{8, 1, 3, 7} | <class 'set'>

{2, 4, 5, 'apple', 10} | <class 'set'>

set() | <class 'set'>

('Антонова Антонина', 34, 'ж') | <class 'tuple'>

type | <class 'str'>

[1, 'title', 2, 'content'] | <class 'list'>
```

## 2.1. Теоретический материал – Условный оператор

### If – Условный оператор

В коде часто приходится проверять выполнимость или невыполнимость каких-то условий. Синтаксис:

if <условие1 (булевское выражение)> :

<код, который выполнится, если условие верно>  
elif <условие2 (булевское выражение)>:  
    <код, который выполнится, если условие1 было неверно, а  
условие2 верно>  
else:  
    <код, который выполнится, если условие1 и условие2 были  
неверны>

Обратите внимание, что код, который должен выполняться внутри каждого условия, записывается с отступом в 4 пробела от уровня if, elif и else: в питоне области видимости переменных обозначаются отступами.

*То есть, отступы позволяют понять, где начинается код, который должен выполняться при выполнении условия в if, и где заканчивается.*

## 2.2. Пример

**Задача:**

Вывести на экран является ли переменная x положительной, отрицательной или равна нулю.

**Решение:**

```
x = 125
if x < 0:
    print("x отрицательный")
elif x == 0:
    print("x равен нулю")
else:
    print("x положительный")
```

**Ответ:**

x положительный

## 2.3. Задание

**Задача:**

Напишите код. Задается x, напечатать какому из интервалов принадлежит:  $(-\infty, -5)$ ,  $[-5, 5]$  или  $(5, +\infty)$

**Решение:**

```
x = -10
interval = ['(-infinity, -5)', '[-5,5]', '(5, +infinity)']
print(f"x принадлежит {interval[0] if x < -5 else interval[2] if x > 5 else interval[1]}")
```

**Ответ:**

x принадлежит  $(-\infty, -5)$

--	--

### 3.1. Теоретический материал – Циклы, массивы, срезы

#### Циклы - **for** и **while**

Цикл **while** проверяет истинность условия, и если условие истинно, то выполняет инструкции цикла.

Синтаксис:

```
while условие:
    инструкции
```

Инструкции, которые относятся к циклу **while**, располагаются на последующих строках и должны иметь отступ от начала ключевого слова **while**.

Цикл **for** проходит по набору значений, помещает каждое значение в переменную с которой в цикле можно производить различные действия.

Синтаксис:

```
for переменная in набор_значений:
    инструкции
```

После ключевого слова **for** идет название переменной, в которую будут помещаться значения. После оператора **in** указывается набор значений и двоеточие. Инструкции, которые относятся к циклу, располагаются на последующих строках и должны иметь отступ.

При создании цикла удобно пользоваться функцией **range(a, b)**, которая создает последовательность чисел от **a** до **b-1**.

#### 3.2.1. Пример

**Задача:**

	Вывести все числа от 1 до 10 с шагом 3.
--	---

**Решение:**

	<pre>x = 1 while x &lt;= 10:     print(x)     x += 3</pre>
--	--

**Ответ:**

	<pre>1 4 7 10</pre>
--	---------------------

#### 3.2.2. Пример

**Задача:**

--	--

	Задан список. Напечатать все элементы списка.
<b>Решение:</b>	
	<pre>models = ['KNN', 'decision tree', 'linear model'] for model in models:     print(model)</pre>
<b>Ответ:</b>	
	<pre>KNN decision tree linear model</pre>
<b>3.2.3. Пример</b>	
<b>Задача:</b>	
	Создать список от 1 до 99 с шагом 7.
<b>Решение:</b>	
	<pre>list_int = range(1, 100, 7) print(list(list_int))</pre>
<b>Ответ:</b>	
	[1, 8, 15, 22, 29, 36, 43, 50, 57, 64, 71, 78, 85, 92, 99]
<b>3.2.4. Пример</b>	
<b>Задача:</b>	
	Вывести на экран числа от 5 до 105 с шагом 25, используя функцию range()
<b>Решение:</b>	
	<pre>for i in range(5, 106, 25):     print(i)</pre>
<b>Ответ:</b>	
	<pre>5 30 55 80 105</pre>
<b>3.2.5. Пример</b>	
<b>Задача:</b>	
	<b>Срез.</b> Необходимо получить каждый k-й элемент массива a, начиная с элемента с индексом i (включительно) и заканчивая элементом с индексом j (не включительно), с шагом 2. Переставить все элементы в обратном порядке.
<b>Решение:</b>	

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
b = a[1:7:2]
c = a[::-1]
print(b)
print(c)
```

**Ответ:**

```
[2, 4, 6]
[9, 8, 7, 6, 5, 4, 3, 2, 1]
```

### 3.3.1 Задание

**Задача:**

Выведите числа из примера на while (3.2.1) в обратном порядке.

**Решение:**

```
x = 10
while x > 0:
    print(x)
    x -= 3
```

**Ответ:**

```
10
7
4
1
```

### 3.3.2 Задание

**Задача:**

При решении задач машинного обучения часто в качестве объектов исследования выступает человек. Создайте список значимых характеристик (признаков), идентифицирующих человека. Выведите список на экран.

**Решение:**

```
human_features = [
    "Имя и фамилия",
    "Возраст",
    "Пол",
    "Рост",
    "Вес",
    "Группа крови",
    "Национальность",
    "Место рождения",
```

```
"Адрес проживания",  
"Семейное положение",  
"Образование",  
"Профессия",  
"Доход",  
]  
  
for i in range(len(human_features)):  
    print(human_features[i])
```

**Ответ:**

```
Имя и фамилия  
Возраст  
Пол  
Рост  
Вес  
Группа крови  
Национальность  
Место рождения  
Адрес проживания  
Семейное положение  
Образование  
Профессия  
Доход
```

### 3.3.3 Задание

**Задача:**

Создать список чисел от 2 до 15 с шагом 1.

**Решение:**

```
a = []  
for i in range(2,16):  
    a.append(i)  
print(a)
```

**Ответ:**

```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```



### 3.3.4 Задание

#### Задача:

Выведите числа из примера на for с функцией range() (3.2.4) в обратном порядке.

#### Решение:

```
for i in range(105, 4, -25):  
    print(i)
```

#### Ответ:

```
105  
80  
55  
30  
5
```

### 3.3.5 Задание

#### Задача:

Срез. Напишите код, который все элементы массива x с четными индексами переставит в обратном порядке. Т.е. если x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], то код должен сформировать [8, 1, 6, 3, 4, 5, 2, 7, 0, 9].

#### Решение:

```
x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
even = x[::2]  
even = even[::-1]  
x[::2] = even  
print(x)
```

#### Ответ:

```
[8, 1, 6, 3, 4, 5, 2, 7, 0, 9]
```

## 4.1. Теоретический материал - Библиотеки

Машинное обучение опирается на фундаментальные понятия и законы математики и информатики. Библиотека математических функций (math) применяется при решении большинства прикладных задач анализа данных. Также, и некоторые другие библиотеки Python содержат математические функции (например, NumPy). Важную роль играет библиотека matplotlib, которая позволяет производить визуализацию расчетов в удобной для восприятия человеком форме: графики, диаграммы, гистограммы.

Подключение библиотек – синтаксис:

```
import math as m
import numpy as np
import matplotlib.pyplot as plt
etc...
```

Библиотеки в python содержат не только функции, позволяющие решать различные задачи, но и некоторые популярные константы, например число Эйлера.

### 4.2.1 Пример

**Задача:**

Рассчитать синус от  $e$  радиан и вывести результат на экран

**Решение:**

```
1 import math as m
2 print (m.sin (m.e))
```

**Ответ:**

0.41078129050290885

### 4.2.2 Пример

**Задача:**

Для функции  $\sin(xe^{\cos \cos(x)})$  на интервале  $[0;10]$  с шагом 0,05 построить график функции, закрасить площадь под ним и вычислить значение этой площади.

Для вычисления площади воспользуйтесь функцией `trapz()`, вычисляющей интеграл по правилу трапеции. Для ее корректной работы необходимо подключить следующие библиотеки:

```
from scipy.integrate import simps
from numpy import trapz
```

**Решение:**

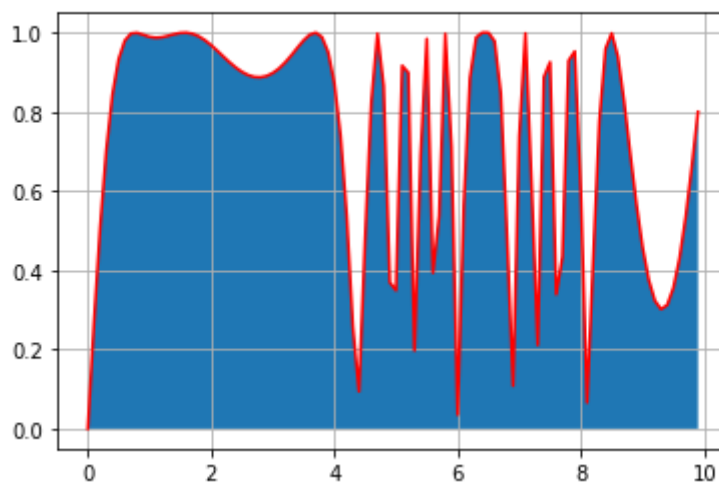
```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import simpson
4 from numpy import trapz
5 x = np.arange(0.0, 10, 0.1)
6 y = np.abs(np.sin(x*np.exp(np.cos(x))))
7 plt.grid()
8 plt.plot(x, y, c = "r")
9 plt.fill_between(x, y)
10
11 area = trapz(y)
12 print (area)

```

**Ответ:**

73.50792907748662



### 4.2.3 Пример

**Задача:**

Дано некоторое распределение оценок в группе за экзамен. Нарисовать круговую и точечную диаграммы распределения.

**Решение:**

```

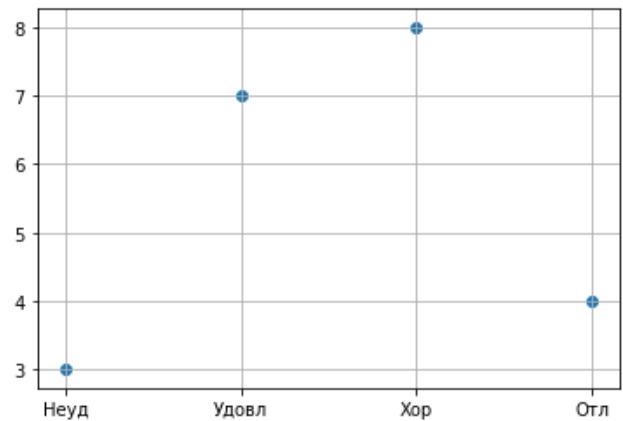
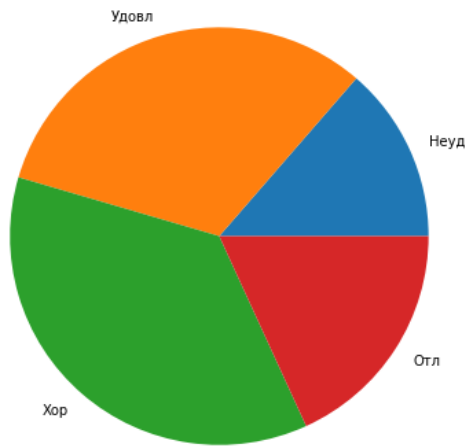
from matplotlib import pyplot as plt
import numpy as np

marks = ['Неуд', 'Удовл', 'Хор', 'Отл']

data = [3, 7, 8, 4]
fig = plt.figure(figsize=(10, 7))
plt.pie(data, labels = marks)
plt.show()
plt.grid()
plt.scatter (marks,data)

```

**Ответ:**



### 4.3.1 Задание

#### Задача:

Задайте массив случайных значений из интервала (0; 1).

Рассчитайте средние и медианные значения для массива, сравните результаты, какие выводы можно сделать о значениях?

Постройте точечную диаграмму рассеяния полученного ряда.

#### Решение:

```
import numpy as np
import random
import matplotlib.pyplot as plt

x = np.array([random.random() for i in range(100)])

bins = np.arange(0.1, 1.1, 0.1)
counts, _ = np.histogram(x, bins=bins)
plt.scatter(bins[:-1], counts)
plt.xlabel('Интервал')
plt.ylabel('Количество точек')
plt.title('Количество точек в интервале (0.1, 0.2, ..., 1.0)')
plt.xticks(bins[:-1])
plt.grid(True)
plt.show()

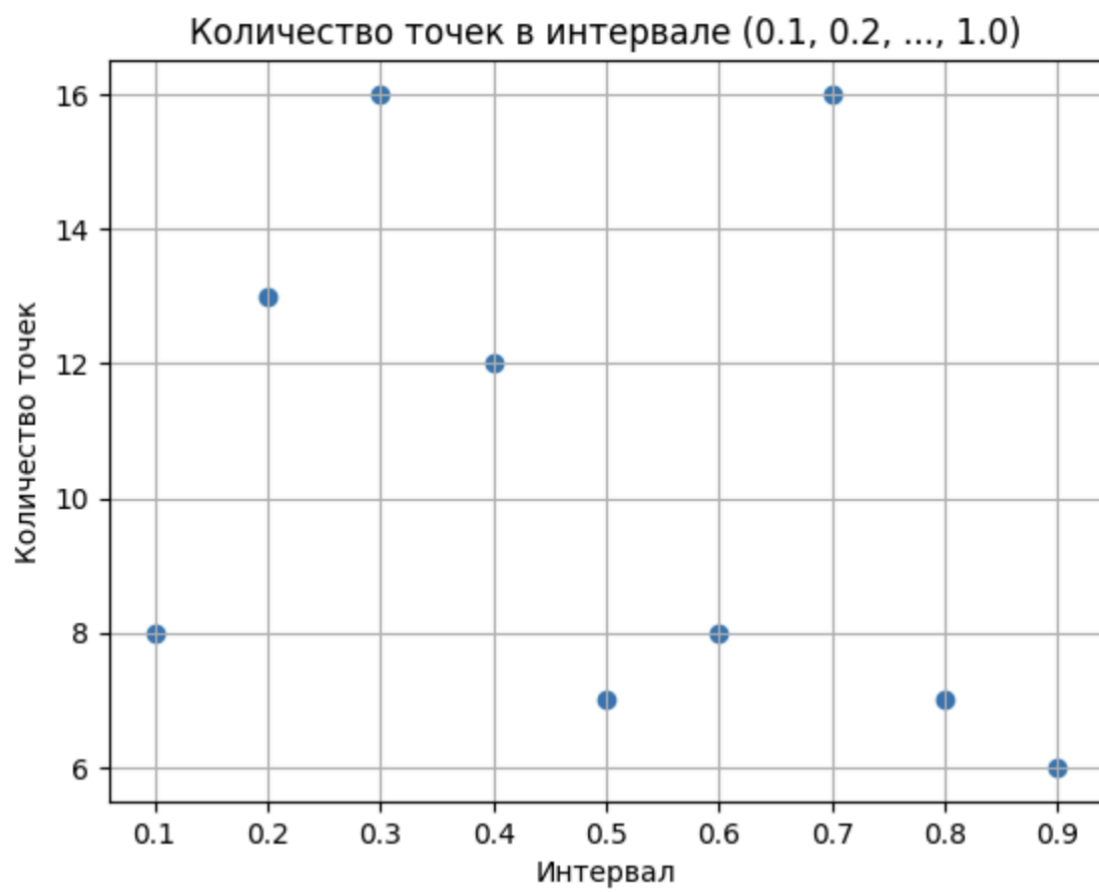
print('Median:', np.median(x))
print('Mean:', np.mean(x))
```

#### Ответ:

```
[0.4040862 0.53401655 0.41156378 0.38401147 0.49326433 0.44362584
0.65260676 0.7874229 0.13227574 0.549398 0.40313877 0.97933588
0.47165091 0.06469387 0.56996513 0.37221387 0.73045104 0.56088042
0.98842835 0.92255929 0.34003011 0.6170457 0.77387003 0.22540127
0.16220763 0.21178467 0.99449737 0.35855377 0.5741272 0.7320413
0.89008526 0.12194102 0.15002672 0.39570468 0.49016607 0.79211988
0.90326343 0.67031311 0.6158249 0.89413646 0.25969957 0.55421087
0.59566232 0.09219522 0.07899312 0.51408799 0.4623645 0.06199449
0.17224548 0.09054767 0.81666815 0.69957795 0.7344848 0.44983274
0.1857812 0.77832336 0.55843891 0.78858403 0.84284149 0.55711028
0.76825183 0.93694921 0.84965216 0.91460321 0.70660993 0.62733677
0.20059323 0.01400731 0.34990056 0.53447432 0.07903655 0.54967003
0.3378981 0.76244091 0.30061358 0.97604048 0.21180202 0.27196081
0.05394287 0.93910913 0.63766397 0.61394193 0.16969738 0.23554541
0.90275175 0.92185097 0.84135005 0.81320566 0.3079535 0.86546661
0.96515154 0.99440747 0.10330433 0.60326633 0.65305695 0.506305
0.46454486 0.58206949 0.98169172 0.13151171]
```

0.5556605759456432

0.5374200348147049



Median: 0.4236302808920376

Mean: 0.48017566202857476

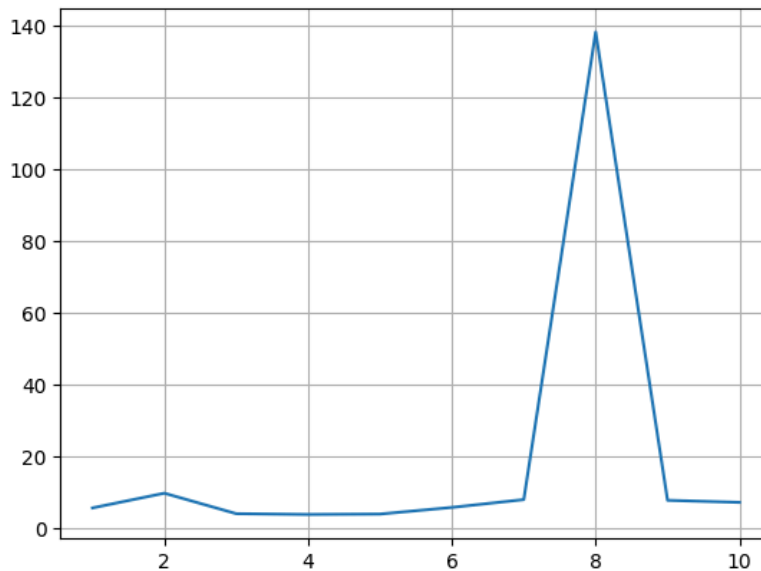
### 4.3.2 Задание

**Задача:**

Дана функция  $\frac{\sqrt{1 + e^{\sqrt{x}} + \cos x^2}}{|1 - \sin^3 x|} + \ln|2x|$ ; Создать массив из 10 значений функции ( $x$ , например, изменяется от 1 до 10). Выделить срез первой половины массива и построить графики для основного массива, используя `plt.plot()`, для среза – точечный (`plt.scatter()`).

**Решение:**

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(1, 11, 1)
func = np.sqrt(1+np.exp(np.sqrt(x))+np.cos(x**2)) / np.abs(1-(np.sin(x)**3)) +
np.log(np.abs(2*x))
plt.grid()
plt.plot(x, func)
y = x[:5]
a = func[:5]
plt.show()
plt.grid()
plt.scatter(y, a)
```

**Ответ:**

**Задача:**

Дана функция  $\left| \cos \cos \left( x e^{(\cos \cos (x) + \ln \ln (x+1))} \right) \right|$  Построить график на интервале (0, 10) с шагом 1 с заливкой площади и найти эту площадь под ним. Для вычисления площади воспользуйтесь функцией `trapz()`, вычисляющей интеграл по правилу трапеции.

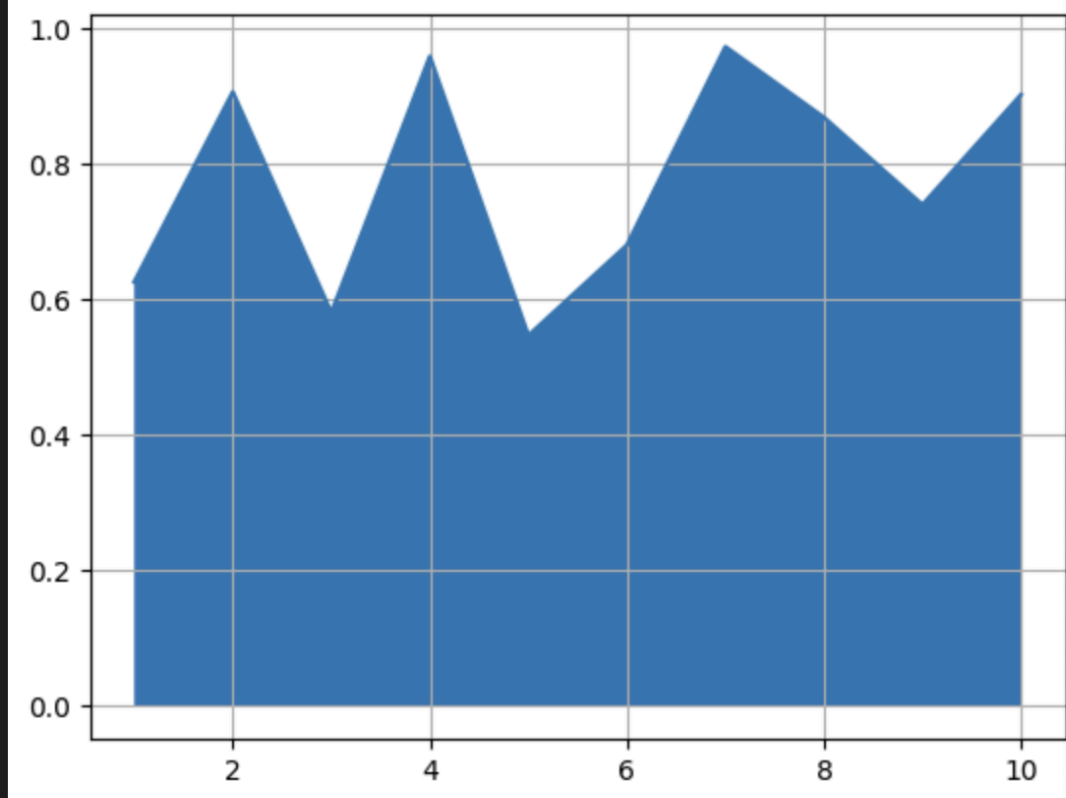
Для ее корректной работы необходимо подключить следующие библиотеки:

```
from scipy.integrate import simpson
from numpy import trapz
```

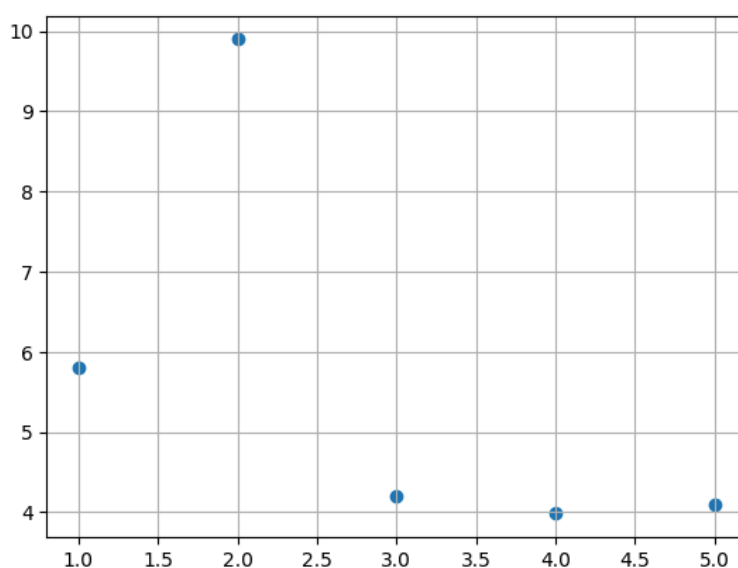
**Решение:**

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simpson
from numpy import trapz
x = np.arange(1, 11, 1)
func = np.abs(np.cos(np.cos(x*np.e**(np.cos(np.cos(x)+np.log(np.log(x+1)))))))
plt.grid()
plt.plot(x, func)
plt.fill_between(x, func)
plt.show()
area = np.trapz(func)
print(area)
```

**Ответ:**



7.018483139251277



#### 4.3.4 Задание

##### Задача:

Откройте статистические данные по стоимости акций Apple, Microsoft, Google за 12 месяцев 2021 г. Постройте 3 графика на плоскости и оцените их динамику. Подумайте, как можно улучшить визуализацию результатов.

Для начальных данных допустимо использовать следующий ресурс:  
<https://finance.yahoo.com>



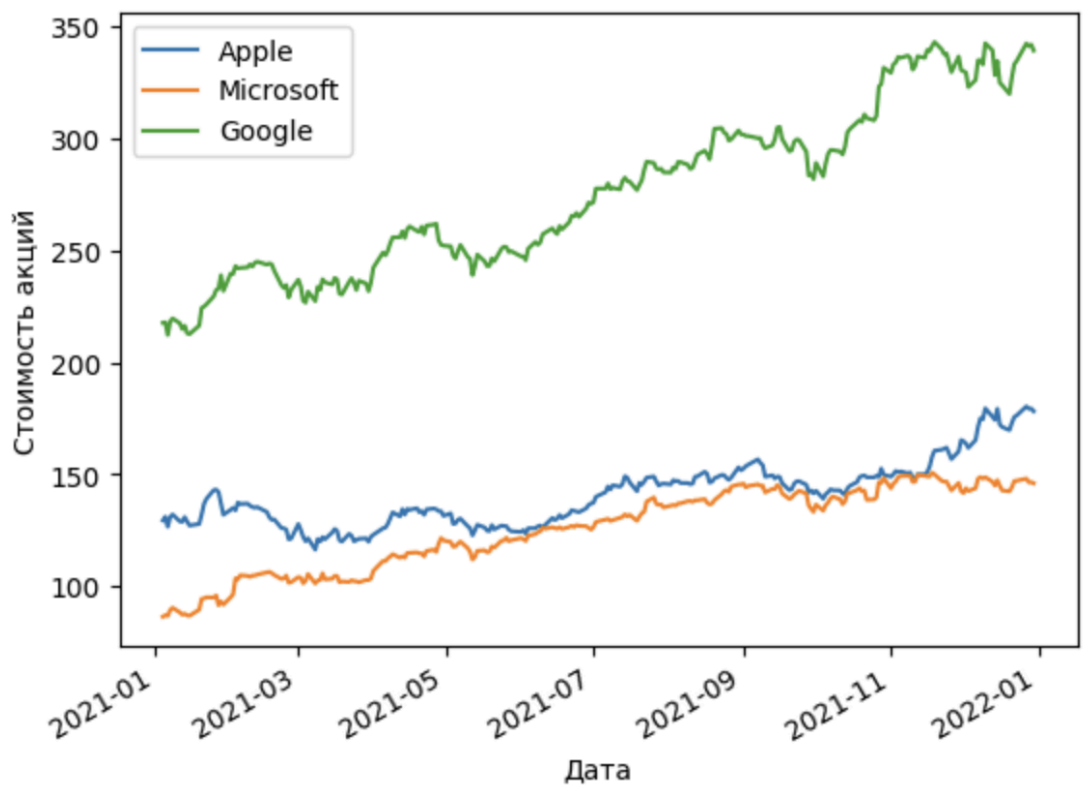
### Решение:

```
import yfinance
import matplotlib.pyplot as plt

data = yfinance.download("AAPL, MSFT, GOOG", "2021-01-01", "2021-12-31")

data['Close'].plot()
plt.xlabel('Дата')
plt.ylabel('Стоимость акций')
plt.legend(labels = ['Apple', 'Microsoft', 'Google'])
plt.show()
```

### Ответ:



### 4.3.5 Задание

#### Задача:

Создайте простейший калькулятор, включающий основные действия для двух переменных '+', '-', '\*', '/', а также вычисление следующих функций:  $e^{x+y}$ ,  $\sin \sin(x+y)$ ,  $\cos \cos(x+y)$ ,  $x^y$ .

#### Решение:

```
import numpy as np
x = int(input())
y = int(input())
```

```

print('1. x+y 2. x-y 3. x*y 4. x/y 5. e^(x+y) 6. sin(x+y) 7. cos(x+y) 8. x^y')
n = int(input())
match n:
    case 1:
        print(f'{x} + {y} = {x+y}')
    case 2:
        print(f'{x} - {y} = {x-y}')
    case 3:
        print(f'{x} * {y} = {x*y}')
    case 4:
        if y != 0:
            print(f'{x} ÷ {y} = {x/y}')
        else:
            print('На ноль делить нельзя')
    case 5:
        print(f'e^({x} + {y}) = {np.e** (x+y)}')
    case 6:
        print(f'sin({x} + {y}) = {np.sin(x+y)}')
    case 7:
        print(f'cos({x} + {y}) = {np.cos(x+y)}')
    case 8:
        print(f'{x}^{{y}} = {x**y}')

```

**Ответ:**

1. x+y 2. x-y 3. x\*y 4. x/y 5. e^(x+y) 6. sin(x+y) 7. cos(x+y) 8. x^y  
3 + 4 = 7

1. x+y 2. x-y 3. x\*y 4. x/y 5. e^(x+y) 6. sin(x+y) 7. cos(x+y) 8. x^y  
3534 0  
На ноль делить нельзя

1. x+y 2. x-y 3. x\*y 4. x/y 5. e^(x+y) 6. sin(x+y) 7. cos(x+y) 8. x^y  
cos(23 + 32555) = 0.950542716022868